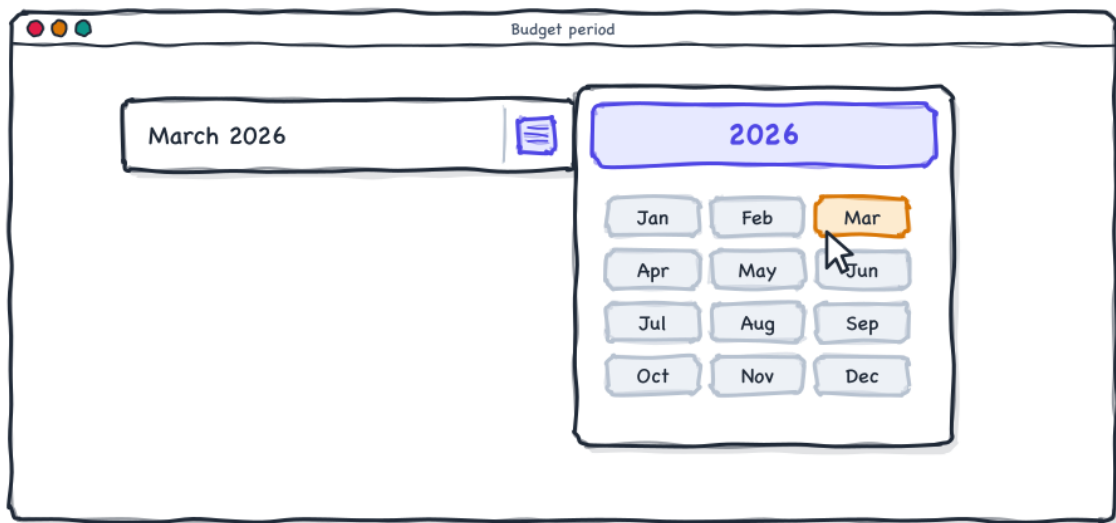


YearMonthPicker

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



YearMonthPicker opens a YearMonthView popup for month selection.

YearMonthPicker is a CustomComboBox for selecting a YearMonth. It uses a text editor, a calendar icon button and a YearMonthView popup, with a converter that defaults to full month name plus year.

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. Editing behaviour	4
6. Button display	5
7. Styling	5
7.1 Style classes and pseudo classes	5
7.2 Styleable CSS properties	6
8. Localization	6
9. Accessibility	6
10. Recipes	6
10.1 ISO year-month strings	6
10.2 Button-only chooser	6
10.3 Preset current quarter start	6
10.4 Customize popup	6
10.5 Checklist	7
11. See also	7

1. Introduction

YearMonthPicker extends `CustomComboBox<YearMonth>`. It starts at `YearMonth.now()`, formats values as `MMMM yyyy` and parses invalid text to null.

1.1 Key features

- Editable `TextField` returned by `getEditor()`.
- `YearMonthView` popup bound bidirectionally to the picker value.
- Calendar trigger button using an Ikonli `mdi-calendar` icon.
- UP and DOWN keys step the current value by one month.
- Button placement inherited from `CustomComboBox`.

1.2 Maven dependency

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

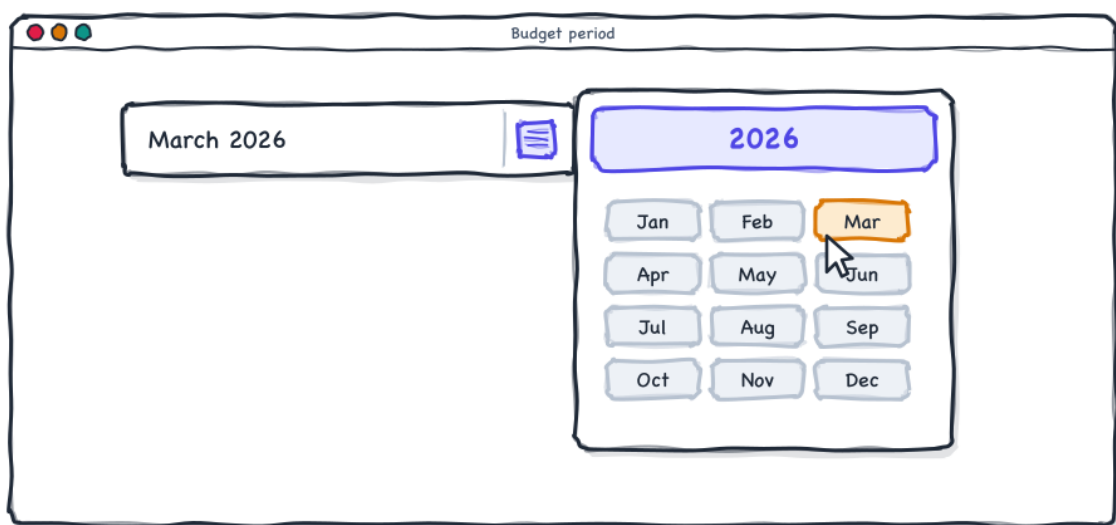
YearMonthPicker is in package `com.dlsc.gemsfx`.

2. Getting started

```
YearMonthPicker picker = new YearMonthPicker();
picker.setValue(YearMonth.of(2026, 3));

picker.valueProperty().addListener((obs, oldMonth, month) -> {
    if (month != null) {
        System.out.println("period starts " + month.atDay(1));
    }
});
```

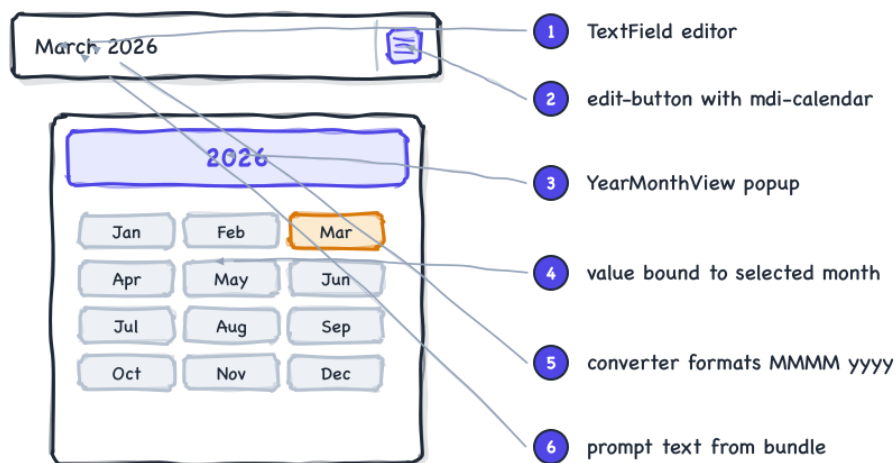
A complete YearMonthPicker setup.



YearMonthPicker in a budget-period form.

Note. The default editor prompt is localized from `prompt.example-month-year` and reads *Example: March 2023* in English.

3. Anatomy



The parts of a YearMonthPicker.

Part	Style class	Description
Root	year-month-picker, text-input	CustomComboBox root with accessible role COMBO_BOX.
Editor	TextField	Editable state is bound to the picker; action and focus loss commit text.
Button	edit-button	Popup trigger, position controlled by buttonDisplay.
Icon	edit-icon / ikonli-font-icon	mdi-calendar icon from Ikonli.
Popup	YearMonthView	Returned by <code>getYearMonthView()</code> ; value is bidirectionally bound on first show.
Converter	converter	Formats and parses YearMonth values.

4. Control API

Property	Type	Default	Description
value	ObjectProperty<YearMonth>	YearMonth.now()	Selected month inherited from ComboBoxBase.
editable	BooleanProperty	true (ComboBoxBase default, editor bound)	Controls whether text can be typed.

Property	Type	Default	Description
converter	ObjectProperty<String Converter<YearMonth>>	MMMM yyyy converter	Formats YearMonth; invalid default parsing returns null. Attempts to set converter to null restore the old converter.
buttonDisplay	ObjectProperty<CustomComboBox.ButtonDisplay>	RIGHT	Styleable inherited property: LEFT, RIGHT, BUTTON_ONLY, FIELD_ONLY.
promptText	StringProperty	ComboBoxBase default null	Not bound to the editor; the editor receives its localized prompt directly.
getEditor()	TextField	created in constructor	Manual input field.
getYearMonthView()	YearMonthView	lazy new YearMonthView()	Popup month view.

```
picker.setConverter(new StringConverter<>() {
    public String toString(YearMonth ym) { return ym == null ? "" : ym.toString(); }
    public YearMonth fromString(String text) { return YearMonth.parse(text); }
});
```

5. Editing behaviour



Text commits and arrow keys all update the YearMonth value.

Action	Effect
ENTER in editor	Calls commit() and parses text through converter.
Editor loses focus	Commits the text.
Touch press	Commits current text and shows popup.
DOWN key	Sets value to <code>getValue().plusMonths(1)</code> .
UP key	Sets value to <code>getValue().minusMonths(1)</code> .
Popup month click	Updates value and hides popup when the value changes.

Careful. The UP/DOWN key handler assumes the value is not null. If your converter may return null, guard keyboard usage or restore a non-null value after invalid input.

6. Button display



The inherited buttonDisplay property changes which nodes are visible.

The skin rebuilds its HBox when buttonDisplay changes. LEFT places the edit button before the editor, RIGHT after it, BUTTON_ONLY hides the editor and FIELD_ONLY hides the button.

```
picker.setButtonDisplay(CustomComboBox.ButtonDisplay.LEFT);
// or in CSS: -fx-button-display: button-only;
```

7. Styling

7.1 Style classes and pseudo classes

Selector	Meaning
.year-month-picker	Root.
.text-input	Additional root class set by constructor.
.year-month-view	Popup view, styled with a drop shadow while inside the picker.
.box	HBox containing editor and edit button.
.edit-button	Calendar trigger button.
.edit-button > .ikonli-font-icon	Calendar icon, style class edit-icon.
.text-field	Editor.
:left, :right, :button-only, :field-only	Inherited button display pseudo classes.
:focused	Mirrors editor focus.

7.2 Styleable CSS properties

CSS property	Type	Default
-fx-button-display	ButtonDisplay	RIGHT

```
.year-month-picker {
    -fx-button-display: left;
}

.year-month-picker > .box > .edit-button > .ikonli-font-icon {
    -fx-icon-color: -fx-accent;
}
```

8. Localization

Key	English default
prompt.example-month-year	Example: March 2023

9. Accessibility

YearMonthPicker sets `AccessibleRole.COMBO_BOX`. Its automatic accessible text is the current value formatted by converter, or null when the value is null. The binding follows `AccessibilityUtil` semantics and yields if application code sets `accessibleText` manually.

10. Recipes

10.1 ISO year-month strings

```
picker.setConverter(new StringConverter<>() {
    public String toString(YearMonth ym) { return ym == null ? "" : ym.toString(); }
    public YearMonth fromString(String text) { return text == null || text.isBlank() ? null
        : YearMonth.parse(text); }
});
```

10.2 Button-only chooser

```
picker.setButtonDisplay(CustomComboBox.ButtonDisplay.BUTTON_ONLY);
```

10.3 Preset current quarter start

```
YearMonth now = YearMonth.now();
int quarterStart = ((now.getMonthValue() - 1) / 3) * 3 + 1;
picker.setValue(YearMonth.of(now.getYear(), quarterStart));
```

10.4 Customize popup

```
picker.getYearMonthView().setShowYear(true);
```

10.5 Checklist

- 1 Keep value non-null when relying on UP/DOWN keyboard stepping.
- 2 Use a converter that matches the prompt shown to users.
- 3 Set `buttonDisplay` through CSS for visual-only changes.
- 4 Customize `getYearMonthView()` before opening the popup.

11. See also

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