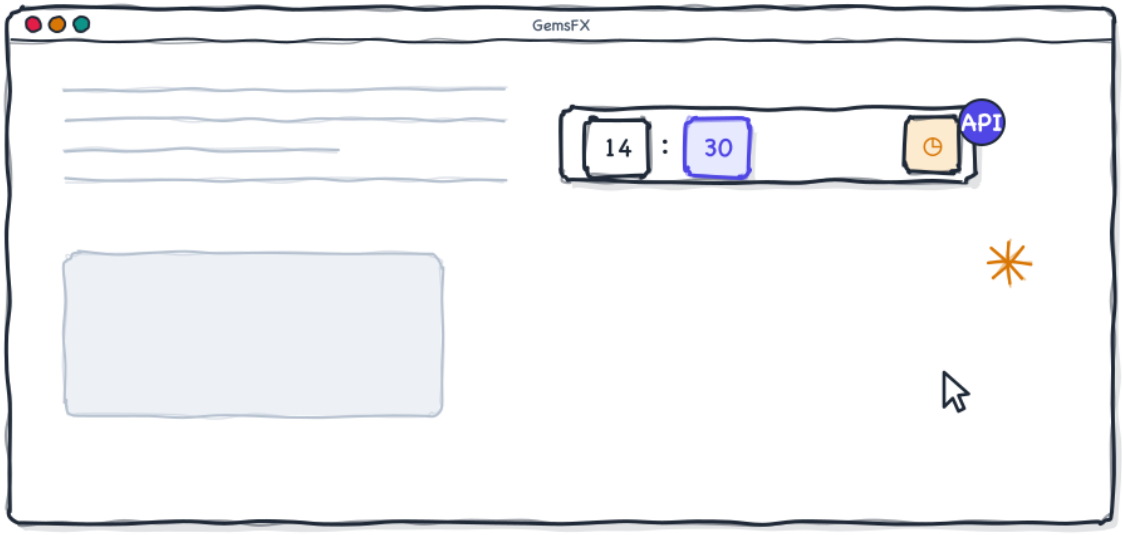
TimePicker

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of TimePicker.

TimePicker edits a LocalTime through separate hour, minute, second and millisecond fields plus an optional popup of list views. It supports bounds, minute step sizes and linked rollover behavior.

Table of Contents

1. Introduction

TimePicker is part of the GemsFX module `com.dlsc.gemsfx`. This manual documents only behavior verified in the control, skin, CSS, resource bundles and demo app.

1.1 Key features

- Supports HOURS_MINUTES, HOURS_MINUTES_SECONDS and HOURS_MINUTES_SECONDS_MILLIS formats.
- Bounds input with earliestTime and latestTime.
- Minute field can step by 1 to 60 minutes.
- Fields can roll over and link to adjacent fields.
- Popup trigger button can be shown, hidden or repositioned through CustomComboBox.

1.2 Maven dependency

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

Use package `com.dlsc.gemsfx`.

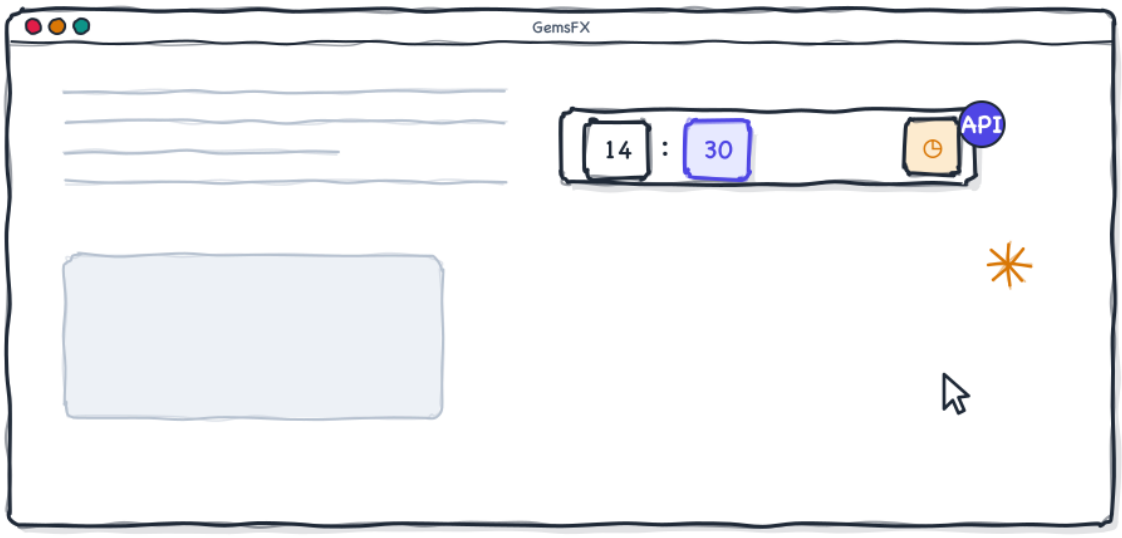
2. Getting started

Create the control, set the properties that define its valid values, and listen to the value or selection property that the control owns.

```
TimePicker picker = new TimePicker();
picker.setFormat(TimePicker.Format.HOURS_MINUTES_SECONDS);
picker.setEarliestTime(LocalTime.of(8, 0));
picker.setLatestTime(LocalTime.of(18, 0));
picker.setStepRateInMinutes(15);

picker.timeProperty().addListener((obs, oldTime, newTime) -> {
    System.out.println("time = " + newTime);
});
```

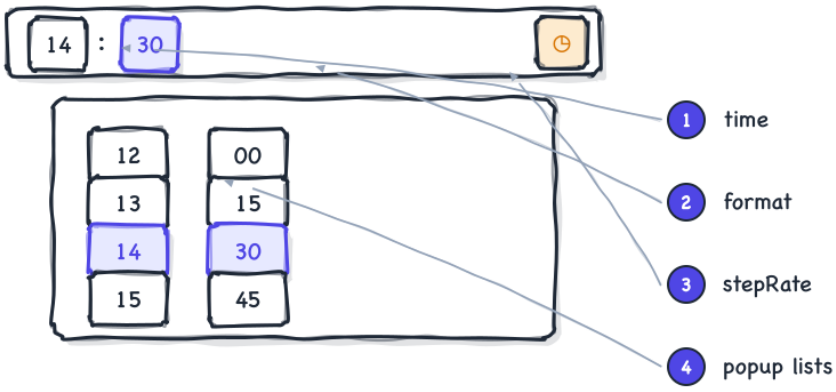
A compact setup for `TimePicker`.



A first look at the control in a simple application window.

3. Anatomy

The skin builds the visible nodes below the public control. The table lists the style classes and nodes that are useful when reading the source or writing CSS.



The main parts of the control.

Part	Style / node	Description
Root	time-picker text-input	CustomComboBox root style classes.
Box	box	HBox containing fields, spacer and edit button.
Fields	fields-box	Hour, minute, optional second and optional millisecond fields.
Separators	separator	Nodes between fields; defaults are ":", ":" and ".".

Part	Style / node	Description
Edit button	edit-button	Clock icon trigger for popup.
Popup	time-picker-popup	HBox containing hour/minute/second/millisecond ListViews.

4. Control API

4.1 Value and bounds

Property	Type	Default	Description
time	ObjectProperty<LocalTime>	LocalTime.now() at construction	Current time. setTime truncates seconds/nanos according to format.
earliestTime	ObjectProperty<LocalTime>	LocalTime.MIN	Earliest allowed time; must not be after latestTime.
latestTime	ObjectProperty<LocalTime>	LocalTime.MAX	Latest allowed time; must not be before earliestTime.
adjusted	ReadOnlyBooleanProperty	false	True after adjust() changed the value to satisfy bounds or step size.

4.2 Format and editing

Property	Type	Default	Description
format	ObjectProperty<TimePicker.Format>	HOURS_MINUTES	Controls visible fields and truncation: minutes only, seconds, or milliseconds.
stepRateInMinutes	IntegerProperty	1	Minute increment for arrow keys and popup minute list; valid 1 through 60.
clockType	ObjectProperty<ClockType>	TWENTY_FOUR_HOUR_CLOCK	Styleable enum. Source contains TWELVE_HOUR_CLOCK but popup has TODO for AM/PM support.
linkingFields	BooleanProperty	true	Rollover increments/decrements the previous field.
rollover	BooleanProperty	true	Fields wrap at their min/max instead of clamping.

4.3 Popup and separators

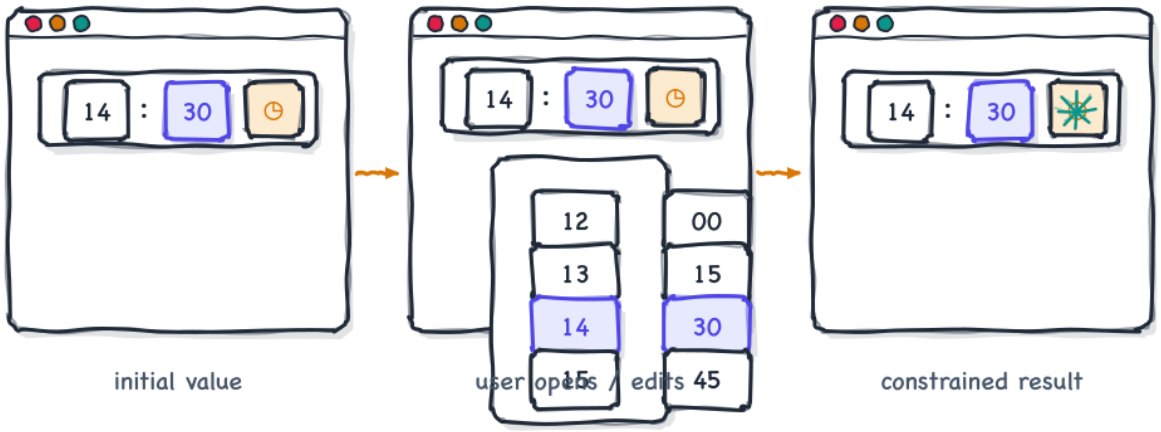
Property	Type	Default	Description
showPopupTriggerButton	BooleanProperty	true	Shows the edit button. Styleable.
buttonDisplay	ObjectProperty<CustomComboBox.ButtonDisplay>	RIGHT	Inherited button placement.
onShowPopup	ObjectProperty<Consumer<TimePicker>>	picker -> show()	Consumer invoked by F4/ENTER and default button behavior.
hoursSeparator	ObjectProperty<Node>	Label ":"	Node between hours and minutes.

Property	Type	Default	Description
minutesSeparator	ObjectProperty<Node>	Label ":"	Node between minutes and seconds.
secondsSeparator	ObjectProperty<Node>	Label "."	Node between seconds and milliseconds.

5. Behaviour

5.1 Adjusting values

Calling adjust() first clamps the current time to earliestTime/latestTime, then snaps minutes to the nearest stepRateInMinutes. When a change happens, adjusted becomes true until the next editing cycle clears it.



Interaction and value-flow behaviour.

5.2 Field navigation

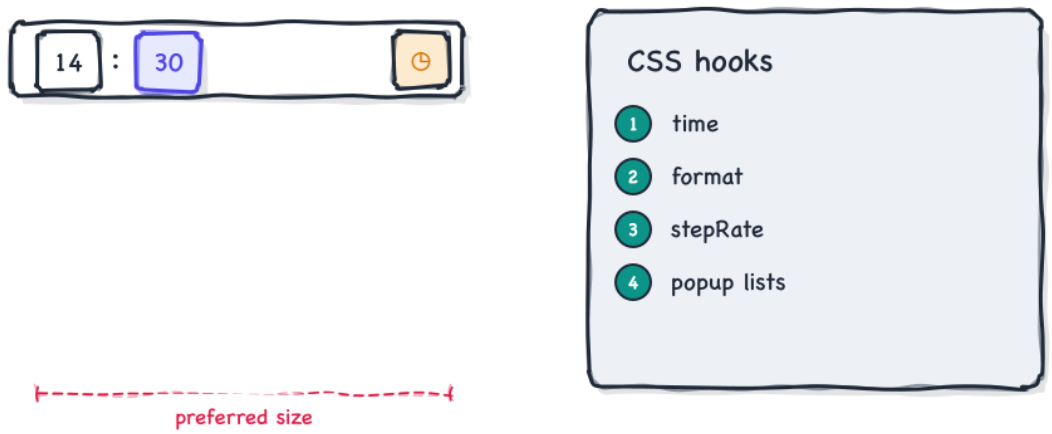
Digit input replaces field text up to its digit width, BACK_SPACE removes a digit, SPACE/RIGHT moves to the next field, LEFT moves to the previous field, and UP/DOWN increment or decrement.

5.3 Popup lists

The popup lists hours from earliestTime.hour to latestTime.hour, minutes by stepRateInMinutes, seconds 0-59 and milliseconds 0-999. It scrolls to the current selection when shown.

6. Styling

The stylesheet is loaded by the control or its base class. The rows below list style hooks verified in the CSS and skin source.



Style classes and pseudo-class states.

6.1 Style classes

Style class	Where used
time-picker	Root style class.
text-input	Additional root style class.
box	Main HBox.
fields-box	Container for time fields and separators.
time-field digits-field hour minute second millisecond	Editable unit fields.
separator	Separator nodes.
edit-button	Clock trigger button.
time-picker-popup	Popup root.
time-list-view hour-list minute-list second-list millisecond-list	Popup lists.
time-cell time-label	Popup list cell and label.

6.2 Pseudo classes

Pseudo class / marker	Meaning
left/right/button-only/field-only	Inherited from CustomComboBox buttonDisplay.
focused	Set while any unit field or edit button has focus.
empty	Applied to fields and separators when time is null.

6.3 Styleable CSS properties

Property	Type	Default	Description
-fx-button-display	ButtonDisplay	RIGHT	Inherited from CustomComboBox.
-fx-show-popup-trigger-button	Boolean	true	Shows the popup trigger button.
-fx-linking-fields	Boolean	true	Links rollover to the previous field.
-fx-rollover	Boolean	true	Wraps fields at boundaries.
-fx-clock-type	ClockType	TWENTY_FOUR_HOUR_CLOCK	TWENTY_FOUR_HOUR_CLOCK or TWELVE_HOUR_CLOCK.
-fx-format	Format	HOURS_MINUTES	Visible time fields.
-fx-step-rate-in-minutes	Integer	1	Minute step, 1 to 60.

```
.time-picker {
    -fx-format: hours-minutes-seconds;
    -fx-step-rate-in-minutes: 15;
}
.time-picker > .box > .fields-box > .time-field:focused {
    -fx-background-color: -fx-accent;
}
```

Example CSS.

7. Localization

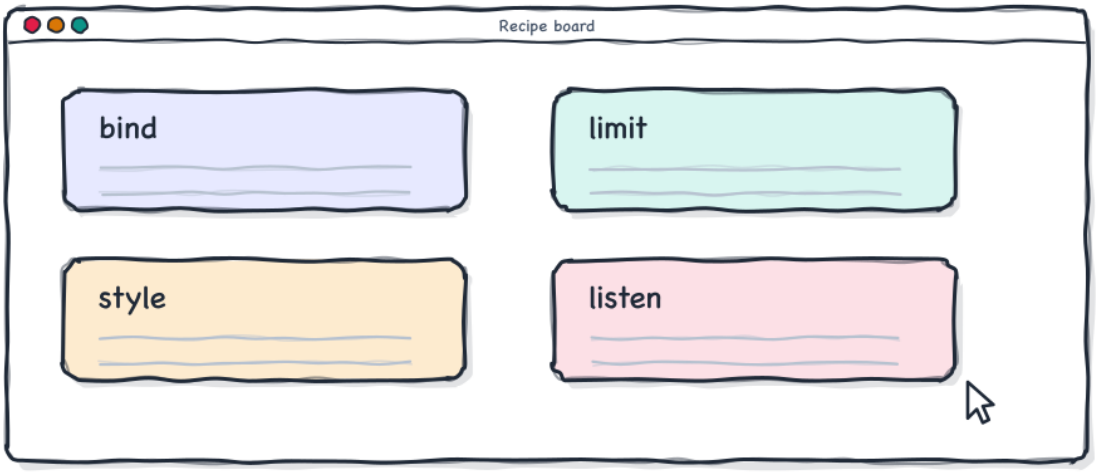
This control uses `ResourceBundleManager` for the following keys.

Key	English default
format.separator.hour-minute	:
format.separator.minute-second	:
format.separator.second-fraction	.

8. Accessibility

The constructor sets AccessibleRole.COMBO_BOX and binds accessible text to time.toString(), or null when time is null.

9. Recipes



Common configuration recipes.

9.1 Business-hours picker

```
TimePicker picker = new TimePicker();
picker.setEarliestTime(LocalTime.of(8, 0));
picker.setLatestTime(LocalTime.of(18, 0));
picker.setStepRateInMinutes(15);
```

9.2 Seconds precision

```
picker.setFormat(TimePicker.Format.HOURS_MINUTES_SECONDS);
```

9.3 Field-only editor

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

9.4 Custom separators

```
picker.setHoursSeparator(new Label("h"));
picker.setMinutesSeparator(new Label("m"));
```

9.5 Validate after programmatic update

```
picker.setTime(LocalTime.now());
picker.adjust();
if (picker.isAdjusted()) {
    System.out.println("snapped to allowed value");
}
```

9.6 Checklist

1. Keep earliestTime <= latestTime.
2. Keep stepRateInMinutes between 1 and 60.

3. Call adjust() after programmatic changes if you need snapping immediately.
4. Use format to control visible fields and truncation.

10. See also

Demo app: `TimePickerApp`. Run it with:

```
mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.TimePickerApp
```

- Related GemsFX controls: DurationPicker, TimeRangePicker.
- API documentation: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/