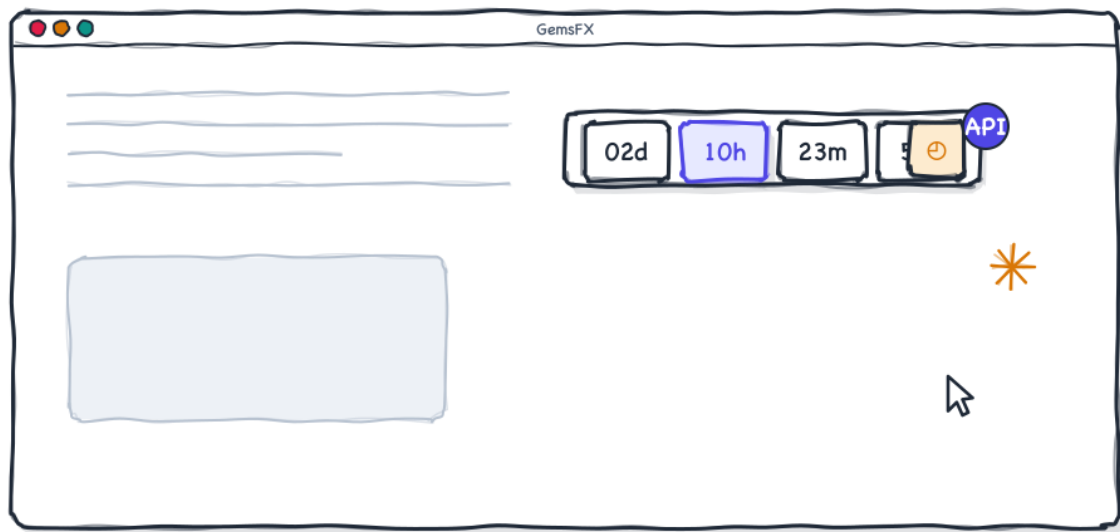


DurationPicker

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



Generated cartoon overview of DurationPicker.

DurationPicker edits a Duration through configurable duration unit fields and a PickerFX popup. It supports minimum and maximum durations, labels, zero padding and linked rollover behavior.

Table of Contents

1. Introduction	3
1.1 Key features	3
1.2 Maven dependency	3
2. Getting started	3
3. Anatomy	4
4. Control API	5
4.1 Value and limits	5
4.2 Fields and labels	5
4.3 Interaction and popup	5
5. Behaviour	6
5.1 Constraining durations	6
5.2 Unit field editing	6
5.3 Popup binding	6
6. Styling	6
6.1 Style classes	7
6.2 Pseudo classes	7
6.3 Styleable CSS properties	7
7. Localization	8
8. Accessibility	9
9. Recipes	9
9.1 Hours and minutes only	9
9.2 Cap to one work day	9
9.3 Use long labels	9
9.4 Remove leading zeroes	9
9.5 Custom separator factory	9
9.6 Checklist	10

10. See also 10

1. Introduction

DurationPicker 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

- Default duration is `Duration.ZERO`.
- Constructor configures fields to `DAYS`, `HOURS`, `MINUTES`, `SECONDS` and `MILLIS`.
- `minimumDuration` defaults to `Duration.ZERO`; `maximumDuration` is set to seven days by the constructor.
- Fields can be relabeled, hidden, padded and reordered through the fields list.
- 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`.

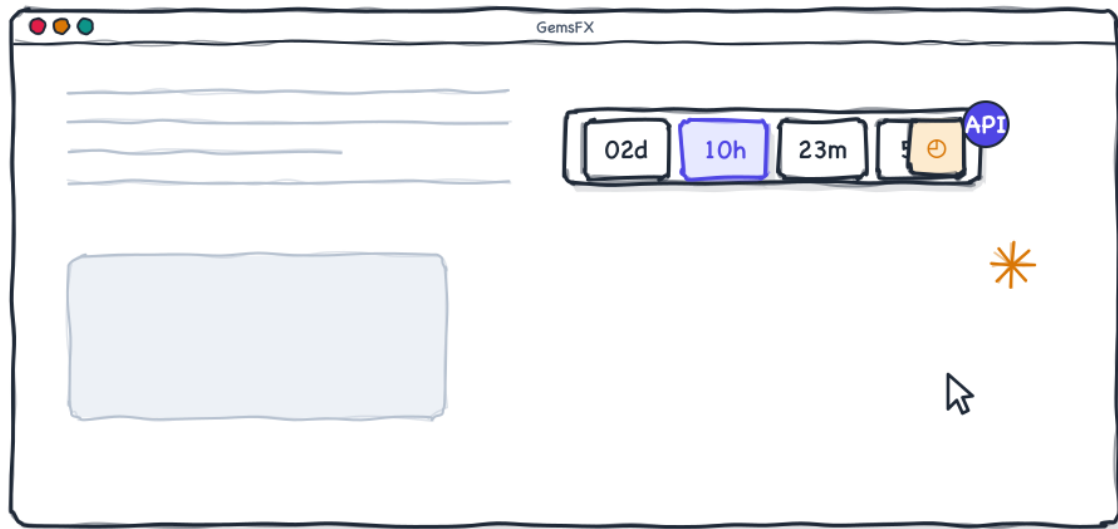
2. Getting started

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

```
DurationPicker picker = new DurationPicker();
picker.setDuration(Duration.ofHours(2).plusMinutes(30));
picker.setMinimumDuration(Duration.ZERO);
picker.setMaximumDuration(Duration.ofDays(1));
picker.getFields().setAll(ChronoUnit.HOURS, ChronoUnit.MINUTES);

picker.durationProperty().addListener((obs, oldDuration, newDuration) -> {
    System.out.println("duration = " + newDuration);
});
```

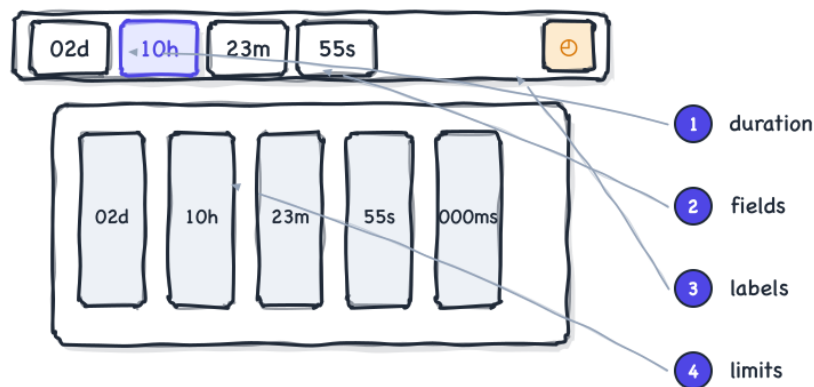
A compact setup for `DurationPicker`.



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	duration-picker text-input	CustomComboBox root style classes.
Box	box	HBox containing fields, spacer and arrow button.
Fields	fields-box	DurationUnitField labels for configured ChronoUnit values.
Unit field	unit-field days/hours/minutes/se conds/millis	Focusable label that accepts digits and arrow keys.

Part	Style / node	Description
Separator	separator	Created by separatorFactory between adjacent units.
Popup	popup / picker	PickerFX duration picker bound bidirectionally to duration.

4. Control API

4.1 Value and limits

Property	Type	Default	Description
duration	ObjectProperty<Duration>	Duration.ZERO	Current duration. Constrained when value/minimum/maximum changes.
minimumDuration	ObjectProperty<Duration>	Duration.ZERO	Minimum allowed duration. Must not be null or negative.
maximumDuration	ObjectProperty<Duration>	Duration.ofDays(7) after constructor	Maximum allowed duration; null means no upper limit.

4.2 Fields and labels

Property	Type	Default	Description
fields	ListProperty<ChronoUnit>	DAYS, HOURS, MINUTES, SECONDS, MILLIS after constructor	Displayed units. Supported by field logic: WEEKS, DAYS, HOURS, MINUTES, SECONDS, MILLIS.
labelType	ObjectProperty<LabelType>	SHORT	NONE, SHORT or LONG unit labels. Styleable.
fillDigits	BooleanProperty	true	Pads hours/minutes/seconds to two digits and millis to three digits. Styleable.
separatorFactory	ObjectProperty<Callback<Pair<ChronoUnit, ChronoUnit>, Node>>	Label ":"	Creates separator nodes between fields.

4.3 Interaction and popup

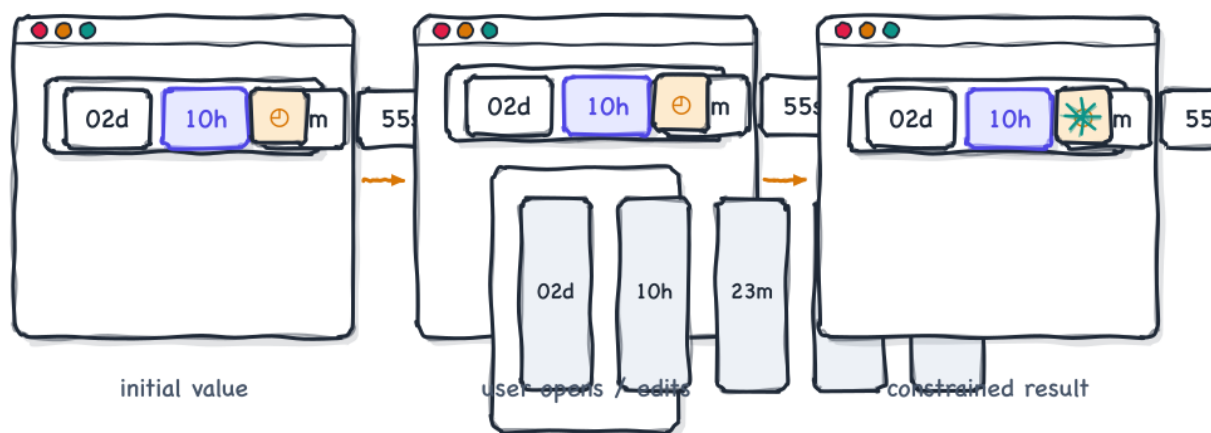
Property	Type	Default	Description
linkingFields	BooleanProperty	true	Rollover increments/decrements the previous field. Styleable.
rollover	BooleanProperty	true	Fields wrap at boundaries instead of clamping. Styleable.
showPopupTriggerButton	BooleanProperty	true	Shows the arrow button. Styleable.
buttonDisplay	ObjectProperty<CustomComboBox.ButtonDisplay>	RIGHT	Inherited button placement.

Property	Type	Default	Description
onShowPopup	ObjectProperty<Consumer<DurationPicker>>	picker -> show()	Consumer used to show alternate duration input UI.

5. Behaviour

5.1 Constraining durations

Whenever duration, minimumDuration or maximumDuration changes, the control clamps the duration to the configured interval. Negative minimum durations and null minimum durations throw `IllegalArgumentException`.



Interaction and value-flow behaviour.

5.2 Unit field editing

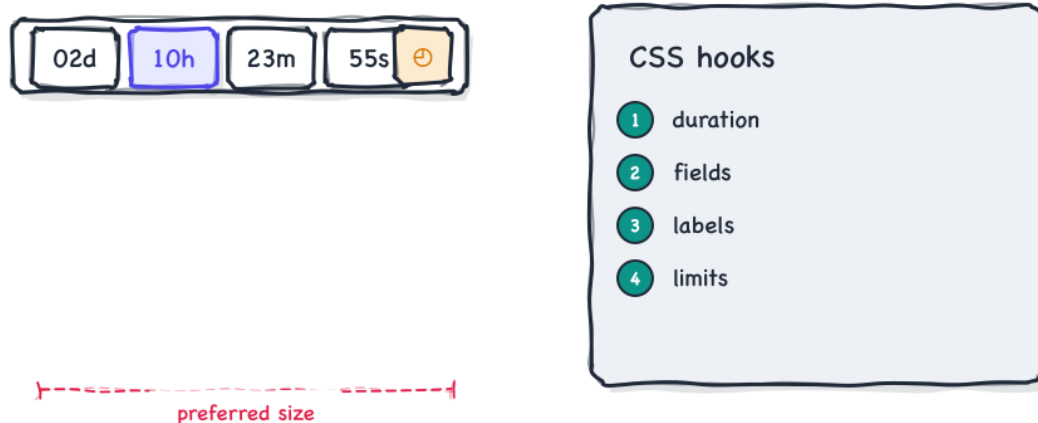
Digit input edits the focused unit, `BACK_SPACE` removes a digit, `SPACE/RIGHT` moves forward, `LEFT` moves back, and `UP/DOWN` increment or decrement. Hours max at 23, minutes/seconds at 59, millis at 999.

5.3 Popup binding

The popup creates a `PickerFX DurationPicker` whose value, fields, minimum and maximum are bound to the `GemsFX` control.

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
duration-picker	Root style class.
text-input	Additional root style class.
box	Main HBox.
fields-box	Container for fields and separators.
unit-field no-label short-label long-label	Field label and label-type classes.
days hours minutes seconds millis	Unit-specific classes on fields.
separator	Separator nodes.
arrow-button arrow	Popup trigger and icon region.
popup	Popup HBox.
picker container segment segment-cell colon indicator shadow segment-separator	PickerFX popup classes styled by duration-picker.css.

6.2 Pseudo classes

Pseudo class / marker	Meaning
left/right/button-only/field-only	Inherited from CustomComboBox buttonDisplay.
focused	Used on arrow button and focused unit field.

6.3 Styleable CSS properties

Property	Type	Default	Description
-fx-button-display	ButtonDisplay	RIGHT	Inherited from CustomComboBox.

Property	Type	Default	Description
-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-label-type	LabelType	SHORT	NONE, SHORT or LONG labels.
-fx-fill-digits	Boolean	true	Pads lower units with leading zeroes.

```
.duration-picker {
    -fx-label-type: long;
    -fx-fill-digits: true;
}
.duration-picker > .box > .fields-box > .unit-field:focus {
    -fx-background-color: -fx-accent;
}
```

Example CSS.

7. Localization

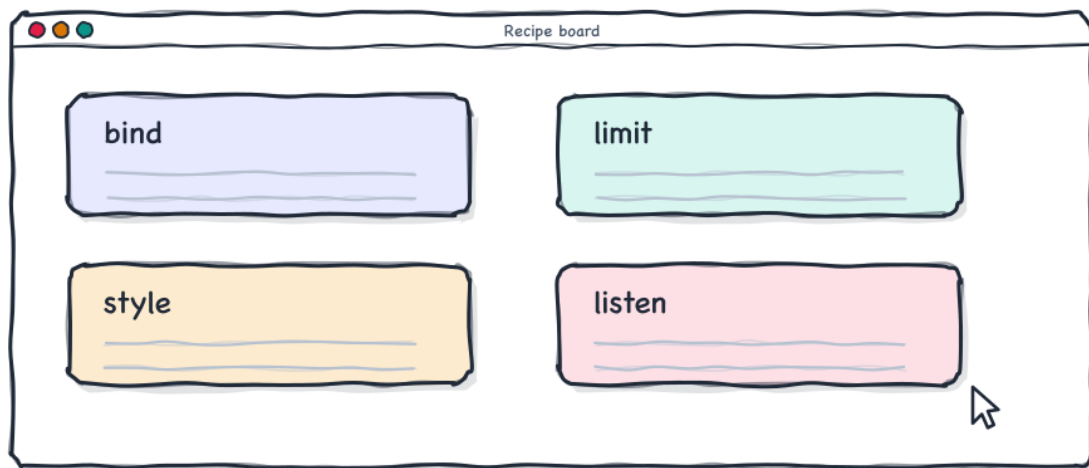
This control uses `ResourceBundleManager` for the following keys.

Key	English default
unit.short.days	d
unit.short.hours	h
unit.short.minutes	m
unit.short.seconds	s
unit.short.millis	ms
unit.long.days	days
unit.long.hours	hours
unit.long.minutes	minutes
unit.long.seconds	seconds
unit.long.millis	millis
popup.unit.title.days	Days
popup.unit.title.hours	Hours
popup.unit.title.minutes	Minutes
popup.unit.title.seconds	Seconds
popup.unit.title.millis	Millis
format.separator.time	:

8. Accessibility

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

9. Recipes



Common configuration recipes.

9.1 Hours and minutes only

```
DurationPicker picker = new DurationPicker();  
picker.getFields().setAll(ChronoUnit.HOURS, ChronoUnit.MINUTES);
```

9.2 Cap to one work day

```
picker.setMinimumDuration(Duration.ZERO);  
picker.setMaximumDuration(Duration.ofHours(8));
```

9.3 Use long labels

```
picker.setLabelType(DurationPicker.LabelType.LONG);
```

9.4 Remove leading zeroes

```
picker.setFillDigits(false);
```

9.5 Custom separator factory

```
picker.setSeparatorFactory(pair -> new Label(" | "));
```

9.6 Checklist

- 1 Do not set `minimumDuration` to null or a negative value.
- 2 Set `maximumDuration` to null only when no upper limit is wanted.
- 3 Use `ChronoUnit` values supported by `DurationUnitField`.
- 4 Remember the constructor replaces the field list with five units.

10. See also

Demo app: `DurationPickerApp`. Run it with:

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

- Related GemsFX controls: `TimePicker`, `TimeRangePicker`.
- API documentation: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>