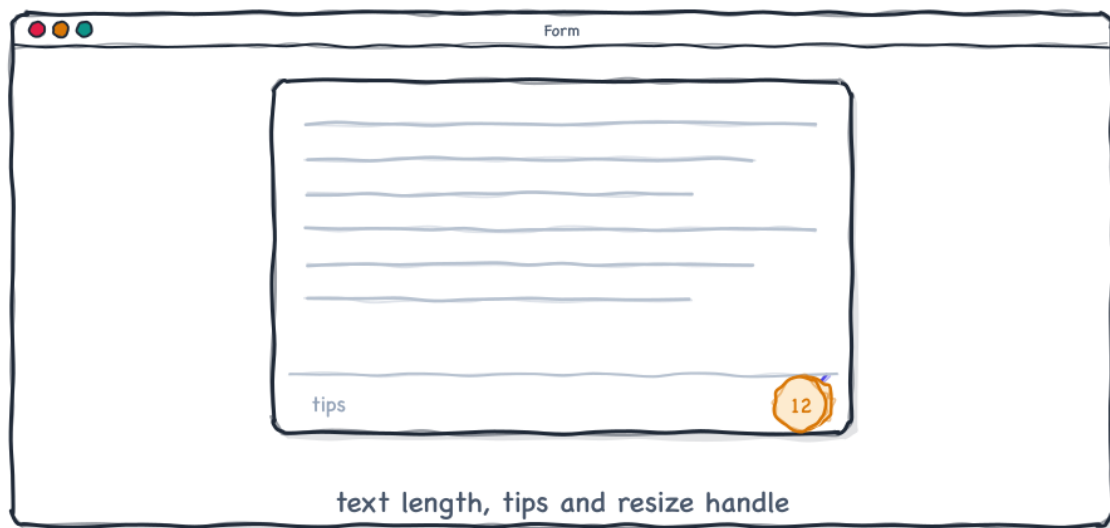


LimitedTextArea

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



A generated cartoon overview of LimitedTextArea.

LimitedTextArea builds on ResizableTextArea and adds a bottom bar with tips, a remaining-character label, a circular progress indicator and warning/error states.

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 Limits and feedback	3
4.2 Bottom bar and filtering	3
5. Length display modes	4
6. Filtering and state rules	5
7. Styling	5
7.1 Style classes and pseudo classes	5
7.2 Styleable CSS properties	5
8. Accessibility	6
9. Recipes	6
9.1 Disallow line breaks	6
9.2 Hide the bottom bar	6
9.3 Warn earlier	6
9.4 Checklist	6
10. See also	6

1. Introduction

LimitedTextArea is a `ResizableTextArea` specialized for text length guidance. It does not block typing beyond the configured range; instead it reports and styles normal, warning and error states.

1.1 Key features

- Character range limit with minimum and maximum.
- Bottom bar with optional tips and a length indicator.
- Circular progress indicator up to the maximum length.
- Warning and error pseudo classes.
- Automatic removal of configured excluded text fragments.

1.2 Maven dependency

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

Maven coordinates for the GemsFX control library.

2. Getting started

```
LimitedTextArea area = new LimitedTextArea();
area.setWrapText(true);
area.setCharacterRangeLimit(new IntegerRange(0, 140));
area.setTips("Write a short summary.");
area.setLengthDisplayMode(LimitedTextArea.LengthDisplayMode.ALWAYS_SHOW);

area.outOfRangeProperty().addListener((obs, old, out) -> saveButton.setDisable(out));
```

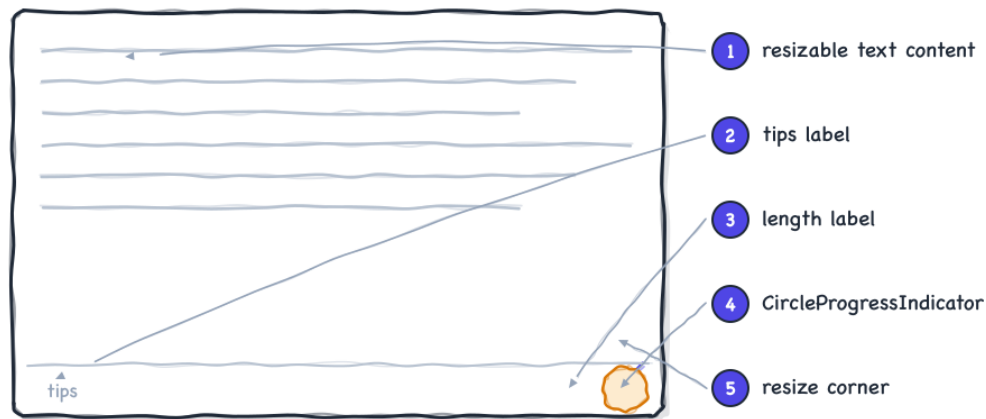
A complete limited text area setup.



AUTO display appears for warning and error

Normal, warning and error length states.

3. Anatomy



Parts added by LimitedTextAreaSkin.

Part	Node	Description
Content	inherited resizable content pane	TextArea with resize corner.
Bottom box	.bottom-box	Contains tips, spacer and length indicator.
Tips	.tips	Label bound to tips, hidden when empty.
Length indicator	.length-indicator	Stacks label and CircleProgressIndicator.

4. Control API

4.1 Limits and feedback

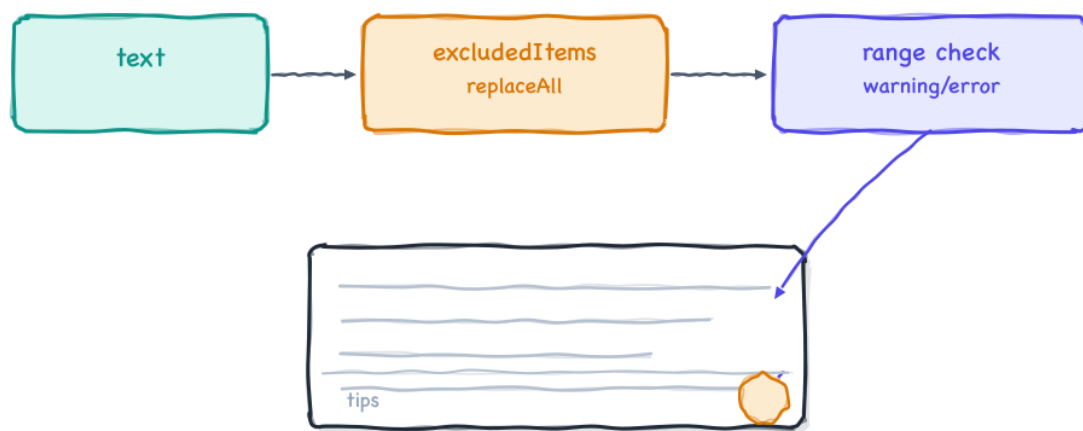
Property	Type	Default	Description
characterRangeLimit	ObjectProperty<IntegerRange>	null	Minimum and maximum allowed length. Null or max <= 0 disables range feedback.
outOfRange	ReadOnlyBooleanProperty	false	True when text length is below min or above max.
warningThreshold	DoubleProperty	0.9	Warning starts at max * clamped threshold, unless already in error.
lengthDisplayMode	ObjectProperty<LengthDisplayMode>	AUTO	AUTO, ALWAYS_SHOW or ALWAYS_HIDE.

4.2 Bottom bar and filtering

Property	Type	Default	Description
showBottom	BooleanProperty	true	Shows the bottom box. Styleable.

Property	Type	Default	Description
tips	StringProperty	null	Hint text displayed at the bottom left.
excludedItems	ObservableList<String>	empty list	Fragments removed from text using replaceAll.
resizeVertical	BooleanProperty	true	Inherited from ResizableTextArea.
resizeHorizontal	BooleanProperty	false	Inherited from ResizableTextArea.

5. Length display modes



Filtering and range checking flow.

Mode	Indicator behaviour
AUTO	Label appears only for warning or error; progress appears while max is known and text is not over max.
ALWAYS_SHOW	Label is always visible; progress follows the max.
ALWAYS_HIDE	Label and progress are hidden.

Without a positive maximum the label displays the current text length only when the selected mode makes it visible. With a maximum it displays remaining characters, abbreviated with k, M, G, T, P or E for large values.

6. Filtering and state rules

The skin updates text and pseudo classes whenever text, excluded items, warning threshold or character range changes. Filtering runs with `Platform.runLater` so it happens after the current edit event.

Rule	Result
No range or max <= 0	Progress is zero and warning / error pseudo classes are false.
Length below min	:error and outOfRange=true.
Length at max * threshold	:warning unless the field is already in error.
Length above max	:error; progress indicator is hidden.

```
area.setCharacterRangeLimit(new IntegerRange(10, 120));
area.setWarningThreshold(0.85);
area.getExcludedItems().addAll(" ", " {,}");
```

Careful. Excluded items are joined with `|` and passed to `String.replaceAll`, so escape regular-expression metacharacters when you want literal matching.

7. Styling

7.1 Style classes and pseudo classes

Selector	Description
<code>.limited-text-area</code>	root control
<code>.limited-text-area:warning</code>	text length at or above warning threshold
<code>.limited-text-area:error</code>	text length below min or above max
<code>.content-box</code>	VBox assembled by the skin
<code>.bottom-box</code>	bottom status bar
<code>.length-label</code>	remaining / count label

7.2 Styleable CSS properties

CSS property	Type	Default
<code>-fx-show-bottom</code>	boolean	true
<code>-fx-warning-threshold</code>	number	0.9
<code>-fx-length-display-mode</code>	LengthDisplayMode	AUTO

```
.limited-text-area {  
    -fx-show-bottom: true;  
    -fx-warning-threshold: .8;  
    -fx-length-display-mode: always-show;  
}  
  
.limited-text-area:error > .content-box > .content-pane {  
    -fx-background-color: #ff5700, -fx-text-box-border, -fx-control-inner-background;  
}
```

8. Accessibility

The constructor sets `AccessibleRole.TEXT_AREA`. The visual length indicator is not automatically announced, so bind your own accessible help text if remaining count is critical to the workflow.

9. Recipes

9.1 Disallow line breaks

```
area.getExcludedItems().setAll("\r", "\n");
```

9.2 Hide the bottom bar

```
area.setShowBottom(false);
```

9.3 Warn earlier

```
area.setWarningThreshold(0.75);
```

9.4 Checklist

- 1 Remember that limits style the field; they do not prevent typing.
- 2 Treat `excludedItems` values as regular-expression fragments.
- 3 Observe `outOfRange` before saving.
- 4 Use inherited resize settings when layout must be fixed.

10. See also

- Demo application: `com.dlsc.gemsfx.demo.LimitedTextAreaApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.LimitedTextAreaApp`)
- `ResizableTextArea` - superclass providing the resize corner.
- `ExpandingTextArea` - automatic height text area.
- API documentation: <https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/>