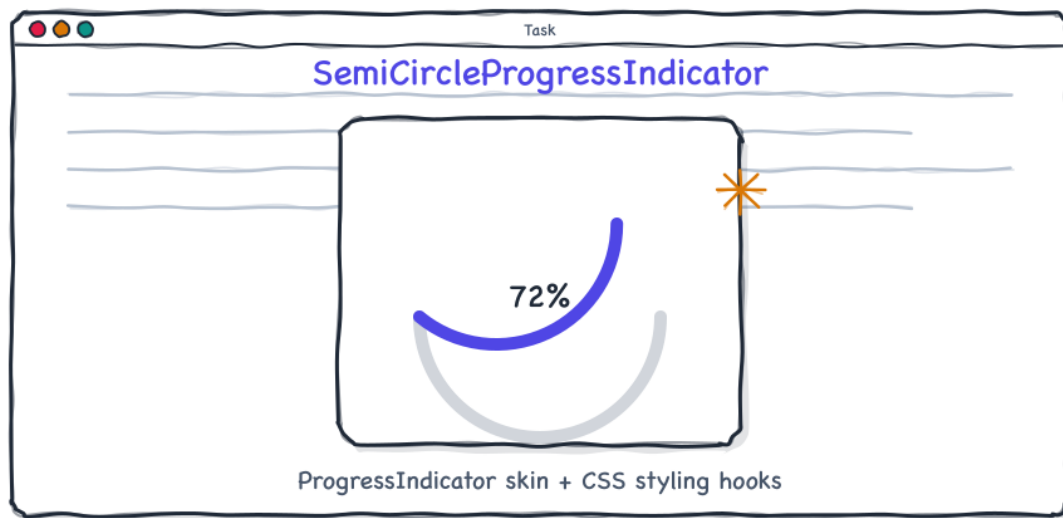


SemiCircleProgressIndicator

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



A semi-circular progress indicator with inherited arc progress behaviour.

SemiCircleProgressIndicator is a concrete ArcProgressIndicator that renders a 180 degree track with the label above the arc center. It supports the same converter, graphic, style type, CSS and accessibility behaviour as the circular variant.

Table of Contents

1. Introduction	2
2. Getting started	2
3. Anatomy	3
4. Control API	4
5. Behaviour and states	4
6. Layout and sizing	5
7. Styling with CSS	5
8. Localization	6
9. Accessibility	6
10. Recipes and demo	7
10.1 Use the demo style combo	7
10.2 Domain-specific label	7
11. Troubleshooting	8

1. Introduction

SemiCircleProgressIndicator extends **ArcProgressIndicator** and uses **SemiCircleProgressIndicatorSkin**. It is suited to dashboard cards, headers and other layouts where a full circle would consume too much vertical space.

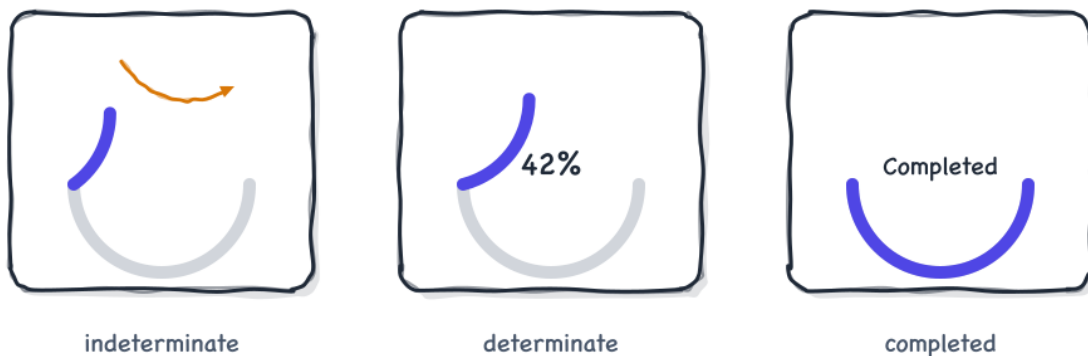
- Default constructor starts indeterminate.
- The **SemiCircleProgressIndicator(double)** constructor sets initial progress.
- Minimum size is 26 x 26 pixels.
- Root style classes are **arc-progress-indicator** and **semi-circle-progress-indicator**.

```
SemiCircleProgressIndicator indicator = new SemiCircleProgressIndicator(0.0);
indicator.setProgress(0.60);
indicator.setStyleType(ArcProgressIndicator.StyleType.DEFAULT);
```

2. Getting started

The API is the same as for the circular variant, except there is no **startAngle** property. The half circle always uses a 180 degree track.

```
Task<Void> task = createTask();
SemiCircleProgressIndicator indicator = new SemiCircleProgressIndicator();
indicator.progressProperty().bind(task.progressProperty());
new Thread(task, "import-task").start();
```

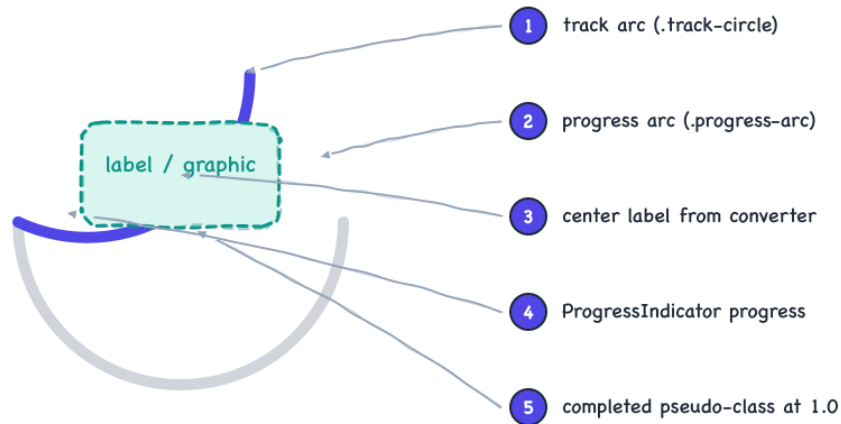


progress < 0 starts the timeline; progress == 1.0 enables :completed

The semi-circle in indeterminate, determinate and completed states.

Note. The default converter is inherited from **ArcProgressIndicator**, so completed progress shows the localized Completed text.

3. Anatomy



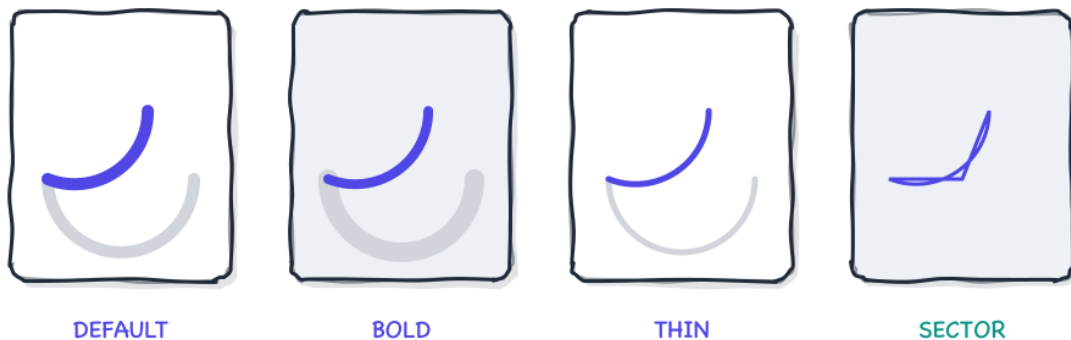
The parts rendered by SemiCircleProgressIndicatorSkin.

Part	Node	Description
Track	Arc	Start angle 0, length 180.
Progress	Arc	Foreground arc; length is $-180 * \text{progress}$.
Label	Label	Positioned above the arc center and constrained to half the diameter in height.
Animation	Timeline	Animates start angle $180 \rightarrow 90 \rightarrow 0$ and length $0 \rightarrow -60 \rightarrow 0$.

4. Control API

Property	Type	Default	Description
converter	ObjectProperty<String Converter<Double>>	percent converter	Inherited. Converts progress to text for the label.
graphic	ObjectProperty<Node>	null	Inherited. Optional graphic displayed by the label.
progressArcType	ObjectProperty<ArcType>	OPEN	Inherited and styleable.
trackArcType	ObjectProperty<ArcType>	CHORD	Inherited and styleable.
styleType	ObjectProperty<StyleType>	DEFAULT	Inherited. DEFAULT, BOLD, THIN or SECTOR.

5. Behaviour and states

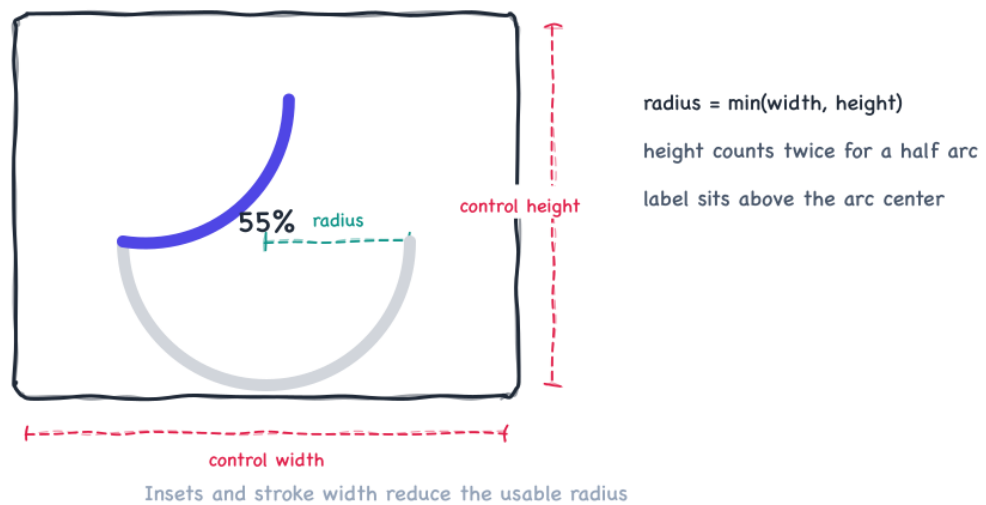


styleType toggles :bold-style, :thin-style and :sector-style

The same style types are available for the semi-circle.

State	Condition	Skin behaviour
indeterminate	progress < 0	Timeline sweeps a short arc across the half-circle.
determinate	0.0 .. 1.0	Progress length is -180 times the progress value.
completed	progress == 1.0	The :completed pseudo-class is active.

6. Layout and sizing



The half-circle radius uses width and twice the available height.

The radius binding uses the smaller of available width and twice the available height, then divides by two. The arc center is below the vertical center so the visible half-circle fits the content area.

```
// Useful in compact cards:
indicator.setPrefSize(180, 90);
indicator.setMaxSize(Region.USE_PREF_SIZE, Region.USE_PREF_SIZE);
```

7. Styling with CSS

Selector	Effect
.semi-circle-progress-indicator	Root class loaded from semi-circle-progress-indicator.css.
.track-circle	Light grey track.
.progress-arc	Foreground arc uses -fx-accent.
:bold-style	Track 10px, progress 5px.
:thin-style	Track and progress 1px.
:sector-style	Uses ROUND progress arc type and filled sector styling.
:completed	Hook when progress is exactly 1.0.

CSS property	Type	Default
-fx-style-type	StyleType	DEFAULT
-fx-progress-arc-type	ArcType	OPEN
-fx-track-arc-type	ArcType	CHORD

```
.semi-circle-progress-indicator {  
    -fx-style-type: bold;  
}  
  
.semi-circle-progress-indicator .track-circle {  
    -fx-stroke: #e5e7eb;  
}
```

8. Localization

SemiCircleProgressIndicator inherits the arc-progress-indicator resource bundle.

Key	English text
status.completed	Completed
accessible.text.loading	loading
accessible.text.percent	{0} percent

9. Accessibility

The inherited constructor sets `AccessibleRole.PROGRESS_INDICATOR`. Accessible text follows progress as localized loading text or a rounded percentage until the application sets its own accessible text.

10. Recipes and demo

10.1 Use the demo style combo

```
ComboBox<ArcProgressIndicator.StyleType> styles = new ComboBox<>();
styles.getItems().addAll(ArcProgressIndicator.StyleType.values());
styles.valueProperty().bindBidirectional(indicator.styleTypeProperty());
```

10.2 Domain-specific label

```
indicator.setConverter(new StringConverter<>() {
    @Override public String toString(Double p) {
        return p == null || p < 0 ? "Connecting" : String.format("%.0f%%", p * 100);
    }
    @Override public Double fromString(String text) { return null; }
});
```

- 1 Choose the semi-circle when vertical space is limited.
- 2 Reserve enough width for the label.
- 3 Use `styleType` for broad visual changes, CSS selectors for precise colors.

11. Troubleshooting

- There is no `startAngle` property on this control.
- If the label is clipped, increase the control height or use a shorter converter string.
- Use exactly `1.0` to enable `:completed` styling.