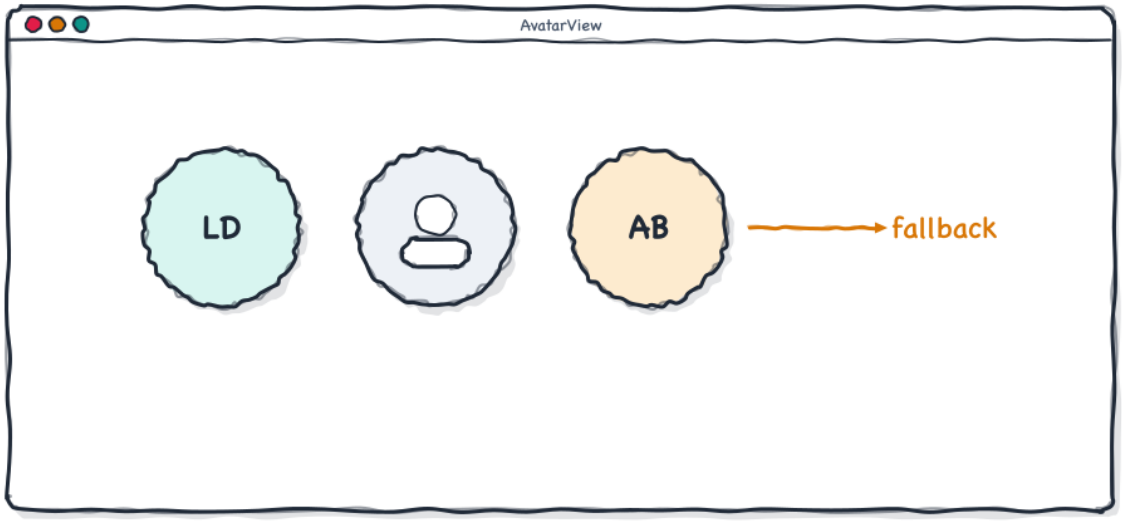
# AvatarView

## Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls




*Generated cartoon overview of AvatarView.*

AvatarView is an image-oriented control for user avatars. It displays a loaded image when available, falls back to initials, and finally shows a default person icon.

# Table of Contents

# 1. Introduction

**AvatarView** AvatarView is an image-oriented control for user avatars. It displays a loaded image when available, falls back to initials, and finally shows a default person icon.

## 1.1 Key features

- Fallback chain: loaded image, then initials, then a default icon.
- Initials produce deterministic style classes style0 through style4 by default.
- AvatarShape.SQUARE is the default; AvatarShape.ROUND clips to a circle.
- size defaults to 50 and is bound to pref/min/max width and height.
- arcSize defaults to 10 and rounds square avatars.
- Accessible text is localized and changes with initials.

## 1.2 Maven dependency

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

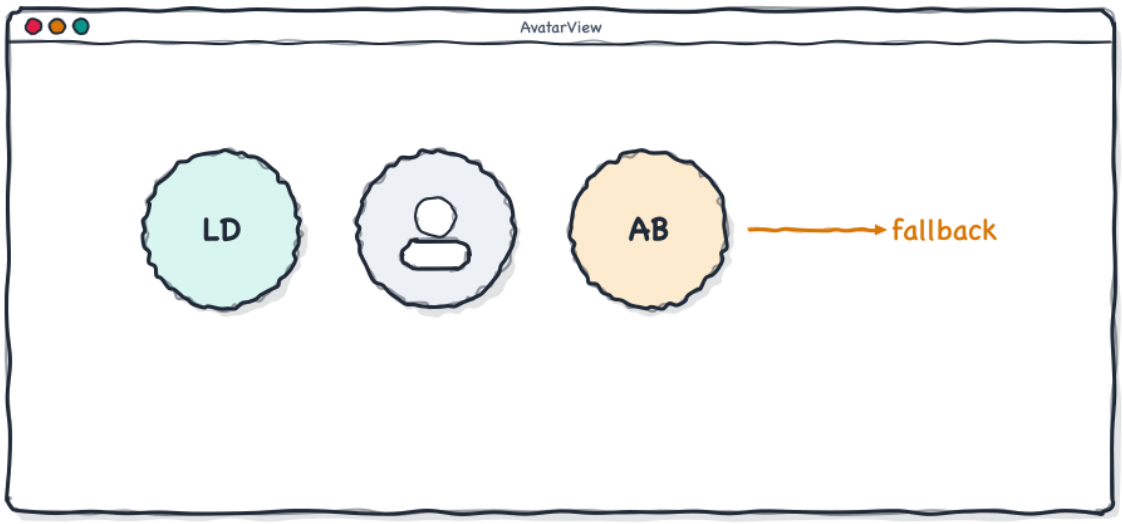
*The control lives in module* `com.dlsc.gemsfx`.

# 2. Getting started

The following snippet uses only public API verified in the control source.

```
AvatarView avatar = new AvatarView("LD");
avatar.setImage(userImage);          // image wins once loaded
avatar.setAvatarShape(AvatarShape.ROUND);
avatar.setSize(96);
avatar.setArcSize(18);              // used only for square avatars
```

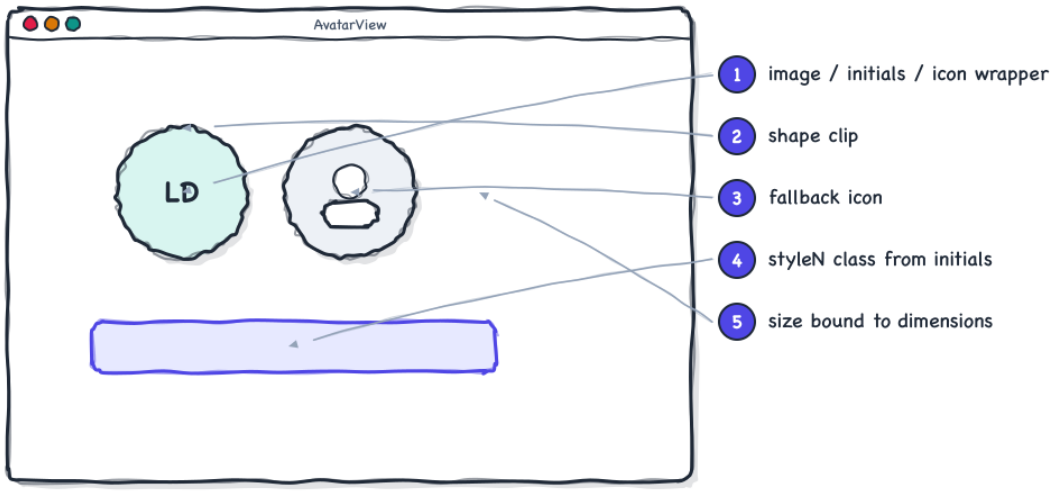
*A compact setup for AvatarView.*




*A generated overview of AvatarView in use.*

## 3. Anatomy

The anatomy diagram identifies the implementation pieces that matter when configuring, styling or debugging the control.




*The parts of AvatarView.*

| Part | Verified detail |
| --- | --- |
| Root | Style class `.avatar-view` is added by the constructor. |
| Stylesheet | User-agent stylesheet `avatar-view.css` is returned by the control. |

# 4. Control API

## 4.1 Properties and callbacks

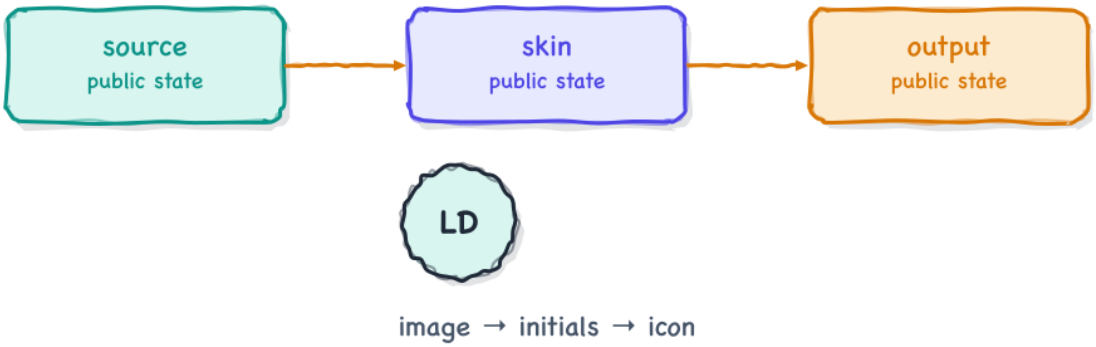
| Property | Type | Default | Description |
|---|---|---|---|
| initials | StringProperty | null | Initials shown when no fully-loaded image is available. |
| image | ObjectProperty<Image> | null | Avatar image; shown once loaded. |
| numberOfStyles | IntegerProperty | 5 | Modulo base for style0 ... styleN classes. |
| arcSize | DoubleProperty | 10 | Corner arc for square avatars; styleable. |
| size | DoubleProperty | 50 | Avatar width and height in pixels; styleable. |
| avatarShape | ObjectProperty<AvatarShape> | SQUARE | SQUARE or ROUND clipping; styleable. |
| magicNumber | IntegerProperty | -1, private | Derived from initials and used to select a style class. |

**Note.** Defaults and property names in this table were checked against the Java source for this batch.

# 5. Behaviour


*Important runtime states of AvatarView.*

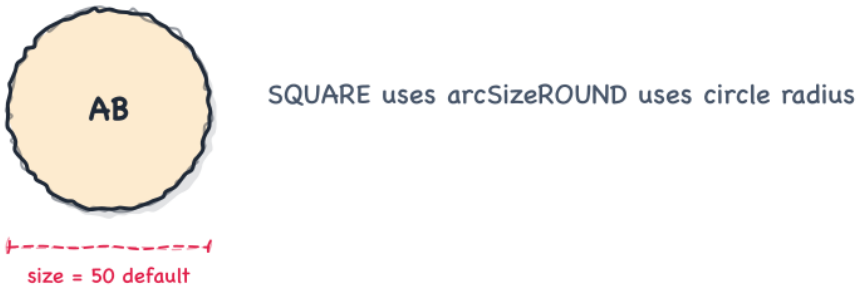
- AvatarViewSkin shows imageWrapper if image is non-null and fully loaded.
- If no loaded image exists and initials are not blank, textWrapper is shown.
- If both image and initials are absent, iconWrapper is shown.
- Changing initials recalculates the magic number and replaces style classes with avatar-view plus one styleN class.
- All three wrapper panes receive clips based on avatarShape.




*How data and geometry flow through AvatarView.*

## 6. Layout and rendering

Rendering is a simple fallback chain. A loaded image fills the avatar; otherwise initials are painted; otherwise a CSS-shaped icon is shown. Each visible wrapper is clipped to ROUND or SQUARE geometry.

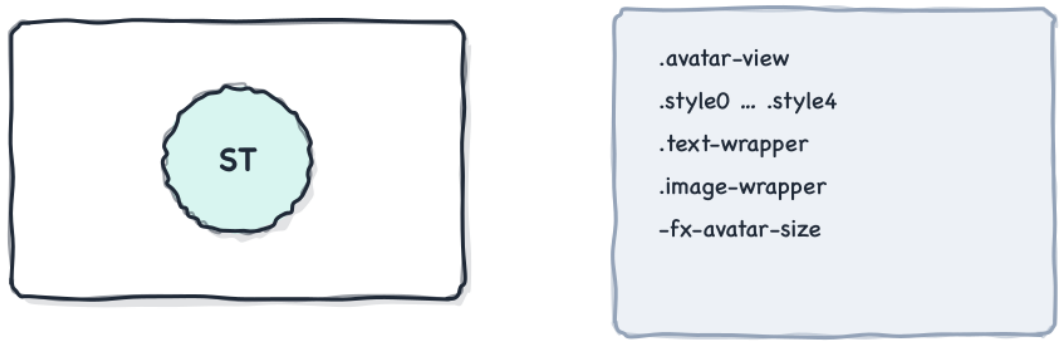



*Rendering and sizing rules for AvatarView.*

| Concern | Rule |
|---|---|
| Size | pref/min/max width and height are bound to size. |
| Shape | ROUND uses a Circle clip; SQUARE uses a Rectangle clip with arcSize. |
| Image scaling | ImageView scale is size divided by the smaller image dimension. |

# 7. Styling

The user-agent stylesheet is avatar-view.css. The table lists selectors and pseudo classes that exist in source, skin or stylesheet.




*Style hooks for AvatarView.*

## 7.1 Style classes and pseudo classes

| Selector / pseudo class | Purpose |
|---|---|
| `.avatar-view` | Verified in source, skin or CSS. |
| `.avatar-view.style0 ... .style4` | Verified in source, skin or CSS. |
| `> .icon-wrapper` | Verified in source, skin or CSS. |
| `> .icon-wrapper .icon` | Verified in source, skin or CSS. |
| `> .text-wrapper` | Verified in source, skin or CSS. |
| `> .text-wrapper > .initials-text` | Verified in source, skin or CSS. |
| `> .image-wrapper` | Verified in source, skin or CSS. |
| `> .wrapper-stack-pane` | Verified in source, skin or CSS. |

## 7.2 Styleable CSS properties

| CSS property | Type | Default |
|---|---|---|
| `-fx-avatar-shape` | AvatarShape | SQUARE |
| `-fx-avatar-arc-size` | size | 10 |
| `-fx-avatar-size` | size | 50 |

```
.avatar-view {
    /* start with the documented root selector */
}
```

*CSS example using documented hooks.*

# 8. Localization

| Key | English text |
|---|---|
| `accessible.text.avatar` | avatar |
| `accessible.text.avatar-of` | avatar of {0} |

# 9. Accessibility

AvatarView sets AccessibleRole.IMAGE_VIEW. AccessibilityUtil.bindAccessibleText uses localized accessible.text.avatar when initials are blank, otherwise formats accessible.text.avatar-of with the initials, for example "avatar of LD".

# 10. Recipes

### 10.1 Programmatic configuration

```
AvatarView avatar = new AvatarView("LD");
avatar.setImage(userImage);          // image wins once loaded
avatar.setAvatarShape(AvatarShape.ROUND);
avatar.setSize(96);
avatar.setArcSize(18);              // used only for square avatars
```

### 10.2 Practical checklist

1. Set initials as a fallback even when an image is usually present.
2. Keep numberOfStyles aligned with CSS styleN rules.
3. Use the public properties listed in the API chapter.
4. Style only through documented selectors and styleable properties.
5. Do not depend on private skin node structure except for documented CSS selectors.

# 11. Integration notes

No pseudo classes are set by the source.

| Topic | Recommendation |
|---|---|
| Threading | Keep image loading and expensive rendering off the UI path when the control exposes a background option. |
| Styling | Scope selectors under the documented root style class. |
| Accessibility | Preserve the source-defined accessible role and add app-specific text when the control does not bind it. |
| State | Prefer public properties over skin node lookup. |

## 12. See also

- Demo application: `com.dlsc.gemsfx.demo.AvatarViewApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.AvatarViewApp`)
- Related GemsFX media controls: `AvatarView`, `PhotoView`, `SVGImageView`, `BeforeAfterView`, `MaskedView`, `ScreensView`.
- API documentation: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/