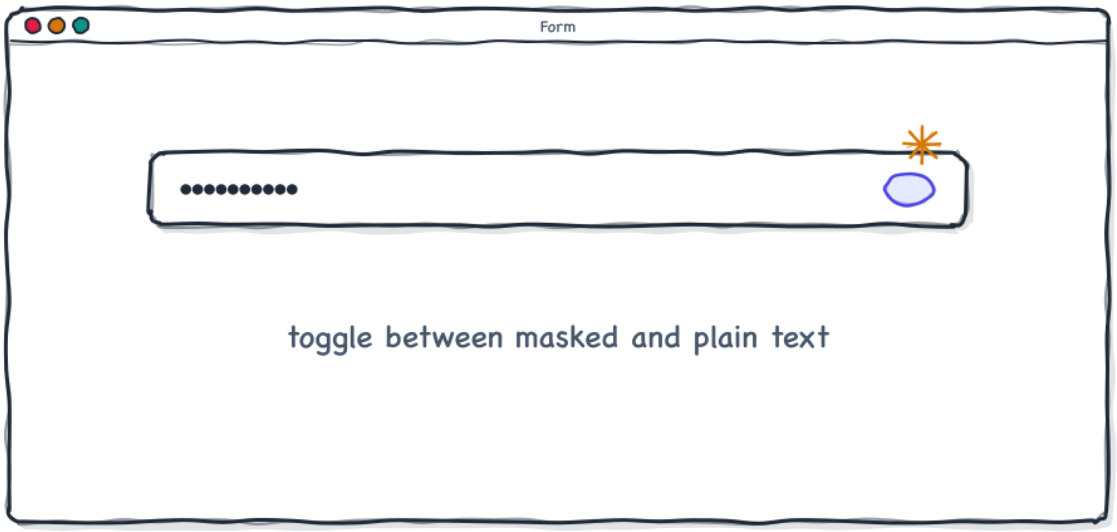
EnhancedPasswordField

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



A generated cartoon overview of EnhancedPasswordField.

EnhancedPasswordField extends PasswordField with left and right embedded nodes, a clickable default eye icon, a show-password state and a styleable echo character.

Table of Contents

1. Introduction

EnhancedPasswordField extends JavaFX `PasswordField`. It keeps the normal text-field API, adds slots for left and right nodes and replaces the skin text binding so the displayed string can be either masked or plain.

1.1 Key features

- Default clickable eye icon toggles `showPassword`.
- Styleable `echoChar` property, default ●.
- Optional application-supplied left and right nodes.
- Root pseudo class `:showing-password` while text is visible.

1.2 Maven dependency

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

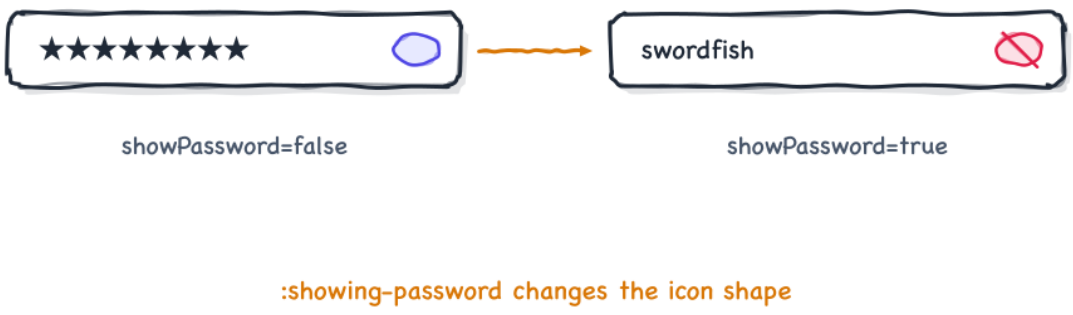
Maven coordinates for the GemsFX control library.

2. Getting started

```
EnhancedPasswordField field = new EnhancedPasswordField();
field.setPromptText("Password");
field.setEchoChar('★');
field.setLeft(new FontIcon(MaterialDesign.MDI_KEY));

field.showPasswordProperty().addListener((obs, old, showing) -> {
    System.out.println(showing ? "plain" : "masked");
});
```

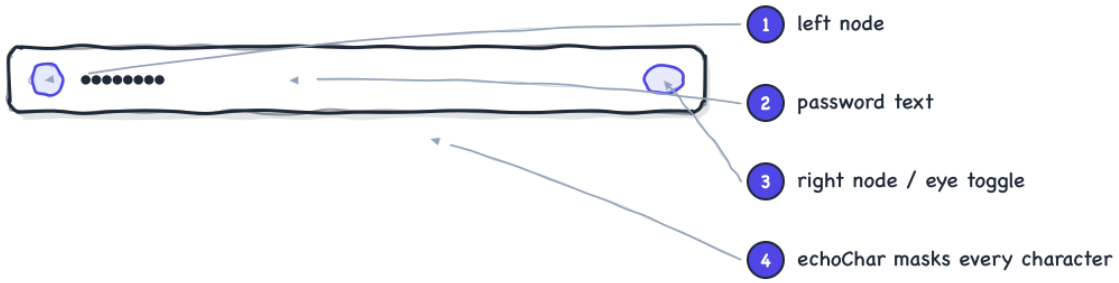
A complete setup with a custom echo character and left node.



Masked and plain-text display states.

3. Anatomy

The control uses `CustomTextFieldSkin` semantics for left and right nodes, then the specialized skin rebinds the internal text node.



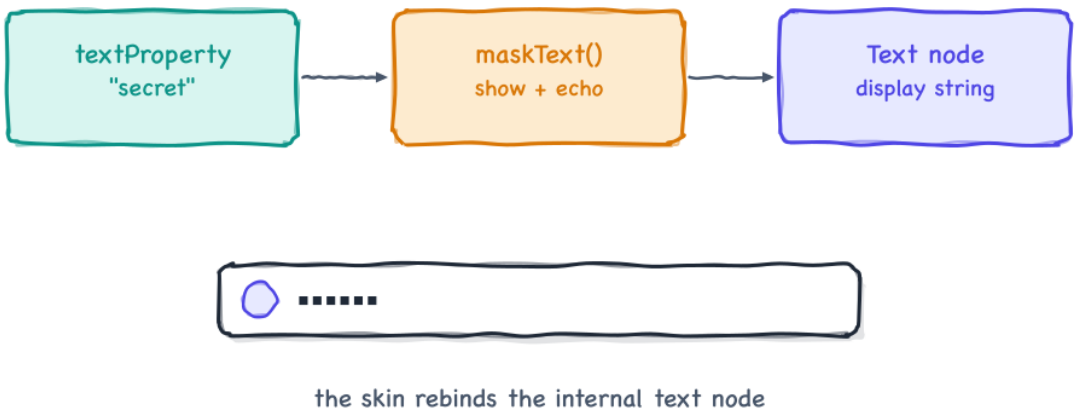
Parts of an EnhancedPasswordField.

Part	Node / property	Description
Root	`.enhanced-password-field`	PasswordField subclass with user agent stylesheet.
Left slot	`left`	Optional node supplied by the application.
Text node	internal `Text`	Shows repeated echo characters or plain text.
Right slot	`right`	Defaults to the clickable eye wrapper.

4. Control API

Property	Type	Default	Description
left	ObjectProperty<Node>	null	Node shown on the left side of the field.
right	ObjectProperty<Node>	eye icon wrapper	Node shown on the right side. The default toggles password visibility.
showPassword	BooleanProperty	false	Shows plain text when true.
echoChar	ObjectProperty<Character>	●	Masking character. Styleable via `-fx-echo-char`.
text	StringProperty	empty string	Inherited PasswordField content.

5. Masking behaviour



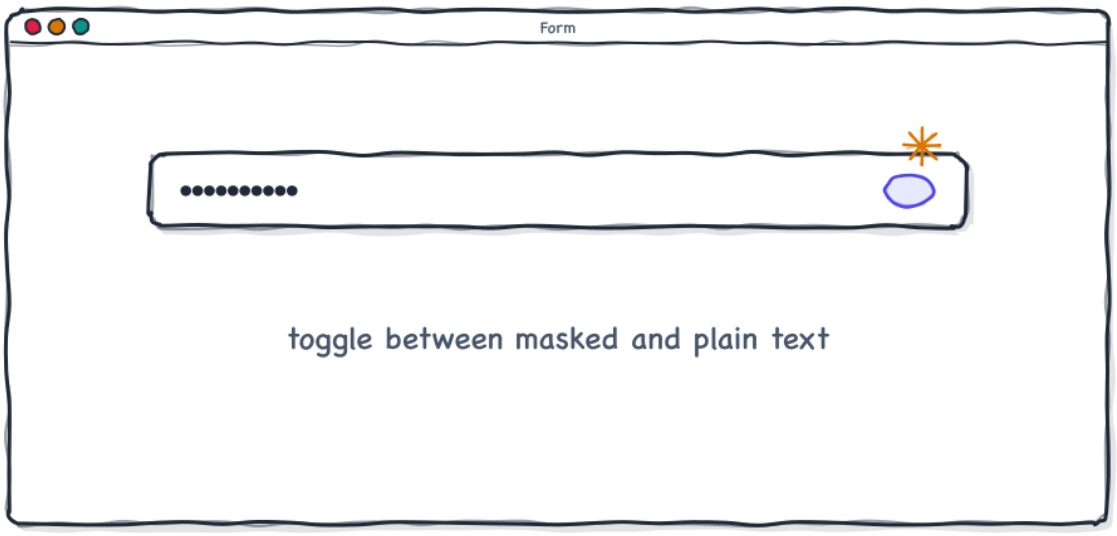
How the skin computes the displayed text.

The skin locates the real text node created by the standard text-field skin, unbinds its text and binds it to `text`, `showPassword` and `echoChar`.

`getEchoCharSafe()` returns the default character when the property is unset, null or cannot be converted from CSS.

```
field.setShowPassword(false);  // displays one echo char per character
field.setShowPassword(true);   // displays field.getText()
```

6. Side nodes and reveal interaction



The default right node toggles the reveal state.

The constructor installs a `StackPane` right node containing a `Region` with style class `right-icon`. A mouse click on that wrapper toggles `showPassword` when `UIUtil.isClickOnNode(event)` reports a click on the node.

Customization	Effect
Set `left`	Adds an application node before the text, for example a key icon.
Set `right`	Replaces the built-in eye toggle completely.
Bind `showPassword`	Lets an external checkbox or button drive reveal state.

```
CheckBox reveal = new CheckBox("Show password");
field.showPasswordProperty().bind(reveal.selectedProperty());
```

Tip. If you replace the right node, the built-in click-to-toggle handler is gone. Recreate that behaviour explicitly if the field should still reveal text.

7. Styling

7.1 Style classes and pseudo classes

Selector	Description
`.enhanced-password-field`	root field
`.enhanced-password-field:showing-password`	root while plain text is visible
`.right-icon-wrapper`	default clickable wrapper
`.right-icon`	eye / eye-off icon region

7.2 Styleable CSS properties

CSS property	Type	Default
`-fx-echo-char`	char	●

```
.enhanced-password-field {
    -fx-echo-char: '*';
}

.enhanced-password-field:showing-password {
    -fx-background-color: #fff8dc;
}
```

8. Accessibility

The constructor sets `AccessibleRole.PASSWORD_FIELD`. The control does not bind a generated accessible text, so applications can provide their own accessible text or description if needed.

9. Recipes

9.1 Replace the right node

```
Button clear = new Button("Clear");
clear.setOnAction(evt -> field.clear());
field.setRight(clear);
```

9.2 Start visible

```
EnhancedPasswordField field = new EnhancedPasswordField("temporary");
field.setShowPassword(true);
```

9.3 Use CSS for the echo character

```
field.setStyle("-fx-echo-char: '■';");
```

9.4 Checklist

1. Avoid logging `getText()` from password fields.
2. Keep the right node keyboard-accessible if replacing the default eye.
3. Use `showPassword` only for deliberate reveal interactions.

10. See also

- Demo application: `com.dlsc.gemsfx.demo.EnhancedPasswordFieldApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.EnhancedPasswordFieldApp`)
- `CustomTextField` - base class for embedded left and right nodes.
- `EmailField` - another field with embedded icons.
- API documentation: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/