13.7 Additional JavaFX Capabilities

This section overviews various additional JavaFX capabilities that are available in JavaFX 8 and JavaFX 9.

`TableView` Control

Section 13.5 demonstrated how to bind data to a `ListView` control. You often load such data from a database (Chapter 24, Accessing Databases with JDBC, and Chapter 29, Java Persistence API (JPA)). JavaFX's `TableView` control (package `javafx.scene.control`) displays tabular data in rows and columns, and supports user interactions with that data.

Accessibility

8

In a Java SE 8 update, JavaFX added *accessibility* features to help people with visual impairments use their devices. For example, the screen readers in various operating systems can speak screen text or text that you provide to help users with

visual impairments understand the purpose of a control. Visually impaired users must enable their operating systems' screen-reading capabilities. JavaFX controls also support:

- GUI navigation via the keyboard—for example, the user can press the *Tab* key to jump from one control to the next. If a screen reader also is enabled, as the user moves the focus from control to control, the screen reader will speak appropriate information about each control (discussed below).
- A high-contrast mode to make controls more readable—as with screen readers, visually impaired users must enable this feature in their operating systems.

See your operating system's documentation for information on enabling its screen reader and high-contrast mode.

Every JavaFX `Node` subclass also has the following accessibility-related properties:

- `accessibleTextProperty`—A `String` that a screen reader speaks for a control. For example, a screen reader normally speaks the text displayed on a `Button`, but setting this property for a `Button` causes the screen reader to speak this property's text instead. You also can set this property to provide accessibility text for controls that do not have text, such as `ImageView`s.
- `accessibleHelpProperty`—A more detailed control description `String` than that provided by the `accessibleTextProperty`. This property's text should help the user understand the purpose of the control in the context of your app.
- `accessibleRoleProperty`—A value from the `enum` `AccessibleRole` (package `javafx.scene`). A screen reader uses this property value to determine the attributes and actions supported for a given control.
- `accessibleRoleDescriptionProperty`—A `String` text description of a control that a screen reader typically speaks followed by the control's contents (such as the text on a `Button`) or the value of the

`accessibleTextProperty`.

In addition, you can add `Label`s to a GUI that describe other controls. In such cases, you should set each `Label`'s `labelFor` property to the specific control the `Label` describes. For example, a `TextField` in which the user can enter a phone number might be preceded by a `Label` containing the text `"Phone Number"`. If the `Label`'s `labelFor` property references the `TextField`, then a screen reader will read the `Label`'s text as well when describing the `TextField` to the user.

Third-Party JavaFX Libraries

JavaFX continues to become more popular. There are various open-source, third-party libraries, which define additional JavaFX capabilities that you can incorporate into your own apps. Some popular JavaFX libraries include:

- ControlsFX (`http://www.controlsfx.org/`) provides common dialogs, additional controls, validation capabilities, `TextField` enhancements, a `SpreadSheetView`, `TableView` enhancements and more. You can find the API documentation at `http://docs.controlsfx.org/` and various code samples at `http://code.controlsfx.org`. We use one of the open-source ControlsFX dialogs in Chapter 22.
- JFXtras (`http://jfxtras.org/`) also provides many additional JavaFX controls, including date/time pickers, controls for maintaining an agenda, a calendar control, additional window features and more.

- Medusa provides many JavaFX gauges that look like clocks, speedometers and more. You can view samples at `https://github.com/HanSolo/Medusa/blob/master/README.md`.

Creating Custom JavaFX Controls

You can create custom controls by extending existing JavaFX control classes to customize them or by extending JavaFX's `Control` class directly.

JavaFXPorts: JavaFX for Mobile and Embedded Devices

A key Java benefit is writing apps that can run on any device with a Java Virtual Machine (JVM), including notebook computers, desktop computers, servers, mobile devices and embedded devices (such as those used in the Internet of Things). Oracle officially supports JavaFX only for desktop apps. Gluon's open-source JavaFXPorts project brings the desktip version of JavaFX to mobile devices (iOS and Android) and devices like the inexpensive Raspberry Pi (`https://www.raspberrypi.org/`), which can be used as a standalone computer or for embedded-device applications. For more information on JavaFXPorts, visit

```
http://javafxports.org/
```

In addition, Gluon Mobile provides a mobile-optimized JavaFX implementation for iOS and Android. For more information, see

```
http://gluonhq.com/products/mobile/
```

Scenic View for Debugging JavaFX Scenes and Nodes

Scenic View is a debugging tool for JavaFX scenes and nodes. You embed **Scenic View** directly into your apps or run it as a standalone app. You can inspect your JavaFX scenes and nodes, and modify them dynamically to see how changes affect their presentation on the screen—without having to edit your code, recompile it and re-run it for each change. For more information, visit

```
http://www.scenic-view.org
```

JavaFX Resources and JavaFX in the Real World

Visit

```
http://bit.ly/JavaFXResources
```

for a lengthy and growing list of JavaFX resources that includes links to:

- articles
- tutorials (free and for purchase)
- key blogs and websites
- YouTube® videos
- books (for purchase)
- many libraries, tools, projects and frameworks
- slide shows from JavaFX presentations and
- various real-world examples of JavaFX in use.