Oracle® JavaFX

JavaFX Scene Builder User Guide

Release 1.1

E25449-03

October 2013

ORACLE®

Oracle JavaFX/JavaFX Scene Builder 1.0 Developer Release

E25449-03

Primary Author: Cindy Castillo

Contributor: Yves Joan

Contents

Part I

About This User Guide

This guide introduces you to the JavaFX Scene Builder user interface (UI). JavaFX Scene Builder (Scene Builder) is a design tool for the JavaFX platform. You drag and drop UI components to a Content panel, and the FXML code for the layout that you are creating is automatically generated in the background. FXML is an XML-based declarative markup language for defining the user interface in a JavaFX application. To learn more about FXML, read `Getting Started with FXML`.

This document assumes that you have already installed the Scene Builder product on your system and have also downloaded the samples bundle file that is provided at the same download site. See the `JavaFX Scene Builder Installation Guide` for more detailed information.

The following topics describe the different JavaFX Scene Builder features. You can also access the topics by expanding the table of contents on the right side of the HTML version or on the left side for PDF version of this document.

1

Starting Up Scene Builder

This chapter describes how to start JavaFX Scene Builder and how to open an existing FXML file, such as the IssueTrackingLite sample provided with the downloadable Scene Builder samples.

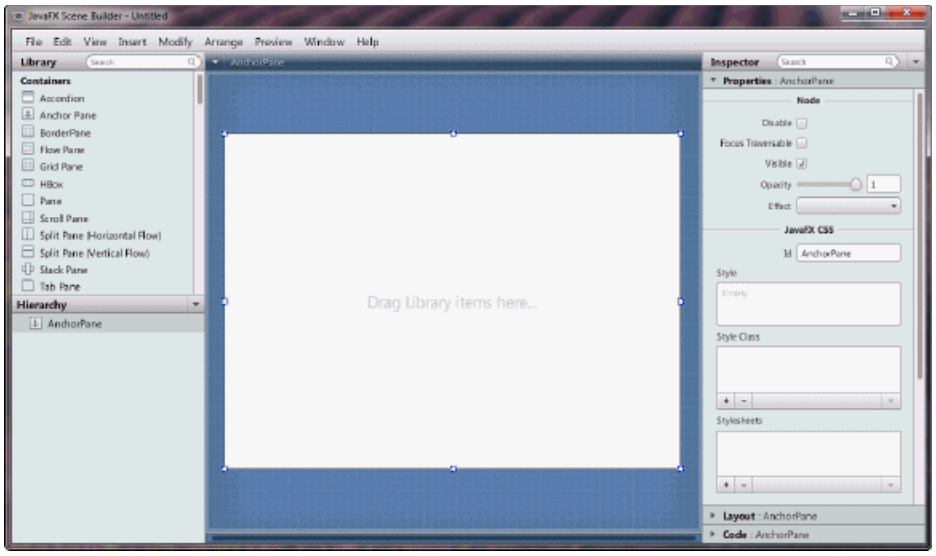
You start a standalone version of JavaFX Scene Builder by using one of the following methods:

- On the Windows platform, do one of the following steps:
  - Double-click the **JavaFX Scene Builder 1.1 desktop icon.**
  - Select **Start**, then **All Program**, then **JavaFX Scene Builder**, and finally, **JavaFX Scene Builder 1.1.**
- On the Mac OS X platform, double-click the J**avaFX Scene Builder 1.1** shortcut in the Applications folder.

The initial window that you see is the main window, as shown in Figure 1–1. It displays an untitled FXML file that contains the root AnchorPane element.

An element is a principal visual object that you can add to the Content panel and manipulate when editing an application's UI layout in JavaFX Scene Builder. An element can be dragged onto the Content panel from the Library panel of predefined UI elements. You can group elements by selecting multiple elements in the Content panel or the Hierarchy panel, and selecting Arrange and then the Wrap In command from the Menu bar to select the specific JavaFX container that you want to use.

Figure 1–1 Startup Window (Click image to enlarge.)



From within Scene Builder, you can also open an existing FXML document from the Menu bar using one of the following steps:

- From the **File** menu, select **Open.**
- From the **File** menu, select **Open Recent** and select the file from the list of recently opened FXML files.

For illustration purposes, use the `IssueTrackingLite.fxml` sample file that is included with the JavaFX Scene Builder samples bundle.

1. Go to where you extracted the `javafx_scenebuilder_samples-1_1.zip` file.
2. Expand the `IssueTrackingLite`, `src`, and `issuetrackinglite` folders.
3. Open the `IssueTrackingLite.fxml` file.
   The file is opened in Scene Builder, if you have Scene Builder properly installed.

You can also use Scene Builder with one of the Java IDEs, such as NetBeans IDE, Eclipse, or IntelliJ IDEA, to create a new FXML file or open an existing one. Use the information in `Using JavaFX Scene Builder with Java IDEs` to learn more.

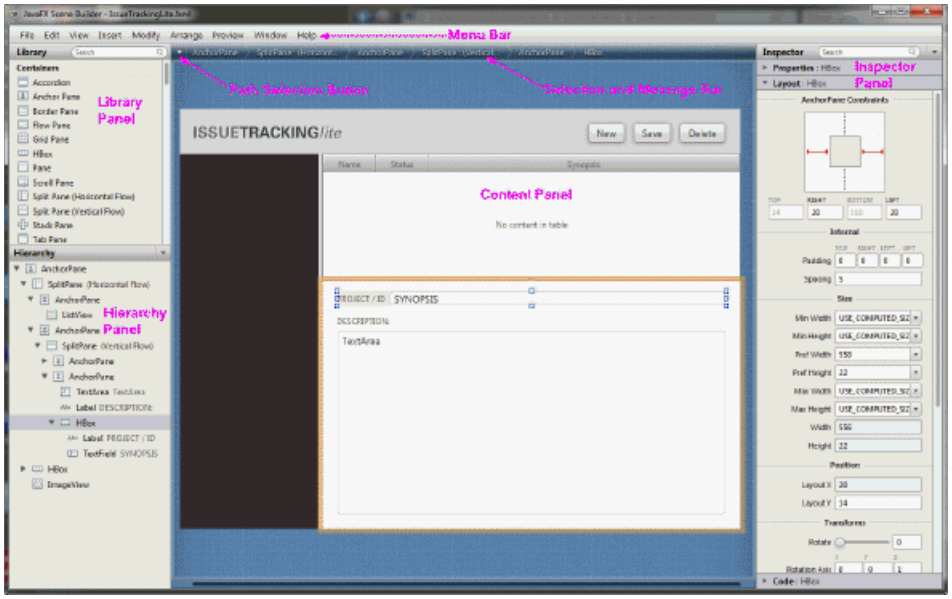
2

Getting to Know the Main Window

This chapter discusses the different major parts of the main window that appears after Scene Builder is started up.

After you open the FXML file to work with, it's contents are displayed in the Content panel of the main window, similar to what is shown in Figure 2–1. Click the image to see a larger version.

Figure 2–1 Main Window of JavaFX Scene Builder (Click image to enlarge.)



By default, the main window of JavaFX Scene Builder includes the following sections, which are labeled in Figure 2–1.

- **Menu Bar**: Provides access to the menu of commands available in JavaFX Scene Builder.
- **Selection and Message Bar**: Displays the path to a selected element. It also displays any error or status messages.
- **Content Panel**: The scene container for the UI elements that make up your FXML layout. By default, a new FXML file that is opened in JavaFX Scene Builder includes a root (top) AnchorPane container.

- **Library Panel**: Lists the available JavaFX UI elements or controls that you can use to build your FXML layout. You select the UI elements from this panel and add them to the Content panel or the Hierarchy panel.
- **Hierarchy Panel**: Displays a tree view representation of the FXML layout that you are building in the Content panel. Elements that are not visible in the Content panel can be placed into focus by selecting it in the Hierarchy panel.
- **Inspector Panel**: Contains the Properties, Layout, and Code sections. The Properties and Layout sections help you manage the properties of the selected UI element in the Content panel or in the Hierarchy panel. The Code section enables you to manage the controller source information and event handling actions to use for the selected UI element. The Inspector panel also contains a Search text field that enables you to isolate specific properties that you want to modify.

The following panel is also displayed in the main window when you select **View** from the Main menu and then **Show CSS Analyzer.**

- **CSS Analyzer Panel**: Allows you to explore all the CSS properties available for a JavaFX component on your FXML layout and helps you to build the CSS rules

The next chapters describe the parts listed above in more detail.

3

Using the Menu Bar

This chapter describes the commands that are available from the Scene Builder main Menu bar.

The Menu bar displays the menus of commands that you can perform with an FXML layout file. The following sections describe the list of subcommands available when you select a command from the Menu bar. Table 3–1 describes the subcommands for the File command.

> **Note:** The shortcut key shown in the tables below are for Windows platform. Replace the Ctrl key with the Cmd key to use the equivalent shortcut key for the Mac OS X platform.

Table 3–1 List of File Subcommands

Command (Shortcut Key)	Description
New (Ctrl+N)	Create a new FXML file and opens it in a separate Scene Builder window.
New with Root Container	From a list of containers, choose the top level container to use when creating a new FXML layout. You can use an Anchor Pane, Border Pane, or a Stack Pane as the root container.
New from Template	Create a new FXML layout using one of the available templates: Alert Dialog, Basic Application, Complex Application, Alert Dialog - CSS, Alert Dialog - Localized, Basic Application - CSS, Basic Application - Localized, Complex Application - CSS, and Complex Application - Localized layouts.
Open (Ctrl+O)	From an Open FXML dialog box, choose the existing FXML file to open.
Open Recent	Choose the file to open from a list of recently opened FXML files.
Preferences (Ctrl+,)	Set the preferred values to use for certain properties in the tool. You can change the default values for the Content panel's width and height, the image to use for the tool's background image, the colors to use for the alignment guides and target rings, whether to automatically generate the Id for a newly added element, the default display mode to use for the Hierarchy panel, and the order in which the CSS Analyzer columns are displayed.
Close (Ctrl+W)	Close the current window. If you have any unsaved changes, you are prompted to save the file first before the window is closed. All other active windows remain open until they are closed or you exit the tool.

Table 3–1 (Cont.) List of File Subcommands

Command (Shortcut Key)	Description
Save (Ctrl+S)	Save the changes made since the last Save action was performed. If it is a new FXML document, you are prompted to provide the new file name. If it is an existing FXML file, the current file name is retained.
Save As (Ctrl+Shift+S)	Save the current FXML file you are working on, with the option to save the changes to a different file.
Reveal in Explorer (Windows), Reveal in Linux, Reveal in Finder (Mac OS X)	Open a native file browser, which displays the folder containing the current FXML file.
Revert to Saved	Undo changes made since the last Save action was performed.
Import	Import an existing image, FXML, or media file. You should import an image, media or FXML file into Scene Builder from inside your project sources directory. If the file you are importing is located somewhere other than your project sources directory, you need to move or copy the file. Scene Builder does not copy the imported file for you. It makes a link to the imported file using a path relative to the location of the FXML file currently being worked on. Consequently, if the relative location of the imported file is different at runtime, then your FXML file may not load properly. You need to ensure all imported files are packaged properly by your IDE and that their relative location at runtime and edit time are the same. The simplest way to achieve this is to only import files which are already located under your project sources directory and to instruct your IDE to include these files in the packaging, which is usually done by default.
Exit	Exit the tool. All opened Scene Builder windows are closed. You are prompted to save any unsaved changes.

Selecting the **Edit** command menu displays the subcommands that enable you to choose the **Undo** or **Redo** command on the last action that you made in the current FXML layout. In addition, you can choose the **Cut**, **Copy**, **Paste**, **Paste Into, Duplicate**, and **Delete** commands that you can apply to the selected elements in the Content or Hierarchy panel. The Cut, Copy, and Paste commands use the native editor and clipboard, unless a text field element is selected. This means that when you copy or cut an element and paste it on a text editor, what is actually copied and pasted is the FXML code that defines a Group element that wraps the selected element and all its children elements. If the focus is on a text field element then the select, copy, and paste commands are performed from and to the text field element. With the Paste Into command, you need to first select where you want to paste an item into before selecting the Paste Into command.

Choosing the **Select All** command selects all of the elements in your FXML layout, while the **Select None** command deselects any selected elements. The **Select Parent** command selects the parent container of the selected element.

The **Trim Document to Selection** command deletes all of the elements up to the selected row in the Hierarchy panel. When you choose the Trim Document to Selection command and the selected element is for a container element, that selected container element becomes the new root node container of your FXML document.

Choosing the **Use "fx:root" Construct** command allows you to use the <fx:root> element, which creates a reference to a previously defined root element. To learn more about the <fx:root> element, refer to the `JavaFX API document`. The UnlockCustom

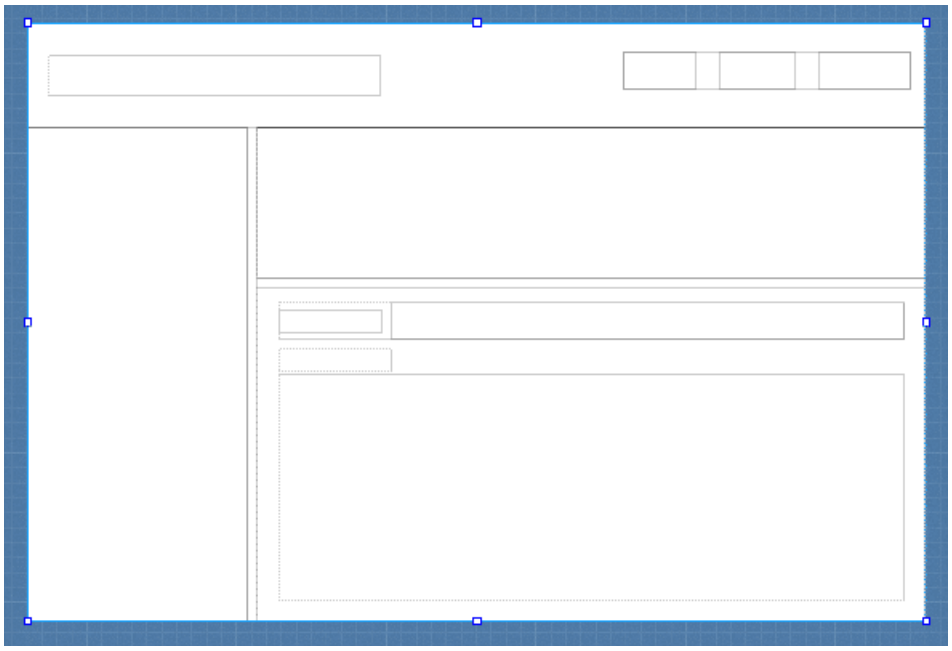
sample, which is included in the JavaFX Scene Builder samples bundle, illustrates the use of the <fx:root> element.

Table 3–2 describes the subcommands available when you select the **View** command from the Menu bar.

Table 3–2 List of View Subcommands

Command (Shortcut key)	Description
Content (Ctrl+0)	Place focus on the Content panel.
Properties (Ctrl+1)	Open the Properties section of the Inspector panel and place the current focus on it.
Layout (Ctrl+2)	Open the Layout section of the Inspector panel and place focus on it.
Code (Ctrl+3)	Open the Code section of the Inspector panel and place focus on it.
Hide/Show Library (Ctrl+4)	Hide or show the Library panel on the left side of the current window. If the Hierarchy panel is still visible, it automatically adjusts to occupy the entire left side of the window. If the Hierarchy panel is not visible, the Content panel adjusts to occupy the rest of the window's left side.
Hide/Show Hierarchy (Ctrl+5)	Hide or show the Hierarchy panel. If the Library panel is still visible, it is automatically adjusted to occupy the entire left side of the window. If the Library panel is not visible, the Content panel is automatically adjusted to occupy the rest of the window's left side.
Show/Hide CSS Analyzer (Ctrl+6)	Show or hide the CSS Analyzer panel. The panel appears at the bottom of the window.
Hide/Show Left Panel (Ctrl+7)	Hide or show the panels on the left side of the window. If both the Library and Hierarchy panels were visible when the Hide Left Panel command was selected, then both panels are displayed with the Show Left command. If only the Hierarchy panel was active when the Hide Left Panel command was executed, the Show Left Panel command redisplays the Hierarchy panel.
Hide/Show Right Panel (Ctrl+8)	Hide or show the Inspector Panel on the right side of the window.
Show/Hide Outlines (Ctrl+E)	Show or hide the outline of each element in your entire layout, as shown in Figure 3–1. The outline shows the LayoutBounds property of each element.
Show/Hide Sample Data	Show or hide demonstration data for those elements, such as a tree view, that have some sample data associated with them. The sample data is not saved in your FXML layout file.
Disable/Enable Alignment Guides	Disable or enable the alignment guidelines that help you align the elements with one another.
Show Sample Controller Skeleton	Open a window that displays sample skeleton code that you can use to create a controller class for the FXML layout you are building

Figure 3–1 Show Outlines



Choosing the **Insert** command from the Menu bar enables you to choose UI elements, such as containers, controls, menu content, shapes, charts, custom components, and miscellaneous elements, that you can add to the current FXML layout. These are the same UI elements available from the Library panel. If none of the existing UI elements in the layout is selected, the insertion point for the new element is the center of the Content panel. If one of the elements is selected before choosing the Insert command, the new element being added is placed slightly off to the side, but over the selected element. The Insert submenu displays only the valid UI controls that can be inserted to the selected element or container.

The subcommands displayed when you select **Modify** from the Menu bar are described in Table 3–3.

Table 3–3 List of Modify Subcommands

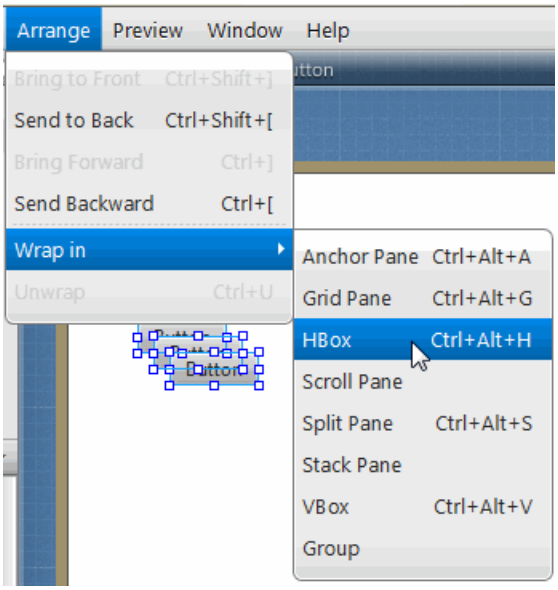
Command (Shortcut Key)	Description
Fit to Parent (Ctrl+K)	Resize a selected element to fill the area that its parent container occupies. The element is also anchored to each of the sides of the parent. This feature is only available when the parent container is an AnchorPane element.
Use Computed Sizes	Reset the selected element's layout property values to the default values. The USE_COMPUTED_SIZE value is used for the Min Width, Min Height, Pref Width, Pref Height, Max Width, or Max Height layout properties. (See the API documentation for JavaFX Region for more information.)
Grid Pane	Display all the subcommands available for working with the selected Grid Pane component in your layout. You can delete, move, add, or resize rows or columns in the selected Grid Pane.
Add Effect	Apply effects, such as a drop shadow or a reflection, to the selected element or group of elements in the layout.

Table 3–3 (Cont.) List of Modify Subcommands

Command (Shortcut Key)	Description
Add Popup Control	Add a Context Menu or a Tooltip element to the selected element in your layout. Note that the added pop-up control is only selectable in the Hierarchy panel.
Edit Included File	Edit any imported or included FXML file that is referenced by an `fx:include` statement defined for the selected element. When you edit an included FXML file, a new JavaFX Scene Builder window is opened. Changes you save in the included FXML file are reflected in the FXML layout file from which you opened the included FXML file.
Reveal Included File in Explorer	Open a native file browser to show the folder that contains the included file.

Use the **Arrange** menu, which is shown in Figure 3–2, to bring the selected element to the front, send it back, or move it forward or backward. Selecting the **Wrap In** subcommand places the selected elements into one of the available containers, such as an Anchor Pane, HBox, or a Grid Pane container. The **Unwrap** command removes the container for the selected elements, but leaves the UI elements it contains untouched.

Figure 3–2 Menu Bar with Arrange Menu Opened



The subcommands displayed when you choose the **Preview** command from the Menu Bar are described in Table 3–4.

Table 3–4 List of Preview Subcommands

Command (Shortcut Key)	Description
Show Preview in Window/Hide Preview in Window (Ctrl+P)	Display or hide the window that lets you preview the user interface design rendered by your current FXML code when it is deployed.
Refresh Preview in Window (Ctrl+R)	Refresh the layout that is displayed in the Preview window. This is useful when you make changes to your current layout.

Table 3–4 (Cont.) List of Preview Subcommands

Command (Shortcut Key)	Description
Choose Background Color	Display a color chooser dialog box in which you define the background color to use for your layout.
Scene Style Sheets	Open a submenu that displays commands that enable you to add a new style sheet that you can apply to your current layout, remove a style sheet from the list of style sheets being used, or open an existing style sheet in a separate editor window specified for the style sheet file type. The layout is immediately updated.
Internationalization	Enables you to set, remove, or reveal a resource file in the native file browser.
Resolve Unknown Types	This option is activated when the FXML document that is loaded in Scene Builder contains custom element types that were created by the user or element types that could not be loaded by the FXML loader. Choosing this command displays the same dialog box that appeared when the element type was first loaded. Use the dialog box to modify the classpath information. See Custom and Unknown Element Types for more information.

Selecting the **Window** menu gives you a list of FXML file names that are currently opened in JavaFX Scene Builder windows. When you select one of the file names in the list, the name is marked with a check mark, and its corresponding Scene Builder window becomes active.

Selecting **Help** and then **Scene Builder Help** opens your default web browser and displays the web site that contains the available JavaFX Scene Builder documentation. Select the **About Scene Builder** command to open a window that displays the release information for the JavaFX Scene Builder you are using.

4

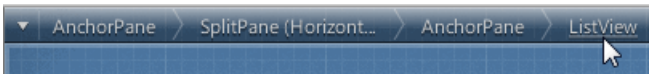
Using the Selection and Message Bar

This chapter describes the Selection and Message bar that is located just above the Content panel of the Scene Builder main window.

The Selection and Message bar is shown in Figure 4–1 and Figure 4–3. As the name indicates, it serves two purposes:

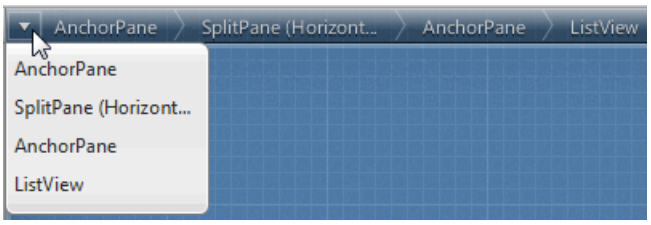
- It displays the containment path of the selected UI element. It enables you to easily go to and place focus on a specific element by clicking that element's name in the containment path. When you hover the cursor over an element's name, it is underlined, as shown in Figure 4–1.

Figure 4–1 Selection Bar



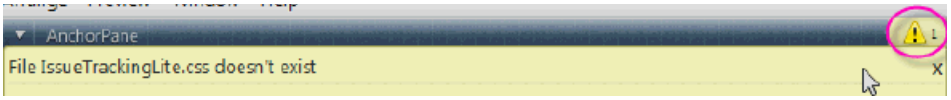
You can also select an element from the list that is displayed when you click the Path Selection button to the left of the containment path, as shown in Figure 4–2.

Figure 4–2 Path Selection Button



- The same area that is used to display the containment path is also used to display feedback and error messages, as shown in Figure 4–3. After a message is displayed for a couple of seconds, the warning icon remains on the right end of the Message bar. Click the warning icon, circled in Figure 4–3, to view the list of messages. The number next to the icon indicates the number of messages that are displayed. You can clear one or all of the messages in the list view.

Figure 4–3 Message Bar



5

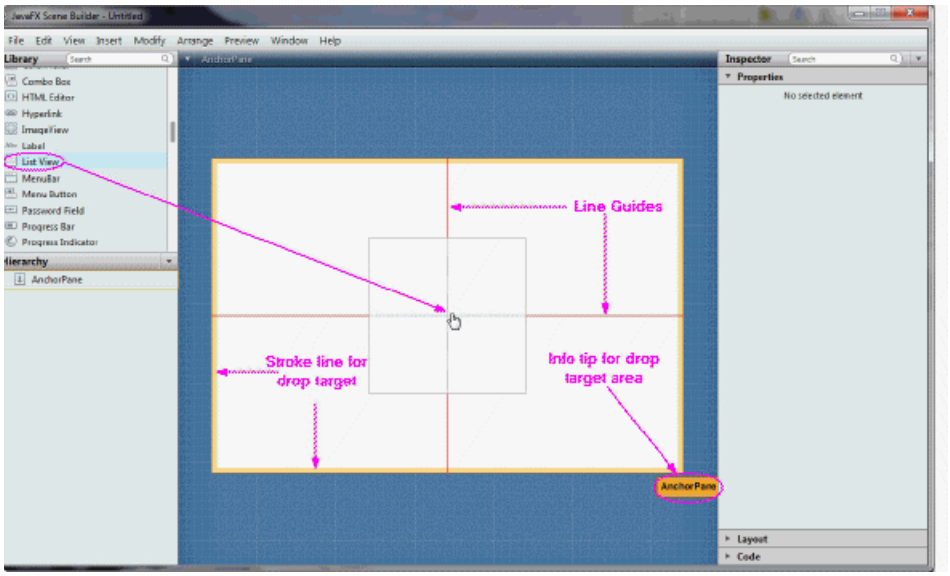
Working with the Content Panel

This chapter describes the Content panel of the JavaFX Scene Builder.

The Content panel is the rectangular area that occupies the center of the JavaFX Scene Builder window. It is your design canvas, and it contains a view of what you are designing. By default, when you create a new FXML file, the Content panel contains an AnchorPane element as the root container. This container element will contain the elements of the layout that you build.

The Content panel enables you to directly manipulate the graphical elements used in your FXML layout. One of the ways to add a UI element to the Content panel is to drag the chosen UI control from the Library panel and place them in a chosen location on the Content panel. As you drag an object onto the Content panel, the golden stroke lines appears around the drop target area, as shown in Figure 5–1. An info tip is also displayed at the bottom right corner to identify the drop target area. Guidelines help you align the component being dropped in relation to other UI elements that are already present in your FXML layout:

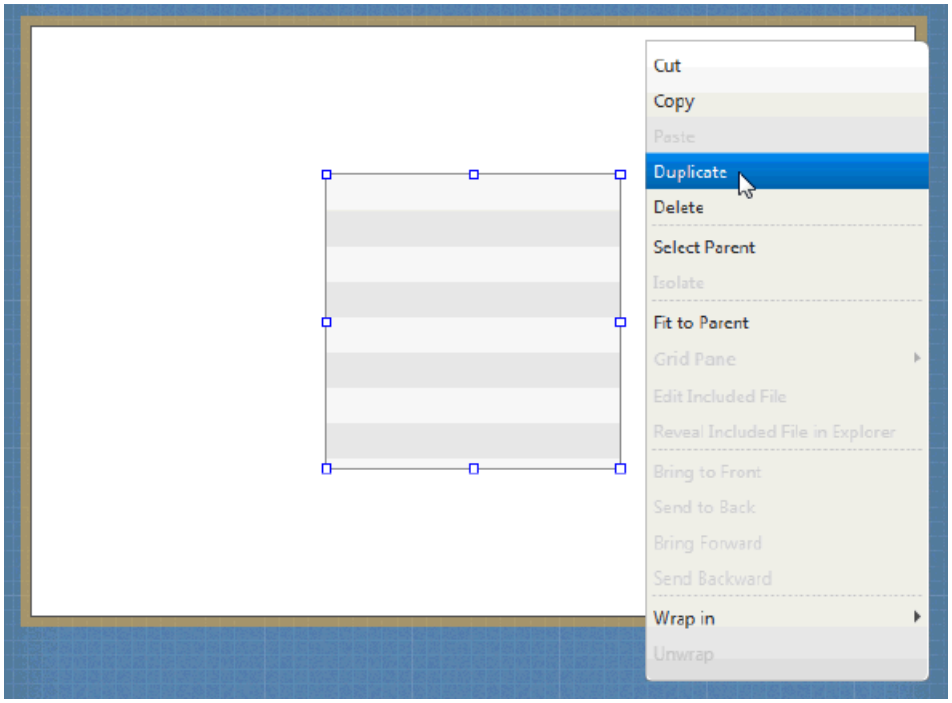
Figure 5–1 Dropping a Component onto the Content Panel (Click image to enlarge.)



The selected elements have special handles that enable you to scale or resize the elements. You move the selected element by using your keyboard's arrow keys or by using the mouse to drag it to a different location. Right-clicking anywhere in the

Content panel displays a contextual menu of commands that can be used on the currently selected element, as shown in Figure 5–2.

Figure 5–2 Contextual Menu in Content Panel



Double-clicking on a UI element in the Content panel places that element in an inline editing mode. If a UI element with a Text property is selected, pressing Return also places the element in an inline editing mode.

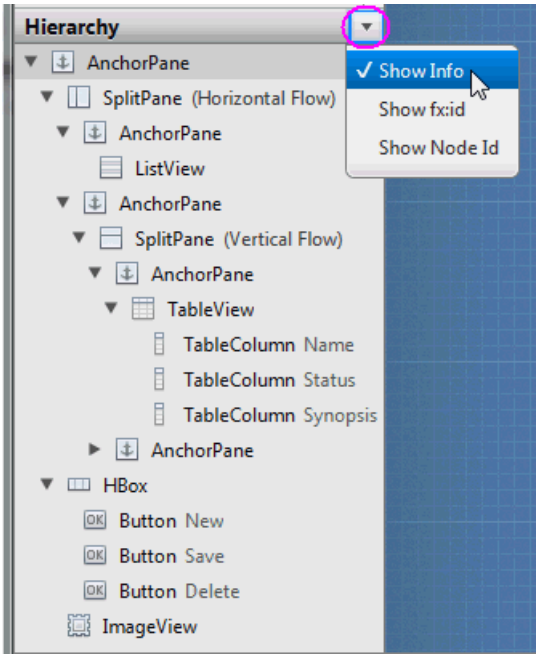
6

Using the Hierarchy Panel

This chapter describes the Hierarchy panel of JavaFX Scene Builder.

The Hierarchy panel, shown in Figure 6–1, is located on the lower left side of the tool's window. You can hide the Hierarchy panel by selecting **View** from the Main menu and then selecting **Hide Hierarchy**, or **Hide Left Side Panel** to hide it along with the Library panel. It shows all the UI elements that compose your FXML layout. You can use the Hierarchy panel to focus on one specific UI element, whether it is a parent node or a leaf node.

Figure 6–1 Hierarchy Panel



After the element is isolated, you can focus your work on that particular element and its properties by using the Content panel or the Inspector panel. The path from the root of the FXML layout to a particular UI element is displayed in the Selection bar when a UI element is isolated in the Hierarchy panel. The selected UI object is highlighted in the Hierarchy panel and in the Content panel.

You can add a new UI element to your FXML layout by dragging it from the Library panel to the desired location in the Hierarchy panel. This is especially helpful when adding a Popup control, such as a Tooltip or ContextMenu. For example, to add a

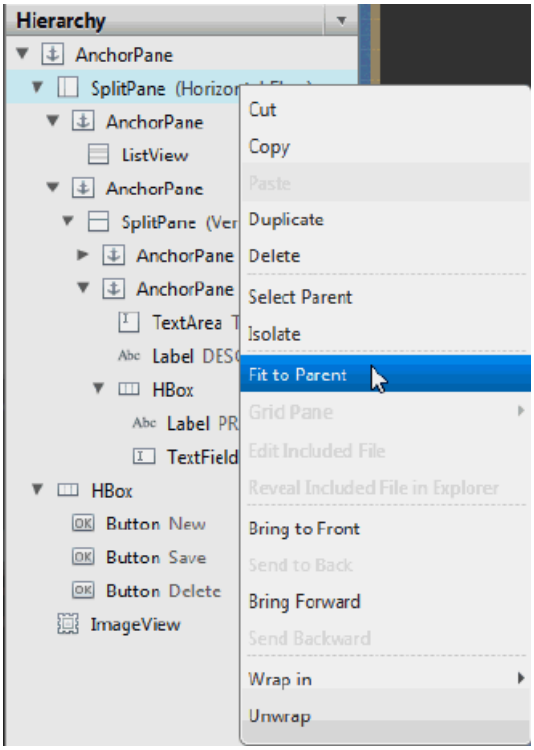
Tooltip to a Button element, drag the Tooltip control from the Library panel to the target Button element in the Hierarchy panel. The Tooltip becomes a child of that Button element. You can rearrange the order of the UI elements in your layout by dragging them in and out of containers within the Hierarchy panel.

The Hierarchy panel is also useful for adding graphic element to a UI element that has the `graphic` property, per its API. For example, you can add a graphic element to a Button element, but not to a GridPane element. Use the following steps to add a Circle graphic element to a Button element using the Hierarchy panel:

1. From the Library panel, drag a Circle element to the target Button layer in the Hierarchy panel.
2. Without releasing the mouse cursor, hover over the Button layer for a second and you will notice a new sub-layer is created for the Circle.
3. Release your mouse cursor and the Circle appears in the Content panel, with its default radius of size 100.
4. In the Layout section of the Inspector panel, reduce the circle's radius size to 6 so that it fits more appropriately in the Button.
   You can only have one graphic added to an element at any one time. To add a new graphic, you must first delete the existing one.

Right-clicking an element in the Content panel or a specific element layer in the Hierarchy panel displays a contextual menu of commands that you can use for the selected element, as shown in Figure 6–2. These commands are a subset of the commands that are available from the Menu Bar.

Figure 6–2 Hierarchy Panel with Drop-down List of Commands



When you click the drop-down menu, which is circled on the top right corner in Figure 6–1, you can choose to display the fx:id or the Node Id for each of the elements

listed in the panel. The default display mode is **Show Info**, which displays any related information for the element, or the text value if the element has a text property. When you select **Show Node Id**, each component level displays the same node names used in the Selection bar. The **Show fx:id** command displays the value of each element's fx:id property, which is used by the application controller class that handles the logic for the UI. If one of these latter two Show commands is chosen, click on the right side of the row for the element type to activate the inline editor so you can edit the Node Id or fx:id values for the UI element.

7

Designing UI with the Library Panel

This chapter describes Scene Builder's Library panel and the JavaFX UI controls that are accessible from it. Included in the chapter is information about some of the container elements available in the Library panel. Also discussed is how to handle custom and unknown UI types.

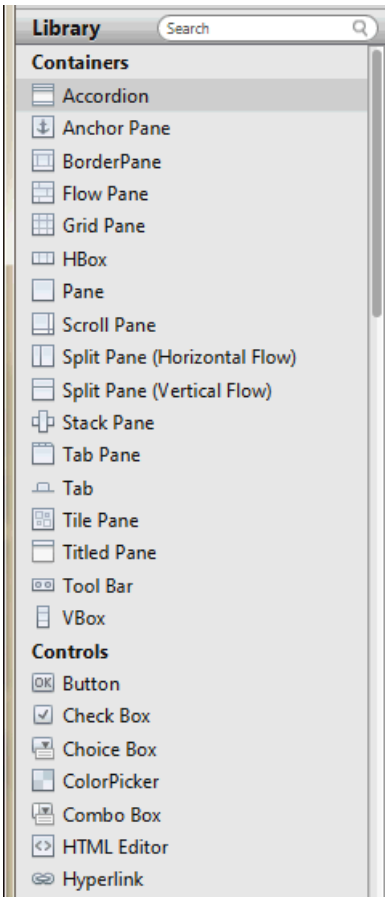
The Library panel, shown in Figure 7–1, is located on the left side of the Main window. You can hide the Library panel by selecting **View** from the Main menu and then selecting **Hide Library** or **Hide Left Side Panel** to hide it along with the Hierarchy panel. The Library panel includes the list of Containers, Controls, Popup Controls, Menu Content, Shapes, and Charts elements that are available in JavaFX Scene Builder. These UI elements are the same UI elements that are delivered with the JavaFX platform.

As mentioned earlier, when you open a new FXML layout in Scene Builder, it contains an AnchorPane as the default root container. You can use other root containers as well. You can add onto that root container as many containers or UI elements as needed for your FXML layout.

To add one of the UI elements from the Library panel, you select an item from the list on the panel and drag it onto the Content panel. You can also press the Enter key or double-click a selected item in the Library panel to add it to the Content panel. Either of these actions is equivalent to choosing the **Insert** command from the Menu bar, or choosing **Modify** and then **Add** command if the highlighted element in the Library is a Popup Control.

Use the Search field, located at the top right corner of the panel, to filter the list of elements available in the Library. For example, typing `button` in the Search field displays all the button type elements.

Figure 7–1 Library Panel



The following sections discuss some of the container elements that are available in the Library panel and how to maximize their usage in the JavaFX Scene Builder tool. Also described are custom and unknown types. To learn more about all the other UI elements offered with the JavaFX SDK platform, see the `JavaFX API documentation`.

Grid Pane Container

The Grid Pane container enables you to create a flexible layout of UI elements that are organized in rows and columns. Use the Grid Pane in your layout by selecting **Grid Pane** from the Container section of the Library panel and dragging it to the Content panel. You can then add other UI elements within the Grid Pane container, and elements are automatically arranged based on the padding, gap, and other properties.

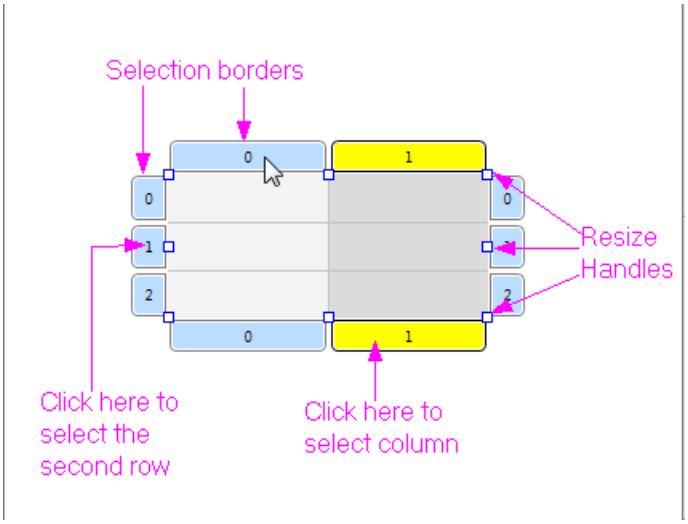
You can also add a Grid Pane container element so that it contains the UI elements that you have already added to your layout. Use the following steps:

1. Choose **Edit** and then **Select All** from the Main menu to select all of the elements in your layout.
2. Add the Grid Pane container by selecting **Arrange** from the Main menu, **Wrap in**, and then **Grid Pane** from the list of containers.

When an empty grid pane is not selected in the Content panel, it is not visible. To make its borders visible, click the corresponding layer in the Hierarchy panel for that Grid Pane. To work with a specific grid pane, click directly within its perimeter, but

not on any of the elements that it contains. When selected, a Grid Pane is surrounded by a light blue colored border tabs and each corresponding index number is displayed, as shown in Figure 7–2. You select either a single column or row by clicking within the selection border tab. The tabs of a selected column or row acquires a yellow background, as shown by column 1 in Figure 7–2.

Figure 7–2 Grid Pane in Selected Mode



To see the submenu of actions that you can take on a grid pane, do one of the following:

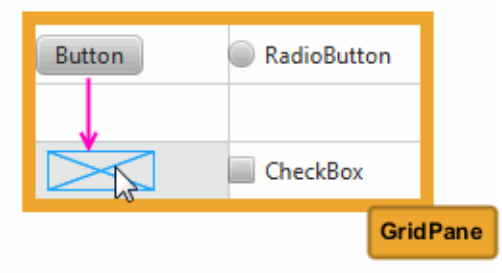
- Right-click anywhere in the Content panel, choose **Grid Pane** from the contextual menu.
- From the Main menu, choose **Modify** and then **Grid Pane**.

You can move a row up or down, and move a column left or right. You can also delete or add a row or column, and increase or decrease the span for a row or column.

The Layout section of the Inspector panel also enables you to modify the currently selected Grid Pane. The properties related to the selected row or column are displayed and can be managed in the GridPane Row or GridPane Column section.

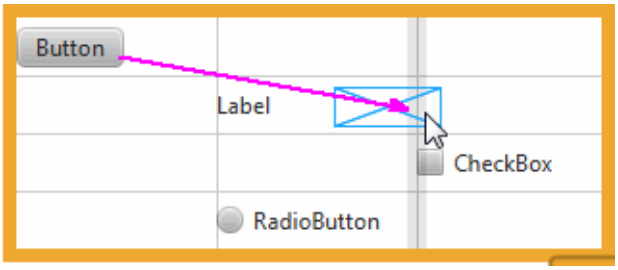
You can move an element from one grid cell to another existing grid cell. In the Content panel, select the element that you want to relocate, and with the mouse button pressed, drag the element to the target grid cell, as shown in Figure 7–3. Notice that the target grid cell changes to have a gray background as soon as you hover your cursor over it. When you release the mouse button, the element is moved to the target grid cell location in the Grid Pane.

Figure 7–3 Moving an Element to Another Grid Cell



To move an element to a new row or column that has not been created, select the element in the Content panel and drag it to the divider line that is between the rows or columns where you want the new row or column to be inserted. As you hover the mouse over the cell divider, the cell divider line becomes thicker and gray. When you release the cursor, the new row or column is created, and the element is dropped in the target grid cell. Figure 7–4 shows a Button element being dragged to a new grid cell on the second row of a new column that is about to be inserted.

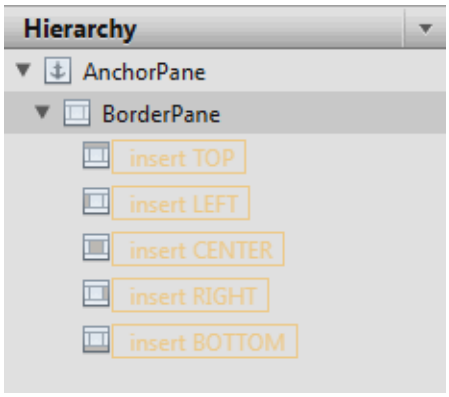
Figure 7–4 Moving an Element to a New Row or Column



Border Pane Container

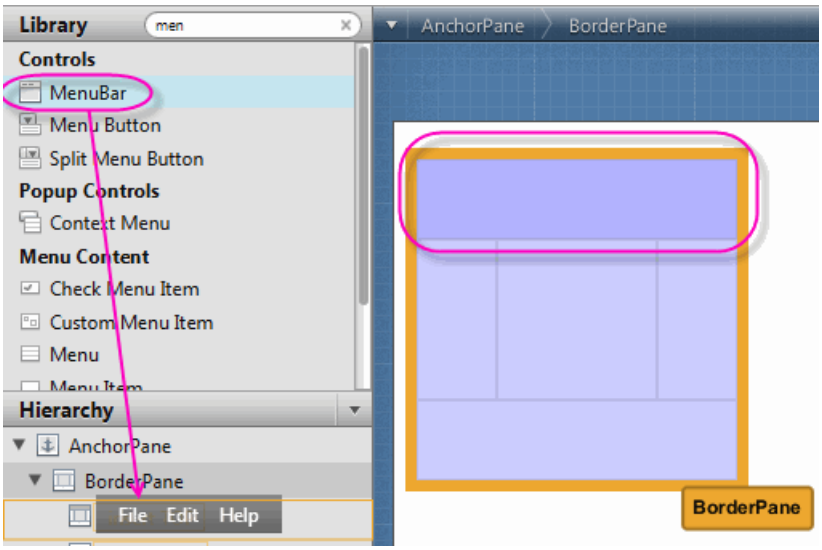
The Border Pane container enables you to layout UI elements in the top, right, left, bottom, and center positions of the container. Use the Border Pane in your layout by selecting the **BorderPane** element from the Container section of the Library panel and dragging it to the Content panel. Figure 7–5 shows the Hierarchy panel after the BorderPane element is added.

Figure 7–5 Hierarchy Panel After BorderPane Container Is Added



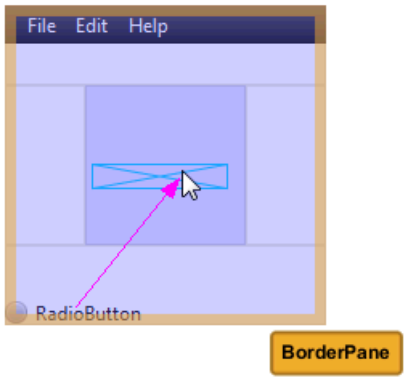
You can add a UI element to the Border Pane container by dragging it from the Library panel onto the specific position in the Border Pane. You can also drag an element from the Library panel to the Hierarchy panel, as shown in Figure 7–6. Notice that when the element is dragged over a target layer in the Hierarchy panel, the layer's corresponding position in the Border Pane changes to a darker shade in color, as shown on the right side of Figure 7–6.

Figure 7–6 Add Menu Bar Element to the BorderPane



You can move an element from one position to another. In the Content panel, select the element that you want to relocate and with the mouse button pressed, drag the element to the target position, as shown in Figure 7–7. Notice that the background color of the target position gets darker as soon as you hover your cursor over it. When you release the mouse button, the element is moved to the target position location in the Border Pane.

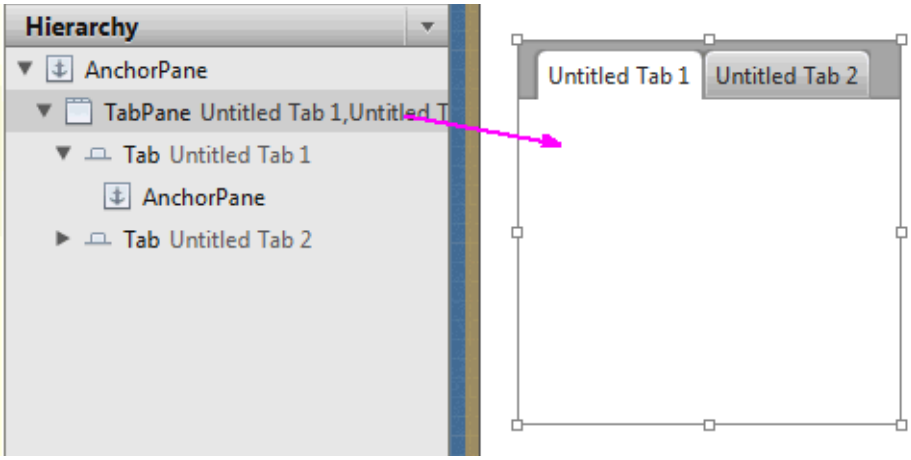
Figure 7–7 Moving an Element to Another Position



Tab Pane Container

The Tab Pane is one of the container elements available in the Library panel. By default, the Tab Pane comes with two Tab elements, which contain an Anchor Pane element each, as shown in Figure 7–8.

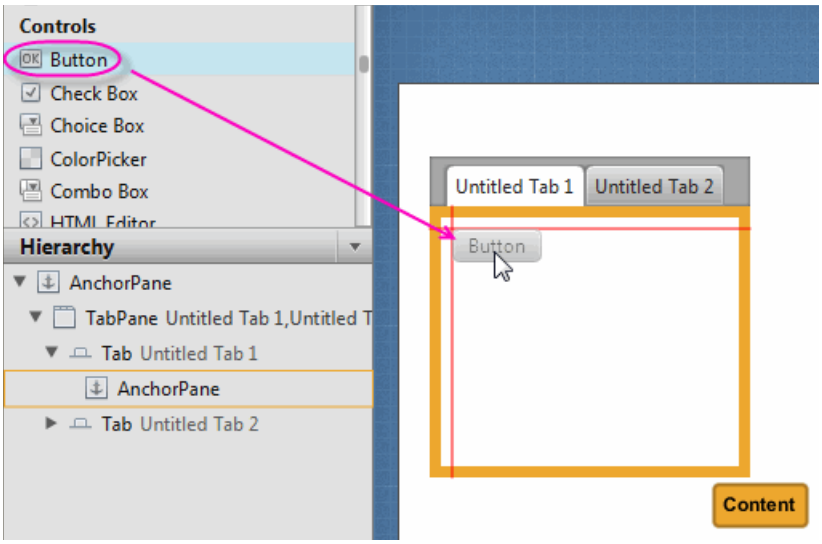
Figure 7–8 Tab Pane Element Added in the Hierarchy Panel and Content Panel



You can change a Tab element's default title by double-clicking the title and directly editing its Text property in the Content panel. To add another Tab element to a Tab Pane container, drag and drop a Tab element from the Library panel to the Tab Pane in the Content panel. The new Tab element is added to the right of any existing Tab elements. Use the Hierarchy panel to change the order of the Tab elements within the Tab Pane. Drag and drop the Tab element's row to the target location within the Tab Pane container in the Hierarchy panel.

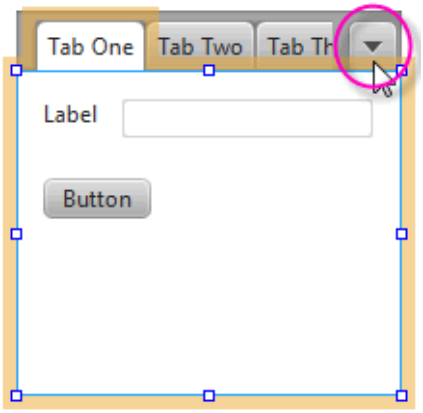
To add UI elements to a Tab element's Anchor Pane, click the title of the target Tab element to select it. Drag and drop the UI element that you want to add from the Library panel to the Tab element, as shown in Figure 7–9.

Figure 7–9 Add UI Elements from Library Panel to the Tab Element in Content Panel



Depending on the length of each Tab element's title and the number of Tab elements in your Tab Pane container, there comes a point when some Tabs become hidden. When this occurs, a drop-down menu is automatically added to the Tab Pane container's menu bar, as shown in Figure 7–10. This drop-down menu enables the tab of choice to be selected. This menu is not active in the Content panel, but is functional when you view the layout in the Preview window. If the Tab Pane container is resized so that all the Tab elements are fully visible, the drop-down menu is automatically removed from the Tab Pane's menu bar.

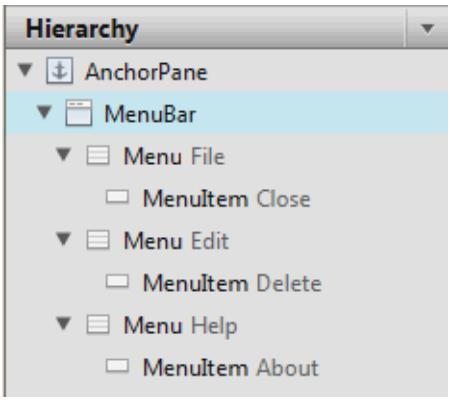
Figure 7–10 Tab Pane with Multiple Tab Elements and a Drop-Down Menu



Menu Controls

The Controls section of the Library panel contains three menu-related control elements: MenuBar, Menu Button, and Split Menu Button. When you drag and drop a MenuBar from the Library panel onto the Content panel, it comes populated with Menu elements that are labeled File, Edit, and Help. By default, each of these Menu elements contains one Menu Item element, as shown in Figure 7–11.

Figure 7–11 Menu Bar and Its Contents Displayed in the Hierarchy Panel



Because the Menu and Menu Item elements do not have graphical representations that are visible in the Content panel, you use the Hierarchy panel to select and manage these elements. You can change the order of how the Menu elements are displayed in the Menu bar by selecting the Menu element in the Hierarchy panel and dragging it to a new location within the Menu where you want it to appear. You can directly modify the text property of the Menu or Menu Item element by selecting it and clicking on the text area, which puts you into edit mode. You can also use the Properties section of the Inspector panel to modify the Text field.

You can add other menu-related items, such as another Menu Item or a Separator Menu Item, by dragging the element from the Library panel to the desired Menu location in the Hierarchy panel. To create a second-level menu, drop a Menu element within an already existing Menu element in the Hierarchy panel.

Use the Preview menu to see your Menu bar in action. From the Preview window, you can click a Menu item and any of its submenus. If your current platform supports it, you can choose the **Use System Menu Bar** property from the Properties section of the Inspector panel to make the Preview window display the menu bar as it would appear on your native system.

Custom and Unknown Element Types

A custom type element is any UI element type that is not part of the JRE 7 (Java with JavaFX) libraries of UI elements. It is a Java class that has been created by the user or by a third party. A custom element type is an "Unknown Type" until it is resolved. Choose the **Preview** command from the Menu bar and then the **Resolve Unknown Types** command to define the classpath to use for the custom element type.

You can also load an FMXL file that already contains your own custom UI controls, unless those UI controls require specific builder or builder factories. When you initially load your FXML file that contains element types that Scene Builder does not recognize, a dialog box appears enabling you to enter the classpath for the custom UI types.

In some cases, JavaFX Scene Builder is able to load a file that contains unknown element types even if you choose to not provide their classpaths. The unknown type will not be displayed in either the Control panel or the Hierarchy panel, but their FXML elements will be preserved.

If you provide a class path for your custom element types, Scene Builder should be able to render them in the Content panel and the Hierarchy panel. The properties that

are specific to the custom element type will appear in the Miscellaneous category of the Properties section of the Inspector panel. Since JavaFX Scene Builder does not know about your custom types, it might not be able to display their subnodes in the Hierarchy panel. You can also copy and paste a custom type to create another element of the same custom type.

Using Your Own Custom Type

If you want to add your own custom UI controls in your FXML layout, use the following suggested steps:

1. Create a skeleton FXML file by choosing the **File** and then **New** commands from the Menu bar in JavaFX Scene Builder.
2. At the place where you would like to insert your own custom UI control, insert an instance of its closest JavaFX super class as a place holder.
3. Choose **Save** from the Menu bar and close the FXML file.
4. Using NetBeans IDE, Eclipse IDE, or your favorite text editor, open the FXML file that you just created. Replace the name of the place holder JavaFX element with the name of your own custom class, and also add the appropriate `<?import?>` clause at the beginning of the FXML file.
5. In JavaFX Scene Builder, choose **File** and then **Open Recent** from the Menu bar to open your file again. In the dialog box, enter the class path name to your custom element type and save the change to continue loading the file.

8

Defining Properties in the Inspector Panel

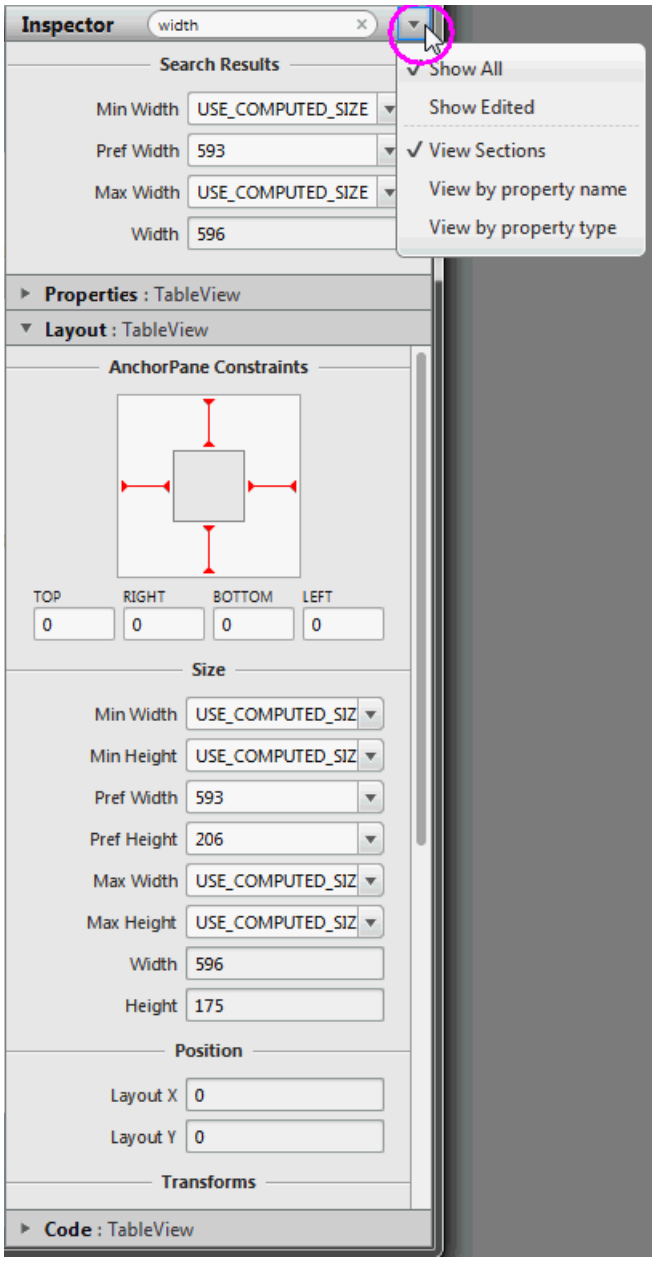
This chapter describes the JavaFX Scene Builder tool's Inspector panel, including its Properties, Layout, and Code sections.

On the right side of the Content panel is the Inspector panel, as shown in Figure 8–1 with the Layout section expanded and the Inspector menu displayed. The Inspector panel is also referred to as the Right Side panel. It contains an Accordion container with multiple sections. The top section contains a Search text field, which you can use to quickly locate the properties you would like to edit. The search results are displayed in the top section, as shown in Figure 8–1. You can directly modify the properties by using the Search Results section. You can display the Inspector menu by clicking the down arrow at the top right corner, which is circled in Figure 8–1. By default, the Show All and View Sections modes are chosen. Choosing the **Show Edited** mode displays only the properties that appears explicitly in the FXML file. For example, only the Text, LayoutX, and LayoutY properties are displayed in the Inspector panel when a Button element is dropped from the Library panel. Choosing the **View by property name** or the **View by property type** modes display all the properties in a single tabbed section, ordered either by name or type.

The Properties, Layout, and Code sections give you access to properties for the selected UI element. Each displayed property name shown in the sections is a representation of the corresponding given property name in the JavaFX API. For example, minWidth property in the JavaFX API is displayed in the Inspector panel as Min Width. When you hover over the property name, a tooltip displays the corresponding JavaFX API name for the property and its description.

You can hide the Inspector panel by selecting **View** from the Main menu and then selecting **Hide Right Side Panel**. This feature enables you to have more room for the Content panel. Select **View** and then **Show Right Side Panel** to make the panel visible again.

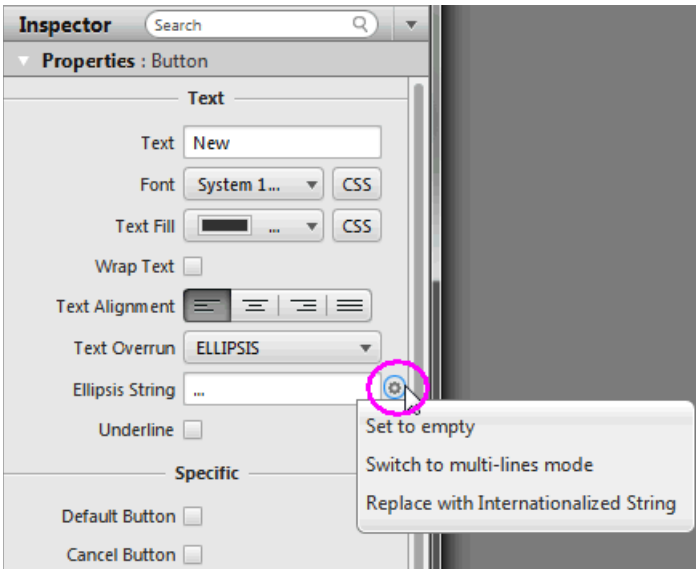
Figure 8–1 Inspector Panel with the Search Results Displayed and Layout Section Opened



When multiple elements of the same type (for example, two button elements) are selected in the Content panel or the Hierarchy panel, and if any of the properties have a different value, the minus sign (-) is displayed in the text field for that property. If a new value is entered, then the new value is set for all the selected elements.

There are some properties that have a cog menu that appears on the right side of the property field when the mouse hovers over the property value editor, as shown in Figure 8–2. The cog menu provides specific actions that can be used for that editor. For example, you can set the existing property value to null or empty. Figure 8–2 shows actions available for a text editor.

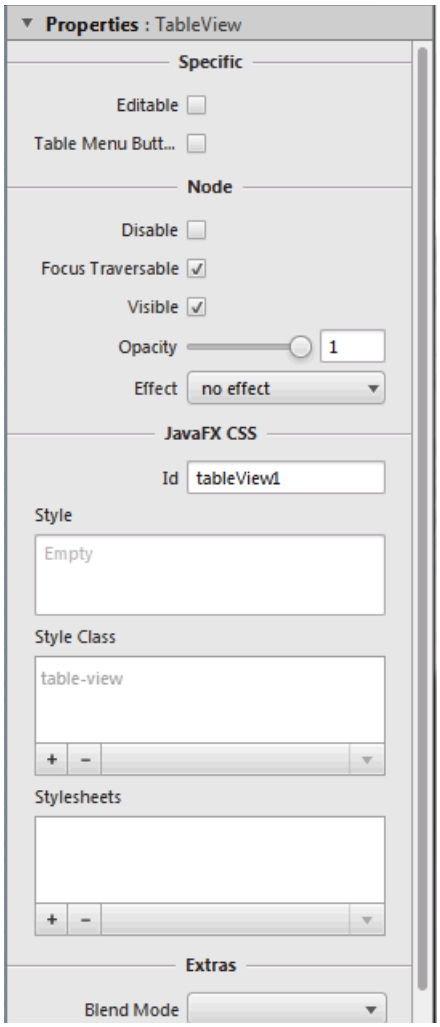
Figure 8–2 Cog Menu for Property Value Editors



Properties Section

The Properties section of the Inspector panel, shown in Figure 8–3, enables you to define the style of the selected element in the Hierarchy panel or the Content panel.

Figure 8–3 Properties Section of the Inspector Panel



When the fx:id value is assigned for the selected UI element via the Code Section, that same value is also assigned to the Id field in the CSS subsection of the Properties section only if that Id field is null.

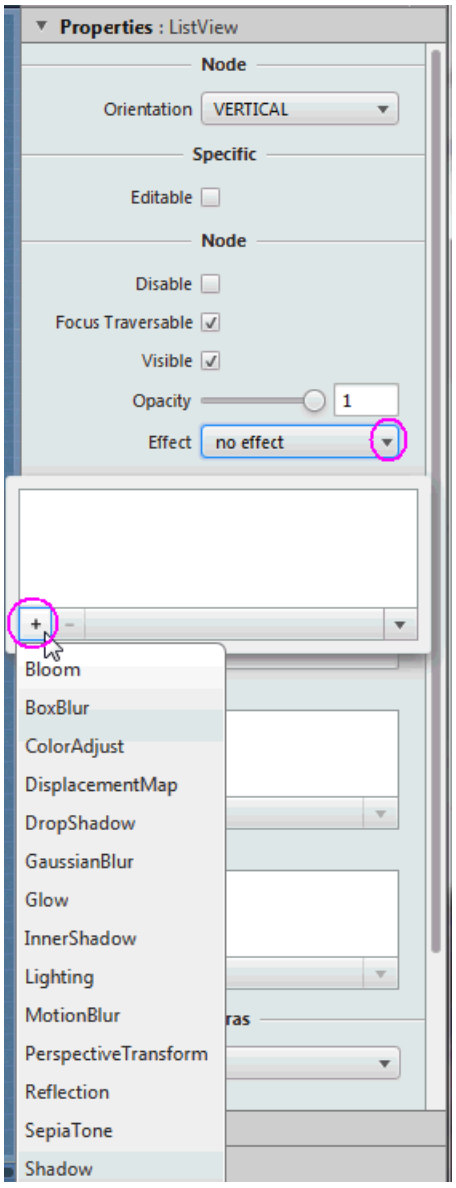
The Style, Style Class, and Stylesheets text areas in the JavaFX CSS subsection enable you to specify the style rule and style sheet to use for a particular UI component and all of its children components. You can add, delete, and put in order the style sheets that will be applied. See the Skinning with CSS and the CSS Analyzer chapter in this document for additional information.

The remaining properties displayed on the Properties section of the Inspector panel depend on the type of UI element that is selected in either the Content panel or in the Hierarchy panel. For example, if you select a radio button, toggle button, or radio menu item in the Content panel, the Properties section shows the Toggle Group property. You can use this property to assign the same `ToggleGroup id` for each toggle control that you want to add to a toggle group.

You can go from one property to another by pressing the Tab key. A new value for a property is saved when you press Enter or when you move to another property field. You can manipulate the look of the selected UI element by directly changing the value of the properties that are available and editable. For example, you can define the

effects of a selected tree view element by clicking the **down arrow** button to open the Effects list view and the **+** button to display the menu of available effects, as shown in Figure 8–4. From the list of effects available, you can select one or more effects to apply to the selected UI element. To determine the order in which the effects are applied, click the **up** or **down** buttons. The effects are applied in chronological order from the top of the list to the bottom.

Figure 8–4 Effects Menu in the Properties Section of the Inspector Panel



Layout Section

The Layout section of the Inspector panel, shown in Figure 8–5, helps you to specify the runtime behavior of the layout when the application's window is resized. It also enables you to change the size (such as, Pref Width and Pref Height) and position (such as, Layout X and Layout Y) of the selected element. Some of the information displayed in the Layout section pertains to the selected element itself (such as Min

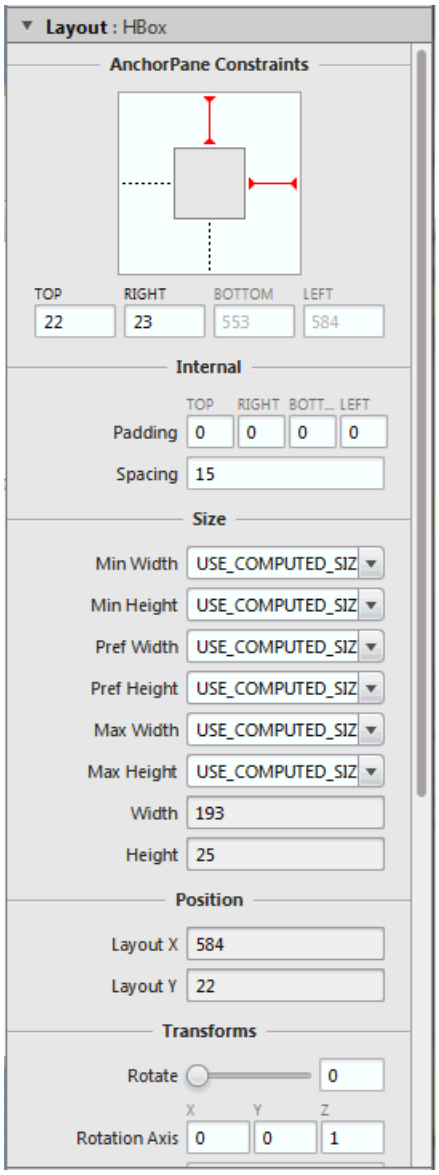
Width, Max Height, or Pref Width), and some of the information depends on the type of the container element. If the selected element is contained in an AnchorPane container, then an AnchorPane section is present in the Layout section. If the selected element is contained in a Grid Pane container, then there is a Grid Pane subsection in the Layout section of the Inspector panel. Some containers, however, do not have any associated contextual information, and so no contextual container subsection appears in the Layout section.

The AnchorPane Constraints subsection only appears when a selected element has an AnchorPane as its most immediate container. For example, in Figure 8–5, the selected HBox element's container is an AnchorPane. You can use either the diagram or the text fields in the AnchorPane Constraint subsection to manage the anchor pane's anchors. The anchors help you manage the behavior of the selected UI element when you resize the window. You can click the anchor lines to specify whether the object's size changes as the application's window is adjusted. The lines change from dashed lines to solid red lines when you click them. You can also specify the numerical values in the fields below the diagram. A grayed out numerical value means the anchor is disabled.

If the HBox element's most immediate container is a StackPane element, then the top subsection of the Layout section will be labeled as StackPane Constraints and different editable properties will be displayed.

The Transforms subsection enable you to define settings for different types of transformations, such as rotation, scaling, and translation.

Figure 8–5 Layout Section of the Inspector Panel



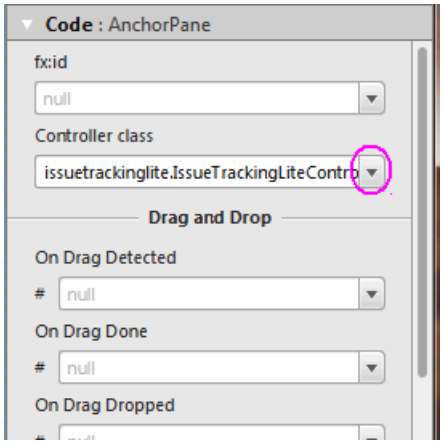
Code Section

Figure 8–6 displays the opened Code section when the topmost Anchor Pane layer is selected in the Hierarchy panel. In the fx:id field, you can assign a unique value for the selected UI element. If the controller class name is already set, you can select the fx:id value from a drop-down list for the fx:id field. In the Controller class field, you set the name of the controller class that provides the logic to handle the behavior of the objects in your FXML layout. If the controller class file already exists in the same folder as the FXML layout file you are working on, you can select the controller class from the drop-down list by clicking the down arrow that is circled in Figure 8–6. The path field contains the fully qualified Java class name of the controller class file, which must be located in the same folder, a sub-folder, or a parent folder in which the FXML file is saved. A suggested Java class name is displayed when you click the down arrow that is circled in Figure 8–6 and the controller class location has already been resolved by

Scene Builder. If you are using NetBeans IDE with JavaFX Scene Builder and your FXML file is located in a NetBeans project, then the Controller class java file can be located in any java package inside your NetBeans project's `src` folder.

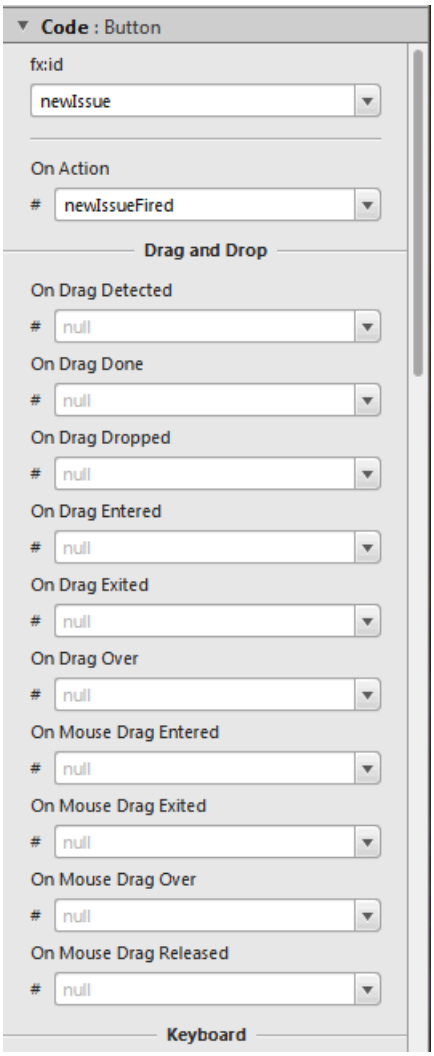
> **Tip:** The Controller Class text field is only visible when the root container is selected in the Hierarchy panel.

Figure 8–6 Code Section of the Inspector Panel When Root Anchor Pane is Selected



The Code section, shown in Figure 8–7, enables you to specify the action that should be taken when a drag-and-drop, keyboard, mouse, or any other event occurs on the selected element. From the drop-down list on the event text field, select the corresponding command that should be executed when a specific event or events occur for that UI element. The given command is a call to the controller method event handler. It is specified by a leading number sign (#), which is already added by default, followed by the name of the handler method. The method is expected to conform to the signature of a standard event handler, which means that it takes a single argument of a type that extends the `javafx.event.Event` class and should return void. For example, Figure 8–7 shows that when the OnAction event occurs on the button, the `newIssueFired()` controller method is executed. If you choose **View** from the Menu bar and then Show **Sample Controller Skeleton**, a skeleton sample source file that gives an example of what the method should look like in your controller source code.

Figure 8–7 Code Section of the Inspector Panel



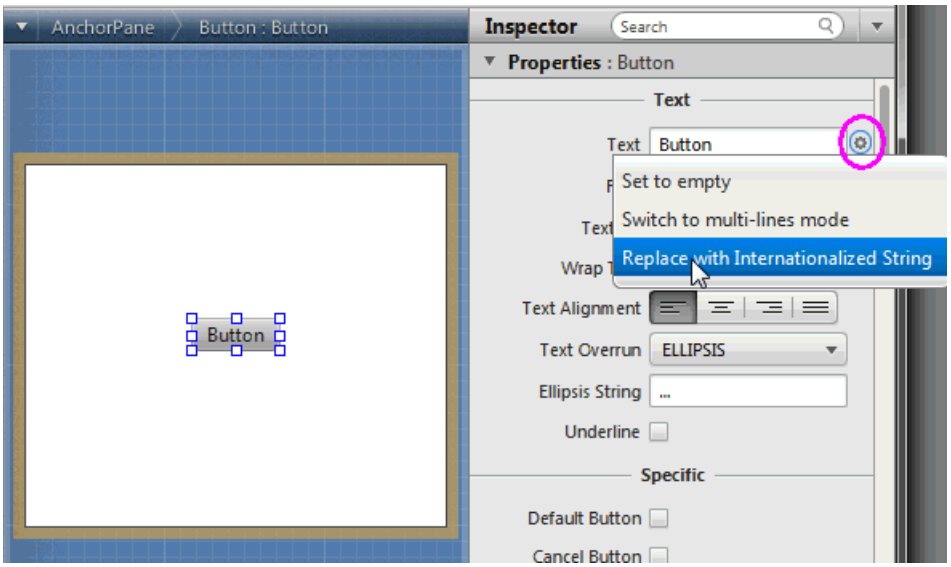
9

Internationalizing Your FXML Layout

This chapter discusses the internationalization support that is provided with JavaFX Scene Builder.

The JavaFX Scene Builder tool enables you to adapt your FXML layout to various languages and regions through the use of resource bundle files. When you add a new element to the Content panel, the element's visible property values are set to default values. For example, a button element has a default Text property value of Button. This default value can be changed with an internationalized string by hovering your mouse cursor over the right of the Text property text field, clicking the round cog icon that appears, and then selecting **Replace with Internationalized String**, as shown in Figure 9–1. Notice that the default value is then replaced by %key.unspecified, which you need to replace with the appropriate resource key name that is defined in your resource file. The percent sign (%) prefix is a visible indicator that the Text property value following it is a resource key name. You can still use the percent sign as a first character for the Text property value. If you have not selected the Replace with Internationalized String setting, then in the FXML file, the percent sign will appear as \%. This indicates that the rest of the property value has no link with internationalization and that it is not a key name.

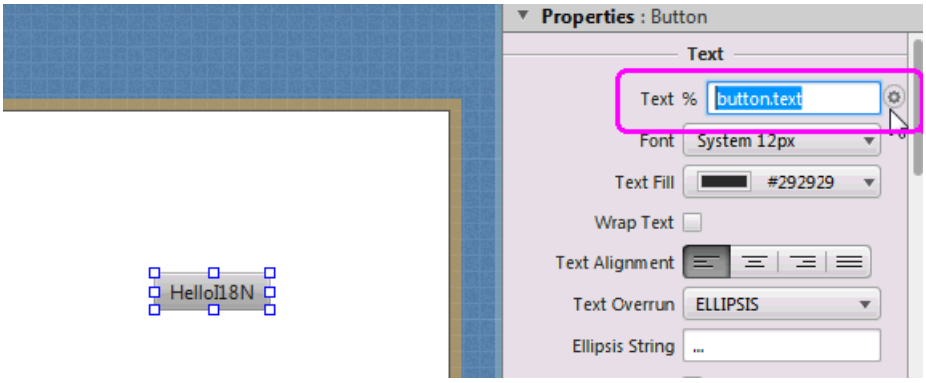
Figure 9–1 Internationalizing a Button Element



The HelloI18N sample can help show the internationalization support. The sample is included with the JavaFX Scene Builder samples bundle. Its `Main` class shows how to load the resource bundle. The sample includes the Bundle.properties and Bundle_fr_FR.properties files. When you open the *<javafx-scenebuilder-samples-1_0-beta-install-dir>*/HelloI18N/src/helloi18n/HelloI18N.fxml file, the FXML file is opened in its own instance of the JavaFX Scene Builder window and you see a button that is labeled HelloI18N. Figure 9–2 shows that the button element's Text property value was set to use the button.text resource key, which is defined in the Bundle.properties file as having the value of HelloI18N. The percent sign in front of the Text property's text field is a visible indicator that the value in the text field is a resource name. If you replace the value assigned to button.text in the Bundles.properties file and save the change, the new value is immediately reflected in the button displayed on the Content panel. You can define additional resource keys in your resource bundle files and use them in the FXML layout you are building.

Select the **Preview** from the Main menu, then **Internationalization** command, and then **Set Resource**. In the Choose Internationalization Resource File dialog box, choose the Bundle_fr_FR.properties file to see how the HelloI18N sample application appears when using the French resource file. Notice that the text displayed on the button is now Bonjour. Changes you make and save to the resource file are reflected in the FXML layout you are working on in the JavaFX Scene Builder tool.

Figure 9–2 Using Resource Key Value to Internationalize the Text Property Value



10

Skinning with CSS and the CSS Analyzer

This chapter describes the Cascading Style Sheet (CSS) support and describes the CSS Analyzer feature that JavaFX Scene Builder provides.

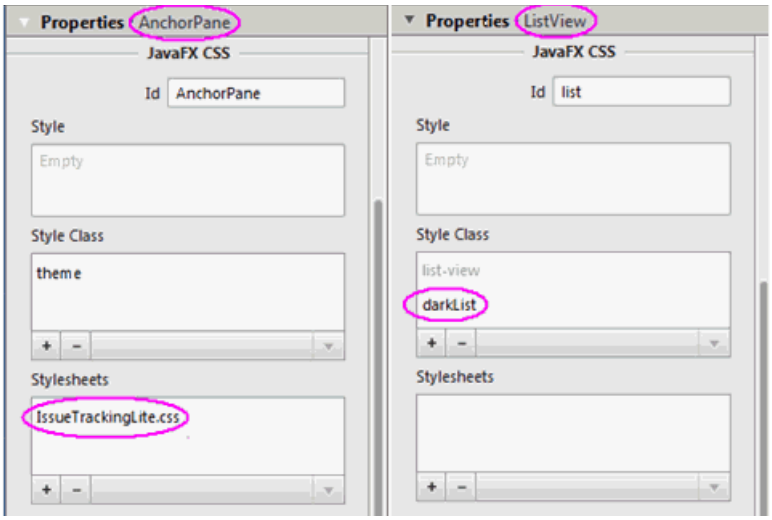
The JavaFX UI controls used by Scene Builder are pre-styled with a default JavaFX look and feel. Scene Builder immediately renders this predefined JavaFX style when you drag the UI control from the Library panel to the Content or Hierarchy panel. You can customize the style used in your application by changing the component's properties via the Properties section of the Inspector panel or by defining your own styling rules in a CSS file.

Scene Builder currently does not generate CSS files, but enables you to use your local CSS editor to create and modify your CSS file. The changes you save in the CSS file that is used by the current FXML layout displayed in the Content panel is immediately rendered by Scene Builder.

You can add the CSS rules at the Scene level, within a given container, or by an inline styling at the Node level. By using the Stylesheets list view in the JavaFX CSS subsection of the Properties section of the Inspector panel, you can assign a CSS file to use on a selected UI element in your FXML layout if that UI element is either a container or a UI control. You can attach a style sheet to any part of your FXML layout, from the topmost parent UI element to the lowest.

The left side of Figure 10–1 shows the Properties section of the Inspector panel with the `IssueTracking.css` file assigned to the topmost Anchor Pane used in the layout. The right side of the figure shows the darkList style class assigned to the ListView element. The darkList style class is defined in the `IssueTracking.css` style sheet and inherited by the ListView element from its container parent, the topmost AnchorPane container.

Figure 10–1 Properties Section with Style Class and Style Sheet List Views Displayed



CSS rules that are defined on a parent element can be used to style the parent UI element itself and all of its children elements. You can define the specific style class to use with a UI element by adding it to the Style Class list view in the Properties section of the Inspector panel. The CSS files referenced from the Stylesheets properties are referenced from within the FXML file and so they are deployed with the FXML file.

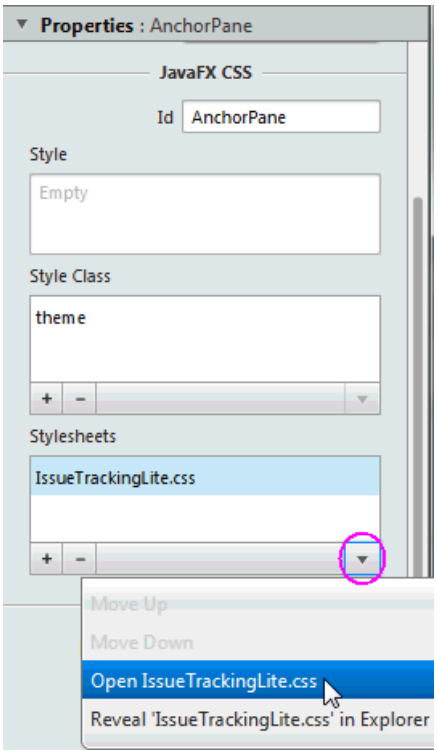
In Scene Builder, you can simulate the attachment of a style sheet to an application Scene by selecting **Preview**, then **Scene Style Sheets**, and finally choosing **Add a Style Sheet** or **Open a Style Sheet** option. This Preview command is useful when the "root" style class is defined in the style sheet. In this case, once the style sheet is attached to the Scene, the styles defined in the "root" class are applied to the layout in the Content panel.

You can edit an existing CSS file with your system editor by using the following steps:

1. In the Stylesheets list view of the Properties section of the Inspector panel, click on the CSS file you want to edit.
2. Click the drop-down arrow on the bottom right of the list view, as shown in Figure 10–2.
3. Select the **Open** command for the CSS file you want to edit.

You can also reveal the CSS file's location in your system using the Reveal command that is also available in the drop-down menu. You can also navigate to the CSS file via the CSS Analyzer panel, as explained later in Using the CSS Analyzer Panel.

Figure 10–2 Open CSS File From the Properties Section of the Inspector Panel

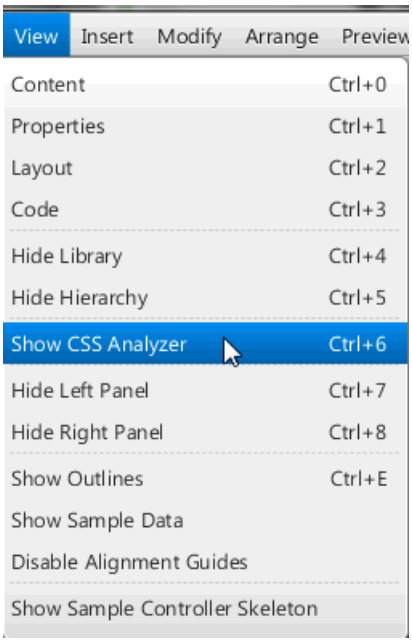


Using the CSS Analyzer Panel

The CSS Analyzer Panel enables you to understand how various possible CSS rules can affect aspects of a currently selected UI element. It displays a synoptic view of all the possible sources for the property values. Each CSS property value assigned to a certain aspect of the selected UI element may originate from either the API or predefined CSS rules. The sources are listed in prioritized order, which enables you to understand why a given source takes precedence over another. The panel also provides a means for you to navigate to the source of CSS property value to help you develop and troubleshoot CSS stylesheets.

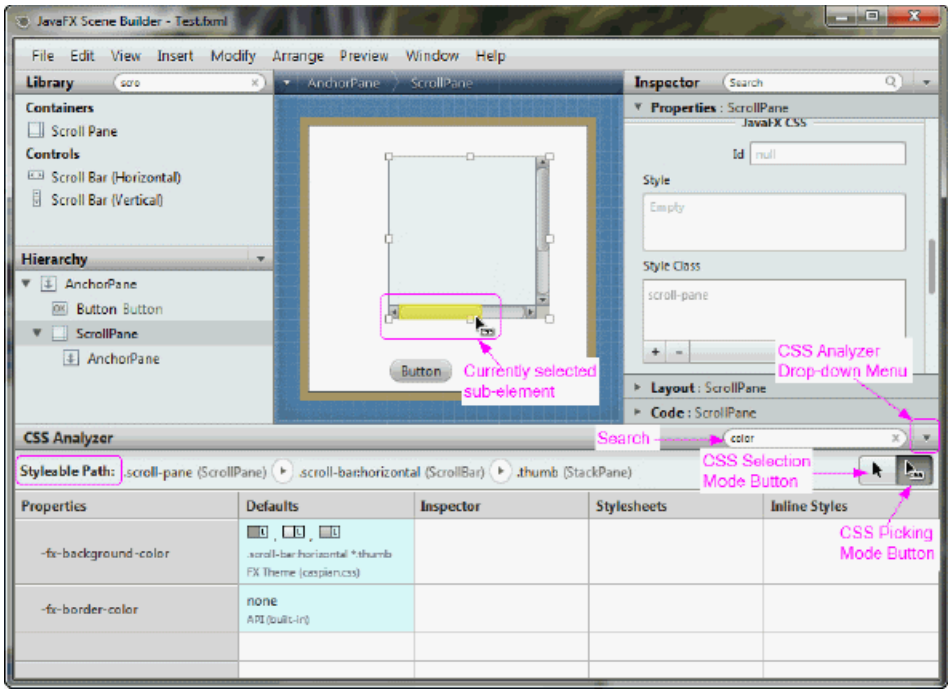
The CSS Analyzer panel is not displayed by default. To display the panel, select **View** from the Main menu and then **Show CSS Analyzer**, as shown in Figure 10–3.

Figure 10–3 Show CSS Analyzer



The CSS Analyzer panel appears at the bottom of the Main window, similar to what is shown in Figure 10–4.

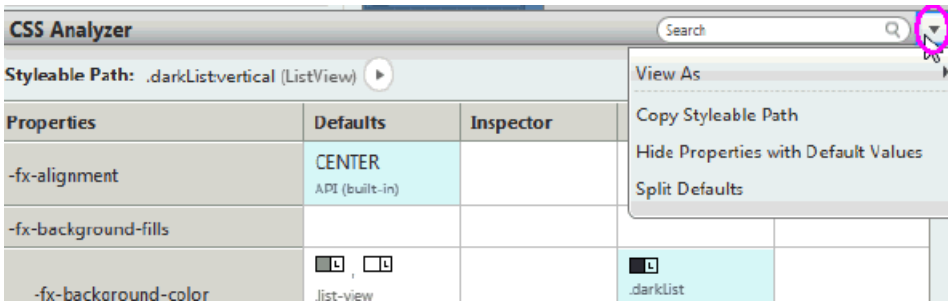
Figure 10–4 CSS Analyzer Panel Displayed (Click image to enlarge.)



The panel includes the following sections, most of which are highlighted in Figure 10–4.

- **Search text field:** Located at the top right corner of the panel. It enables you to isolate the specific properties that you want to view.
- **CSS Analyzer Menu:** Located to the right of the Search text field. Click on the drop-down arrow to see the menu of available commands, as shown in Figure 10–5.
  - **View As:** Enables you to choose the display format for the style properties. The default is Table view. The Rules view mode displays the properties in CSS rules formatting. The Text view mode displays the properties in text styling.
  - **Copy Styleable Path:** Enables you to copy the current value in the Styleable Path text field and you can paste it into your CSS file to modify the style of the currently selected component or its sub-element.
  - **Hide Properties with Default Values:** Removes from view all the properties that only have default values assigned to them. Properties that have non-default values, such as Stylesheets and Inline Styles, continue to be displayed. The **Show Properties with Default Values** shows all the properties for the component.
  - **Split Defaults:** Refreshes the view of the properties to include two columns for the default values of the style properties: API Defaults and the FX Theme Defaults. The Join Defaults command displays the combined default values into a single column, which is the default.

Figure 10–5 CSS Analyzer Menu



- **Styleable Path text field:** Located on the top left corner of the panel. It enables you to discover the sub-elements when you click on the arrows in the path. You can copy that path using the CSS Analyzer menu and paste the path in your CSS file to assign a new style value.
- **CSS Picking Mode button:** Located below the Search text field. This is the default mode when CSS Analyzer is opened. It allows you to select a sub-element of the currently selected component. In Figure 10–4, the CSS Picking Mode button is selected. This allowed you to select the bottom horizontal scrollbar of the ScrollPane component in the Content panel. The scrollbar is highlighted in yellow to indicate that it is the currently selected sub-element of the ScrollPane. Correspondingly, the Styleable Path shows the complete path to the currently selected element. This feature shows how skins can be styled.
- **CSS Selection Mode button:** Located next to the CSS Picking Mode button. It is the standard selection mode and allows you to select a component.
- **Properties column:** First column in the table. Displays all of the available style properties that is available for the currently selected element.

- **Default column:** Displays the default values, both from the API and the JavaFX theme, that are delivered for the style property.
- **Inspector column:** Displays the property value that has been set using the Inspector panel. Some properties appear in both the Properties section of the Inspector panel and the CSS Analyzer panel. To edit these properties, hover the mouse cursor in the corresponding cell of the CSS Analyzer panel and click the cog icon in the top right corner to select **Reveal in Inspector**. The corresponding property is highlighted with a blue ring in the Properties section of the Inspector panel.
- **Stylesheets column:** Displays the value that is defined for the property via the CSS file set in the Stylesheets text view in the Properties section of the Inspector panel. The name of the CSS file the value is derived from is included in the column. You can also click on the cog icon on the upper right corner of the cell and select the **Open** command to open your CSS file in your default CSS editor.
- **Inline Styles column:** Displays the inline style value that is defined for the property via the Style text view in the JavaFX CSS subsection of the Properties section of the Inspector panel. You can select the **Reveal in Inspector** command using the cell's cog icon on the upper right corner of the table cell to display the Style text view in the Inspector panel.

You can modify the properties values from your CSS file. Alternatively, you can use the JavaFX CSS subsection of the Properties section of the Inspector panel to either edit a CSS property or use the Style property to override any CSS property.

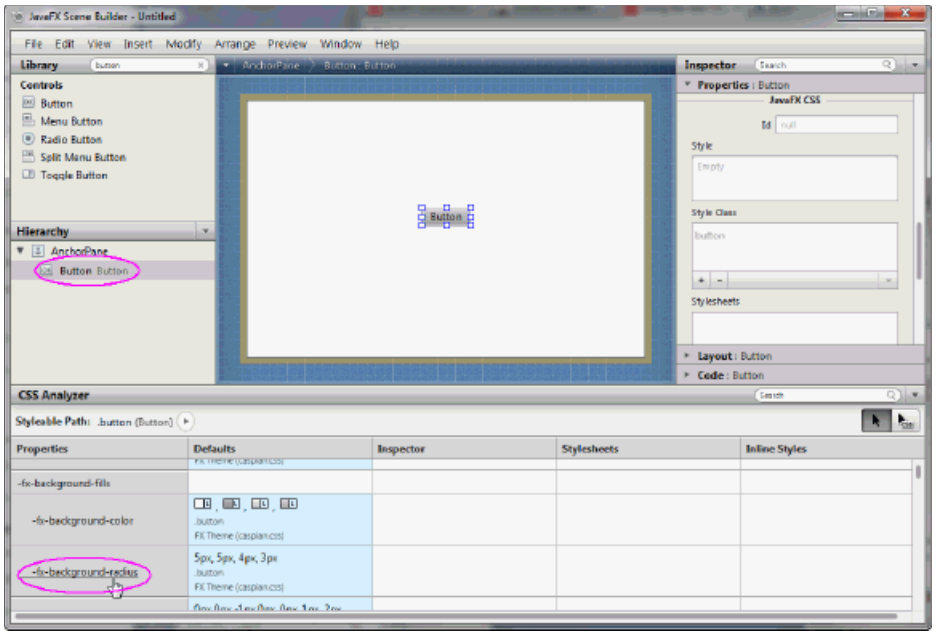
When a CSS property value is assigned to a currently selected UI component, the new style is immediately rendered in the Content panel.

To learn more about the CSS Analyzer's functionalities, use the following steps to customize the Button component to have rounded edges and use that style for all buttons that are subsequently added to your FXML layout.

1. Drag a Button from the Library to the Content panel.

   You will see the CSS Analyzer displays the value of `-fx -background-radius`, shown in Figure 10–6, as one of the CSS properties for the button.

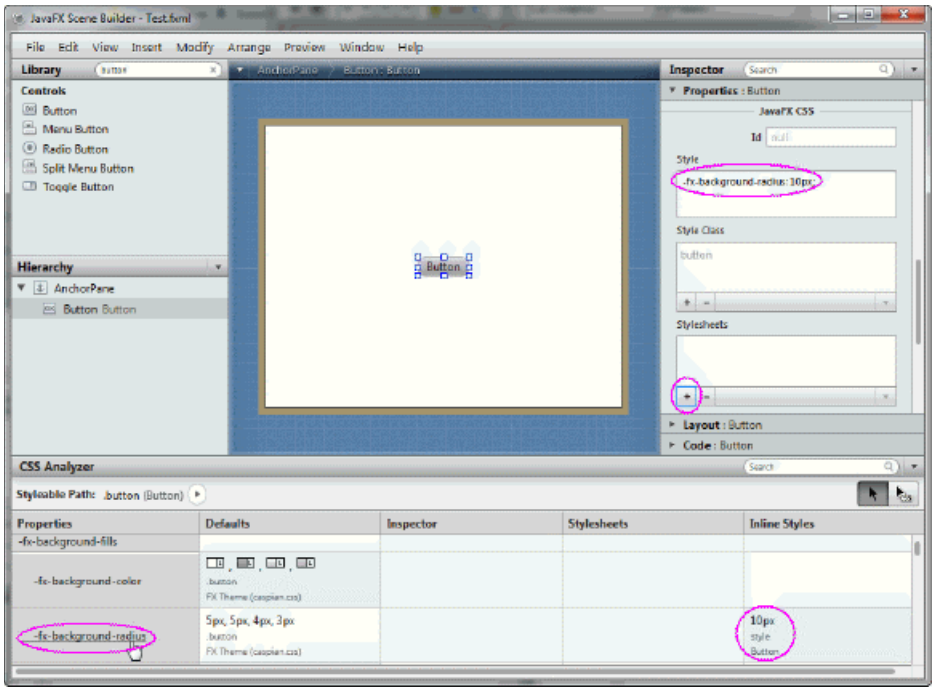
Figure 10–6 A Button's Properties Displayed in CSS Analyzer (Click image to enlarge.)



2. Click the **-fx -background-radius** property name in the Properties column and you are taken to the corresponding online API documentation in the `JavaFX CSS Reference Guide`, which provides the syntax you can use with each property's value.

3. In the Scene Builder window, click on the Properties section of the Inspector panel. Enter `-fx-background-radius: 10px` in the Style text field of the JavaFX CSS subsection, as shown in Figure 10–7.
   Notice that the Button component in the Content panel is now rounded around the corners. Also, in the CSS Analyzer panel, you'll see that the row for the `-fx-background-radius` property now has two values: the default and the inline style value of `10px` that you just entered in the Style property, as shown in Figure 10–7. That cell's background color also changed to blue, which indicates that the value is now the current value being rendered on the currently selected component.

Figure 10–7 Making a Button Rounded (Click image to enlarge.)



4. Create a CSS rule so that you can apply the new button style to all of the buttons you add to your current FXML layout.

   a. Using your system's command, create am empty CSS file, e.g. test.css, in the same directory that contains your current FXML layout.

   b. In the Properties section of the Inspector panel, click the **+** button in the Stylesheets list view, as indicated in Figure 10–7.

   c. In the Add Style Sheet dialog, navigate to the `test.css` file location, select it, and click Open to add it to the Stylesheets property.
   You'll see the `test.css` file added in the Stylesheets list view.

   d. Edit the `test.css` file by clicking on the pull-down arrow in Stylesheets property and select **Open test.css**.

   e. Add the following CSS rule to `test.css` and save the file.

Example 10–1 Add CSS Rule for a Rounded Button

```
.button {
-fx-background-radius: 10px;
}
```

   f. Select the button again in the Content panel. Notice that in the CSS Analyzer panel, the button is now inheriting the CSS rule from the test.css file because the button is a child of the AnchorPane component. Since it still has the inline style value assigned to it, that value has precedence and is the style rendered in the Content panel. If you do not have a need to change the style for this particular button, you can remove that style value from the Properties section of the Inspector panel.

11

Next Step

Now that you are familiar with the JavaFX Scene Builder user interface, use Getting Started with JavaFX Scene Builder to create a simple issue-tracking application.