

Styling UI Controls

There are two methods for customizing the look of a Control. The most difficult and yet most flexible approach is to write a new Skin for the Control which precisely implements the visuals which you desire for the Control. Consult the Skin documentation for more details.

The easiest and yet very powerful method for styling the built in Controls is by using CSS. Please note that in this release the following CSS description applies only to the default Skins provided for the built in Controls. Subsequent releases will make this generally available for any custom third party Controls which desire to take advantage of these CSS capabilities.

Each of the default Skins for the built in Controls is comprised of multiple individually styleable areas or regions. This is much like an HTML page which is made up of <div>'s and then styled from CSS. Each individual region may be drawn with backgrounds, borders, images, padding, margins, and so on. The JavaFX CSS support includes the ability to have multiple backgrounds and borders, and to derive colors. These capabilities make it extremely easy to alter the look of Controls in JavaFX from CSS.

The colors used for drawing the default Skins of the built in Controls are all derived from a base color, an accent color and a background color. Simply by modifying the base color for a Control you can alter the derived gradients and create Buttons or other Controls which visually fit in with the default Skins but visually stand out.

As with all other Nodes in the scenegraph, Controls can be styled by using an external stylesheet, or by specifying the style directly on the Control. Although for examples it is easier to express and understand by specifying the style directly on the Node, it is recommended to use an external stylesheet and use either the styleClass or id of the Control, just as you would use the "class" or id of an HTML element with HTML CSS.

Each UI Control specifies a styleClass which may be used to style controls from an external stylesheet. For example, the Button control is given the "button" CSS style class. The CSS style class names are hyphen-separated lower case as opposed to camel case, otherwise, they are exactly the same. For example, Button is "button", RadioButton is "radio-button", Tooltip is "tooltip" and so on.

The class documentation for each Control defines the default Skin regions which can be styled. For further information regarding the CSS capabilities provided with JavaFX, see the CSS Reference Guide.