### Component Specific Properties

The look, and to some degree the feel of Synth can be customized by way of component specific properties. These properties are accessed from [SynthStyle#get](http://docs.google.com/SynthStyle.html). Refer to [synth file format](http://docs.google.com/synthFileFormat.html#e.property) for examples of how to specify these properties in a synth configuration file.

This file specifies the expected class type each of the values are to take. The behavior of supplying the wrong type is unspecified: typically a ClassCastException is thrown, but it is implementation specific.

#### ArrowButton

ArrowButton is a special type of JButton that renders an arrow. ArrowButton is typically not created directly, rather some of the Components will create it to render a button with an arrow. The components that make use of ArrowButton are: JComboBox, JScrollBar and JSplitPane (for the buttons on the divider). In addition to the [Button properties](#_3j2qqm3), ArrowButton supports the following propeties:

##### ArrowButton Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| ArrowButton.size | Integer | 16 | Preferred size of the arrow button. |

#### JButton

JButton paints text using the TEXT\_FOREGROUND ColorType. In addition to the [Button properties](#_3j2qqm3), JButton supports the following property:

##### JButton Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| Button.defaultButtonFollowsFocus | Boolean | true | Whether or not the default JButton should change to the button that is receiving focus. |

#### JCheckBox

JCheckBox paints text using the TEXT\_FOREGROUND ColorType. In addition to the [Button properties](#_3j2qqm3), JCheckBox supports the following property:

##### JCheckBox Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| CheckBox.icon | Icon | null | Icon used to render the check. |

#### JComboBox

JComboBox is a composite component that consists of the following child Components:

| Name | Type | Description |
| --- | --- | --- |
| ComboBox.arrowButton | SynthArrowButton | Renders the drop down indicator. |
| ComboBox.list | JList | JList used to render the values. |
| ComboBox.listRenderer | Component | The renderer used for the JComboBox's JList. |
| ComboBox.popup | JPopupMenu | JPopupMenu used to show the ComboBox.list of the JComboBox. |
| ComboBox.renderer | Component | The renderer used for the JComboBox. This is ONLY set if the renderer is a UIResource. |
| ComboBox.scrollPane | JScrollPane | JScrollPane containing the JList. |
| ComboBox.textField | JTextField | The editor. |

##### JComboBox Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| ComboBox.showPopupOnNavigation | Boolean | false | Whether or not the JPopupMenu should appear when navigating with the keyboard. |

#### JFileChooser

##### JFileChooser Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| FileChooser.cancelIcon | Icon | null | Icon displayed on cancel button of the file chooser. |
| FileChooser.okIcon | Icon | null | Icon displayed on the ok button of the file chooser. |
| FileView.directoryIcon | Icon | null | Icon used for directories. |
| FileView.fileIcon | Icon | null | Icon used for files. |
| FileView.computerIcon | Icon | null | Icon used for directories that represent the computer. Not all platforms will make use of this icon. |
| FileView.hardDriveIcon | Icon | null | Icon used to represent the root of a hard drive. For example, on Windows this would be used when viewing the C drive. |
| FileView.floppyDriveIcon | Icon | null | Icon used to represent a floppy disk. |
| FileChooser.newFolderIcon | Icon | null | Icon used by the button that creates a new folder. |
| FileChooser.upFolderIcon | Icon | null | Icon used by the button that navigates to the parent folder. |
| FileChooser.homeFolderIcon | Icon | null | Icon used by the button that navigates to the current user's home directory. |
| FileChooser.detailsViewIcon | Icon | null | Icon used by the button that toggles the detailed files list view. |
| FileChooser.listViewIcon | Icon | null | Icon used by the button that toggles the regular files list view, showing only an icon and the name of each file and directory. |

#### JInternalFrame

##### JInternalFrame Specific Properties

| Property | Expected Type | DefaultValue | Description |
| --- | --- | --- | --- |
| InternalFrame.icon | Icon | null | Icon drawn representing the system  icon of the internal frame.  If pressed  the system menu will be shown. |

#### JInternalFrameTitlePane

JInternalFrameTitlePane is the control bar located at the top of the internal frame similar to that found in a frame.

##### JInternalFrameTitlePane Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| InternalFrameTitlePane.maximizeIcon | Icon | null | Icon drawn to indicate the ability to maximize the internal frame. |
| InternalFrameTitlePane.minimizeIcon | Icon | null | Icon drawn to indicate the ability to restore the internal frame back to its previous state. |
| InternalFrameTitlePane.iconifyIcon | Icon | null | Icon drawn to indicate the ability to minimize the internal frame. |
| InternalFrameTitlePane.closeIcon | Icon | null | Icon drawn to indicate the abililty to close the internal frame. |
| InternalFrameTitlePane.titleSpacing | Integer | 2 | Space between the buttons on the title pane and the title. |
| InternalFrameTitlePane.buttonSpacing | Integer | 2 | Space between the buttons on the title pane. |
| InternalFrameTitlePane.maxFrameIconSize | Dimension | 16x16 | Maximum size of the frame icon that will be rendered on the title pane. If the icon is bigger than this size, it will be scaled down. |
| InternalFrameTitlePane.titleAlignment | leading|trailing|center | leading | Alignment for the title. With a left to right component orientation leading is left and trailing right. With a right to left component orientation leading is right and trailing left. |

#### JList

JList's sets the name of the renderer to List.renderer. JList supports the following properties:

##### JList Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| List.rendererUseListColors | Boolean | true | If true the renderers state is not updated, and the text colors come from JList's getSelectionBackground and getSelectionForeground methods. If false, the renderer's state is updated and the colors will instead come from the Style. |
| List.rendererUseUIBorder | Boolean | true | If true setBorder on the renderer will succeed regardless of the border passed in, otherwise setBorder on the renderer will only succeed if it is a Synth Border. |
| List.cellHeight | Integer | -1 | Cell height of the JList. Is this is -1, setFixedCellHeight is not invoked. |

#### Menu Properties

The Menu classes (JCheckBoxMenuItem, JMenu, JMenuItem, and JRadioButtonMenuItem) all support the same set of properties and behave similarly. Each component consists of two Regions: the region specific to the component and Region.MENU\_ITEM\_ACCELERATOR. MENU\_ITEM\_ACCELERATOR is used for painting the accelerator. Both Regions paint text using the TEXT\_FOREGROUND ColorType. The following set of properties are supported:

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| prefix.acceleratorDelimiter | String | + | String separator between the description of the modifiers and the key, for example Ctrl <acceleratorDelimiter> X. |
| prefix.arrowIcon | Icon | null | Icon drawn to the right of the text (or left when the ComponentOrientation is right to left) of the text. This is typically only defined for JMenu. |
| prefix.checkIcon | Icon | null | Icon drawn to the left of the text (or right when the ComponentOrientation is right to left) of the text. This is typically only defined for JCheckBoxMenuItem and JRadioButtonMenuItem to provide the check or radio button. |
| prefix.margin | Insets | Empty Insets (0, 0, 0, 0) | Margin for the JMenuItem. Refer to the javadoc of javax.swing.AbstractButton#setMargin(java.awt.Insets) for details of how the margin is used. |
| prefix.textIconGap | Integer | 4 | Padding between the icon and text. Refer to the javadoc for javax.swing.AbstractButton#setIconTextGap(int) for details of how this is used. |

Prefix is one of: CheckBoxMenuItem, Menu, MenuItem, or RadioButtonMenuItem.

JMenu also supports the following properties:

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| Menu.delay | Integer | 200 | Sets the delay before the menu appears. |
| Menu.shortcutKeys | int[] | KeyEvent.ALT\_MASK | Array of the KeyEvent modifiers to use for shortcut keys. |

#### JOptionPane

JOptionPane is a composite component and may consist of numerous child components, they are: OptionPane.button, OptionPane.label, OptionPane.comboBox, OptionPane.scrollPane, OptionPane.list, OptionPane.textField, OptionPane.iconLabel.

##### JOptionPane Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| OptionPane.buttonAreaBorder | Border | null | Border around the button area. |
| OptionPane.buttonClickThreshhold | Integer | 0 | Passed to the created buttons setMultiClickThreshhold. |
| OptionPane.buttonOrientation | Integer | SwingConstants.CENTER | How the buttons should be layed out, one of SwingConstants.LEFT, SwingConstants.RIGHT or SwingConstants.CENTER (this will be flipped when in a right to left locale). |
| OptionPane.buttonPadding | Integer | 6 | Amount of space between buttons. |
| OptionPane.cancelIcon | Icon | null | Icon for the cancel button. |
| OptionPane.errorIcon | Icon | null | Icon used for ERROR\_MESSAGE. |
| OptionPane.informationIcon | Icon | null | Icon used for INFORMATION\_MESSAGE. |
| OptionPane.isYesLast | Boolean | false | If true the 'yes' button leads other buttons, otherwise it follows the other buttons. |
| OptionPane.questionIcon | Icon | null | Icon used for error QUESTION\_MESSAGE. |
| OptionPane.minimumSize | Dimension | 262x90 | Minimum size for the JOptionPane. |
| OptionPane.noIcon | Icon | null | Icon for the no button. |
| OptionPane.okIcon | Icon | null | Icon for the ok button. |
| OptionPane.sameSizeButtons | Boolean | true | Whether or not all the buttons should be the same size. |
| OptionPane.separatorPadding | Integer | 6 | Amount of padding between the message area and separator. |
| OptionPane.yesIcon | Icon | null | Icon for the yes button. |
| OptionPane.warningIcon | Icon | null | Icon used for error WARNING\_MESSAGE |

#### JProgressBar

##### JProgressBar Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| ProgressBar.repaintInterval | Integer | 50 | Number of milliseconds between repaints for indeterminate progress bars. |
| ProgressBar.cycleTime | Integer | 3000 | Number of milliseconds used to determine how far to move  the bouncing box per frame when the progress bar is indeterminate. |

#### JRadioButton

JRadioButton paints text using the TEXT\_FOREGROUND ColorType. In addition to the [Button properties](#_3j2qqm3), JRadioButton supports the following property:

##### JRadioButton Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| RadioButton.icon | Icon | null | Icon used to render the radio indicator. |

#### JScrollBar

JScrollBar is a composite component that consists of the following child Components:

| Name | Type | Description |
| --- | --- | --- |
| ScrollBar.button | SynthArrowButton | Identifies the increment/decrement buttons. |

##### JScrollBar Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| ScrollBar.allowsAbsolutePositioning | Boolean | false | If true, middle mouse click in the track will set the position of the track to where the mouse is. |
| ScrollBar.maximumThumbSize | Dimension | 4096x4096 | Maximum thumb size. |
| ScrollBar.minimumThumbSize | Dimension | Derived from ScrollBar.thumbHeight | The minimum thumb size. If this is not specified the minimum thumb size is ScrollBar.thumbHeight plus the insets of the scrollbar along the opposite axis of the scrollbar and 7 along the other axis. For example, if you have not specified Insets for a scrollbar, and the orientation of the scrollbar is vertical, than the minimum thumb size width is ScrollBar.thumbHeight with a height of 7. |
| ScrollBar.squareButtons | Boolean | false | If true, the width and height of the scrollbar buttons will be made equal. |
| ScrollBar.thumbHeight | Integer | 14 | The preferred height, when the orientation of the scrollbar is horizontal, or preferred width when the orientation of the scrollbar is vertical. |

#### Separators

All of the separator classes, JSeparator, JPopupMenu.Separator and JToolBar.Separator use the same property:

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| Separator.thickness | Integer | 2 | Preferred width, for vertically aligned separators, or preferred height for horizontally aligned separators. The resulting preferred size will include the Insets. |

JToolBar.Separator also supports the following property:

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| ToolBar.separatorSize | Dimension | null | The value of this is passed to the javax.swing.JToolBar$Separator#setSeparatorSize(java.awt.Dimension) method. If unspecified setSeparatorSize is not invoked. |

#### JScrollPane

JScrollPane is unique in that it provides a method for setting the Border around the JViewport with JViewport throwing an IllegalArgumentException from setBorder. To accomodate this a special border is installed on the JScrollPane that uses the insets from the key ScrollPane.viewportBorderInsets. The SynthPainter method paintViewportBorder is called to paint the Viewports border.

##### JScrollPane Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| ScrollPane.viewportBorderInsets | Insets | null | Insets for the viewport Border. |

#### JSplitPane

JSplitPane is a composite component that will contain a divider and potentially two buttons, if setOneTouchExpandable(true) has been invoked. The two buttons will be named: SplitPaneDivider.leftOneTouchButton and SplitPaneDivider.rightOneTouchButton.

##### JSplitPane Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| SplitPane.centerOneTouchButtons | Boolean | true | If true, the one touch buttons are centered on the divider. |
| SplitPane.oneTouchButtonOffset | Integer | 2 | Distance between the one touch buttons. |
| SplitPane.oneTouchButtonSize | Integer | 6 | Size to make the one touch buttons. |
| SplitPane.oneTouchExpandable | Boolean | null | If non null, the value of this is passed to setOneTouchExpandable. |
| SplitPane.size | Integer | 6 | Size of the divider |

#### JSlider

##### JSlider Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| Slider.thumbWidth | Integer | 30 | Width of the slider thumb |
| Slider.thumbHeight | Integer | 14 | Height of the slider thumb |
| Slider.trackBorder | Integer | 1 | Width of the track border |
| Slider.paintValue | Boolean | true | Whether or not to paint the current value  of the slider. |

#### JTabbedPane

##### JTabbedPane Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| TabbedPane.tabRunOverlay | Integer | 0 | Number of pixels to overlap tabs when there is more than one row. |
| TabbedPane.textIconGap | Integer | 0 | Padding added between the icon and text on a tab. If there is no text or icon this value is not used. |
| TabbedPane.selectedTabPadInsets | Insets | Empty Insets (0, 0, 0, 0) | Extra insets added to the selected tab. |
| TabbedPane.selectionFollowsFocus | Boolean | true | If true the selection of the tabbed pane changes as the user navigates the tabs with a mouse. |

#### JTable

JTable sets the name of the renderer to Table.cellRenderer. JTable supports the following properties:

##### JTable Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| Table.gridColor | Color | null | Color used for the grid. If null the foreground from the style is used. |
| Table.rendererUseTableColors | Boolean | true | If true the renderers state is not updated, and the text colors come from JTable's getSelectionBackground and getSelectionForeground methods. If false, the renderer's state is updated and the colors will instead come from the Style. |
| Table.rendererUseUIBorder | Boolean | true | If true setBorder on the renderer will succeed regardless of the border passed in, otherwise setBorder on the renderer will only succeed if it is a Synth Border. |
| Table.rowHeight | Integer | -1 | If not -1 setRowHeight is invoked on the JTable. |

#### JTree

JTree sets the name of the renderer to Tree.renderer, the name of the editor is Tree.cellEditor.

##### JTree Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| Tree.collapsedIcon | Icon | null | Icon to use when a node is collapsed. |
| Tree.drawHorizontalLines | Boolean | true | If true nodes have a horizontal connecting them to the leading edge of their parent. |
| Tree.drawVerticalLines | Boolean | true | If true a vertical line is drawn down from expanded nodes. |
| Tree.expandedIcon | Icon | null | Icon to use when a node has been expanded. |
| Tree.leftChildIndent | Integer | 0 | This plus Tree.rightChildIndent account for the total space, along the y axis, to offset nodes from their parent. |
| Tree.rightChildIndent | Integer | 0 | This plus Tree.leftChildIndent account for the total space, along the y axis, to offset nodes from their parent. |
| Tree.rowHeight | Integer | -1 | Row height for the Tree. |
| Tree.scrollsHorizontallyAndVertically | Boolean | false | If false and scrolling needs to happen to accomodate cells it will only happen along the vertical axis, if true, scrolling may happen along both the horizontal and vertical axis. |
| Tree.scrollsOnExpand | Boolean | true | Whether or not the JTree should scroll when a node is expanded. |
| Tree.linesStyle | String |  | The styles of the lines in the tree. Synth supports solid lines (empty value) and dashed lines ("dashed" value). It is possible to add new styles by creating a new [SynthGraphicsUtils](http://docs.google.com/SynthGraphicsUtils.html) and [binding it](http://docs.google.com/synthFileFormat.html#e.graphicsUtils) to the tree. |

#### JToggleButton

JToggleButton paints text using the TEXT\_FOREGROUND ColorType. In addition to the [Button properties](#_3j2qqm3), JToggleButton supports the following property:

##### JToggleButton Specific Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| ToggleButton.icon | Icon | null | Icon used to render the radio indicator. |

#### Button Properties

Each of the Button classes (JButton, JCheckBox, JRadioButton, JToggleButton and SynthArrowButton) support a similar set of properties. These properties are:

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| prefix.contentAreaFilled | Integer | true | Refer to the javadoc of javax.swing.AbstractButton#contentAreaFilled(boolean) for details of how this is used. It is up to the Painter to properly honor this property. |
| prefix.iconTextGap | Integer | If unspecified, JButton.setIconTextGap is not invoked. | Padding between the icon and text. Refer to the javadoc of javax.swing.AbstractButton#setIconTextGap(int) for details of how this is used. |
| prefix.margin | Insets | Empty Insets (0, 0, 0, 0) | Margin for the JButton. Refer to the javadoc of javax.swing.AbstractButton#setMargin(java.awt.Insets) for details of how the margin is used. |
| prefix.textShiftOffset | Integer | 0 | Amount to shift the text and icon, along the x and y axis, from where it would normally be layed out. This is only used when the button is in a pressed state and does not have a pressed icon. |

Prefix is one of: Button, CheckBox, RadioButton or JToggleButton.

#### Text Properties

| Property | Expected Type | Default Value | Description |
| --- | --- | --- | --- |
| prefix.caretForeground | Color | #000000 | Color of the caret. |
| prefix.margin | Insets | Empty Insets (0, 0, 0, 0),  EditorPane and TextPane (3, 3, 3, 3) | Margins of the text component. |
| prefix.caretBlinkRate | Integer | 500 | Number of milliseconds defining the blink rate fo the caret. |

Prefix is one of: EditorPane, FormattedTextField, PasswordField, TextArea, TextField or TextPane.