| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/javax/swing/plaf/multi/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/swing/table/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/synth/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.swing.plaf.synth

Synth is a skinnable look and feel in which all painting is delegated.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**SynthConstants**](http://docs.google.com/javax/swing/plaf/synth/SynthConstants.html) | Constants used by Synth. |

| **Class Summary** | |
| --- | --- |
| [**ColorType**](http://docs.google.com/javax/swing/plaf/synth/ColorType.html) | A typesafe enumeration of colors that can be fetched from a style. |
| [**Region**](http://docs.google.com/javax/swing/plaf/synth/Region.html) | A distinct rendering area of a Swing component. |
| [**SynthContext**](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) | An immutable transient object containing contextual information about a Region. |
| [**SynthGraphicsUtils**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html) | Wrapper for primitive graphics calls. |
| [**SynthLookAndFeel**](http://docs.google.com/javax/swing/plaf/synth/SynthLookAndFeel.html) | SynthLookAndFeel provides the basis for creating a customized look and feel. |
| [**SynthPainter**](http://docs.google.com/javax/swing/plaf/synth/SynthPainter.html) | SynthPainter is used for painting portions of JComponents. |
| [**SynthStyle**](http://docs.google.com/javax/swing/plaf/synth/SynthStyle.html) | SynthStyle is a set of style properties. |
| [**SynthStyleFactory**](http://docs.google.com/javax/swing/plaf/synth/SynthStyleFactory.html) | Factory used for obtaining SynthStyles. |

## Package javax.swing.plaf.synth Description

Synth is a skinnable look and feel in which all painting is delegated. Synth does not provide a default look. In order to use Synth you need to specify a [file](http://docs.google.com/doc-files/synthFileFormat.html), or provide a [SynthStyleFactory](http://docs.google.com/javax/swing/plaf/synth/SynthStyleFactory.html). Both configuration options require an understanding of the synth architecture, which is described below, as well as an understanding of Swing's architecture.

Unless otherwise specified null is not a legal value to any of the methods defined in the synth package and if passed in will result in a NullPointerException.

## Synth

Each [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) implementation in Synth associates itself with one [SynthStyle](http://docs.google.com/javax/swing/plaf/synth/SynthStyle.html) per [Region](http://docs.google.com/javax/swing/plaf/synth/Region.html), most Components only have one Region and therefor only one SynthStyle. SynthStyle is used to access all style related properties: fonts, colors and other Component properties. In addition SynthStyles are used to obtain [SynthPainter](http://docs.google.com/javax/swing/plaf/synth/SynthPainter.html)s for painting the background, border, focus and other portions of a Component. The ComponentUIs obtain SynthStyles from a [SynthStyleFactory](http://docs.google.com/javax/swing/plaf/synth/SynthStyleFactory.html). A SynthStyleFactory can be provided directly by way of [SynthLookAndFeel.setStyleFactory(javax.swing.plaf.synth.SynthStyleFactory)](http://docs.google.com/javax/swing/plaf/synth/SynthLookAndFeel.html#setStyleFactory(javax.swing.plaf.synth.SynthStyleFactory)), or indirectly by way of [SynthLookAndFeel.load(java.io.InputStream, java.lang.Class)](http://docs.google.com/javax/swing/plaf/synth/SynthLookAndFeel.html#load(java.io.InputStream,%20java.lang.Class)). The following example uses the SynthLookAndFeel.load() method to configure a SynthLookAndFeel and sets it as the current look and feel:

SynthLookAndFeel laf = new SynthLookAndFeel();  
 laf.load(MyClass.class.getResourceAsStream("laf.xml"), MyClass.class);  
 UIManager.setLookAndFeel(laf);

Many JComponents are broken down into smaller pieces and identified by the type safe enumeration in [Region](http://docs.google.com/javax/swing/plaf/synth/Region.html). For example, a JTabbedPane consists of a Region for the JTabbedPane ([Region.TABBED\_PANE](http://docs.google.com/javax/swing/plaf/synth/Region.html#TABBED_PANE)), the content area ([Region.TABBED\_PANE\_CONTENT](http://docs.google.com/javax/swing/plaf/synth/Region.html#TABBED_PANE_CONTENT)), the area behind the tabs ([Region.TABBED\_PANE\_TAB\_AREA](http://docs.google.com/javax/swing/plaf/synth/Region.html#TABBED_PANE_TAB_AREA)), and the tabs ([Region.TABBED\_PANE\_TAB](http://docs.google.com/javax/swing/plaf/synth/Region.html#TABBED_PANE_TAB)). Each Region of each JComponent will have a SynthStyle. This allows you to customize individual pieces of each region of each JComponent.

Many of the Synth methods take a [SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html). This is used to provide information about the current Component and includes: the [SynthStyle](http://docs.google.com/javax/swing/plaf/synth/SynthStyle.html) associated with the current [Region](http://docs.google.com/javax/swing/plaf/synth/Region.html), the state of the Component as a bitmask (refer to [SynthConstants](http://docs.google.com/javax/swing/plaf/synth/SynthConstants.html) for the valid states), and a [Region](http://docs.google.com/javax/swing/plaf/synth/Region.html) identifying the portion of the Component being painted.

All text rendering by non-JTextComponents is delegated to a [SynthGraphicsUtils](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html), which is obtained using the [SynthStyle](http://docs.google.com/javax/swing/plaf/synth/SynthStyle.html) method [SynthStyle.getGraphicsUtils(javax.swing.plaf.synth.SynthContext)](http://docs.google.com/javax/swing/plaf/synth/SynthStyle.html#getGraphicsUtils(javax.swing.plaf.synth.SynthContext)). You can customize text rendering by supplying your own [SynthGraphicsUtils](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html).

## Notes on specific components

### JTree

Synth provides a region for the cells of a tree: Region.TREE\_CELL. To specify the colors of the renderer you'll want to provide a style for the TREE\_CELL region. The following illustrates this:

<style id="treeCellStyle">  
 <opaque value="TRUE"/>  
 <state>  
 <color value="WHITE" type="TEXT\_FOREGROUND"/>  
 <color value="RED" type="TEXT\_BACKGROUND"/>  
 </state>  
 <state value="SELECTED">  
 <color value="RED" type="TEXT\_FOREGROUND"/>  
 <color value="WHITE" type="BACKGROUND"/>  
 </state>  
 </style>  
 <bind style="treeCellStyle" type="region" key="TreeCell"/>

This specifies a color combination of red on white, when selected, and white on red when not selected. To see the background you need to specify that labels are not opaque. The following XML fragment does that:

<style id="labelStyle">  
 <opaque value="FALSE"/>  
 </style>  
 <bind style="labelStyle" type="region" key="Label"/>

### JList and JTable

The colors that the renderers for JList and JTable use are specified by way of the list and table Regions. The following XML fragment illustrates how to specify red on white, when selected, and white on red when not selected:

<style id="style">  
 <opaque value="TRUE"/>  
 <state>  
 <color value="WHITE" type="TEXT\_FOREGROUND"/>  
 <color value="RED" type="TEXT\_BACKGROUND"/>  
 <color value="RED" type="BACKGROUND"/>  
 </state>  
 <state value="SELECTED">  
 <color value="RED" type="TEXT\_FOREGROUND"/>  
 <color value="WHITE" type="TEXT\_BACKGROUND"/>  
 </state>  
 </style>  
 <bind style="style" type="region" key="Table"/>  
 <bind style="style" type="region" key="List"/>

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/javax/swing/plaf/multi/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/swing/table/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/synth/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).