

Nimbus BOF

Jasper Potts & Richard Bair
Swing Team • Sun Microsystems

Topics

- States
- Native Keybindings
- Custom Renderers
- Derived Colors
- 3rd Party Component Development

States

- Standard States
- Custom States
- State Ordering

Standard States

- Enabled
- Mouse Over
- Pressed
- Disabled
- Focused
- Selected
- Default*

Custom States

- Extend `com.sun.java.swing.plaf.nimbus.State`
- Implement `isInState` method

isInState(T c)

```
/**
 * <p>Gets whether the specified JComponent is in the custom state represented
 * by this class. <em>This is an extremely performance sensitive loop.</em>
 * Please take proper precautions to ensure that it executes quickly.</p>
 *
 * <p>Nimbus uses this method to help determine what state a JComponent is
 * in. For example, a custom State could exist for JProgressBar such that
 * it would return <code>>true</code> when the progress bar is indeterminate.
 * Such an implementation of this method would simply be:</p>
 *
 * <pre><code> return c.isIndeterminate();</code></pre>
 *
 * @param c the JComponent to test. This will never be null.
 * @return true if <code>c</code> is in the custom state represented by
 *         this <code>State</code> instance
 */
protected abstract boolean isInState(T c);
```

State Ordering

- Order of states is important
 - State Resolution

Registering States

- `Button.States = Enabled, Disabled, Foo`
- `Button.Foo = new MyFoo();`
- `Button[Foo].backgroundPainter = ...`

Native Keybindings

- Principle: Native Feel, Custom Look
- Current support Windows & Unix
- Working with Apple on Mac support

Custom Renderers

- *Always* extend Default renderers
- Nimbus needed extra hooks into renderers

Derived Colors

- Derive based on hue, saturation, brightness
- Parent color based on UIDefault key

Derived Colors

```
/**
 * Get a derived color, derived colors are shared instances and its color
 * value will change when its parent UIDefault color changes.
 *
 * @param uiDefaultParentName The parent UIDefault key
 * @param hOffset             The hue offset
 * @param sOffset             The saturation offset
 * @param bOffset             The brightness offset
 * @param aOffset             The alpha offset
 * @param uiResource          True if the derived color should be a
 *                            UIResource, false if it should not be
 * @return The stored derived color
 */
public Color getDerivedColor(String uiDefaultParentName,
                             float hOffset, float sOffset,
                             float bOffset, int aOffset,
                             boolean uiResource) {
```

3rd Party Comps

- Create UI Delegate
- Register with NimbusLookAndFeel
- Use NimbusStyle
- Delegate to Painters