BatikRenderingSummary

A Summary of Using the JSVGCanvas Class to Display a SVG Document

First of all it's important to note the class inheritance hierarchy being used here since several methods called from the JSVGCanvas class are actually implemented by some of the classes it inherits from.

JSVGCanvas : JSVGComponent : JGVTComponent : JComponent

After creating a JSVGCanvas instance the setURI() method is called with a URI of the SVG document to load. If the URI is valid the loadSVGDocument() method of the JSVGComponent class is called with the URI.

In the loadSVGDocument() method all document processing for the current document is stopped, and then a DocumentLoader and SVGDocumentLoader are created. The SVGDocumentLoader handles the firing of document loading events and the DocumentLoader uses a SAXSVGDocumentFactory to create a SVGDocument instance. The SAXSVGDocumentFactory in turn relies on a SAXDocumentFactory to parse the SVG data referenced by the given URI.

The JSVGCanvas relies on the SVGDocumentLoaderListener, GVTTreeBuilderListener, and SVGLoadEventDispatcherListener listeners that are added in the JSVGComponent constructor in order to know what is going on with the loading, building, and rendering of the SVGDocument. The listeners are implemented in the JGVTComponent.Listener and JSVGComponent.SVGListener classes.

When the document has finished being loaded and parsed the setSVGDocument() method of the JSVGComponent class is called, which then calls the installSVGDocument() method in the same class.

The installSVGDocument() method:

1. Gets rid of any resources from the previous document (if any)

2. Create a new BridgeContext to associate the SVG DOM and the GVT tree

3. Configures the component and BridgeContext instance based on the document state (ALWAYS_STATIC, ALWAYS_DYNAMIC, ALWAYS_INTERACTIVE, AUTODETECT)

4. Creates a GVTTreeBuilder with the SVGDocument and BridgeContext

5. Initializes event handling for mouse and keyboard events

When the GVTTreeBuilder is done building the GVT tree the JSVGComponent performs some additional configuration so that dynamic documents work correctly, and then calls the scheduleGVTRendering() method of the JGVTComponent class.

The scheduleGVTRendering() method calls the renderGVTTree() method of the same class and:

1. Sets the size of the visible rect to be the size of the component

2. Creates either a static or dynamic image rendered based on the document state (StaticRenderer or DynamicRenderer)

3. Finds the inverse of the rendering transform, which is set to the identity matrix

4. Creates a shape by transforming the rectangle found in step 1 with the matrix from step 3

5. Creates a GVTTreeRenderer and starts the rendering thread

The GVTTreeRenderer fires GVT tree rendering events and calls the repaint() method of the renderer. Once repainting is complete, the JGVTComponent is notified of the fact by a GVT tree rendering event, and in response repaints the component using the updated image that was rendered offscreen by the renderer.