

ORACLE®



JavaOne™

ORACLE®

Introduction To JavaFX Scenic View

Jonathan Giles
Principle Member of Technical Staff
Java Client Group
October, 2015



Safe Harbor Statement

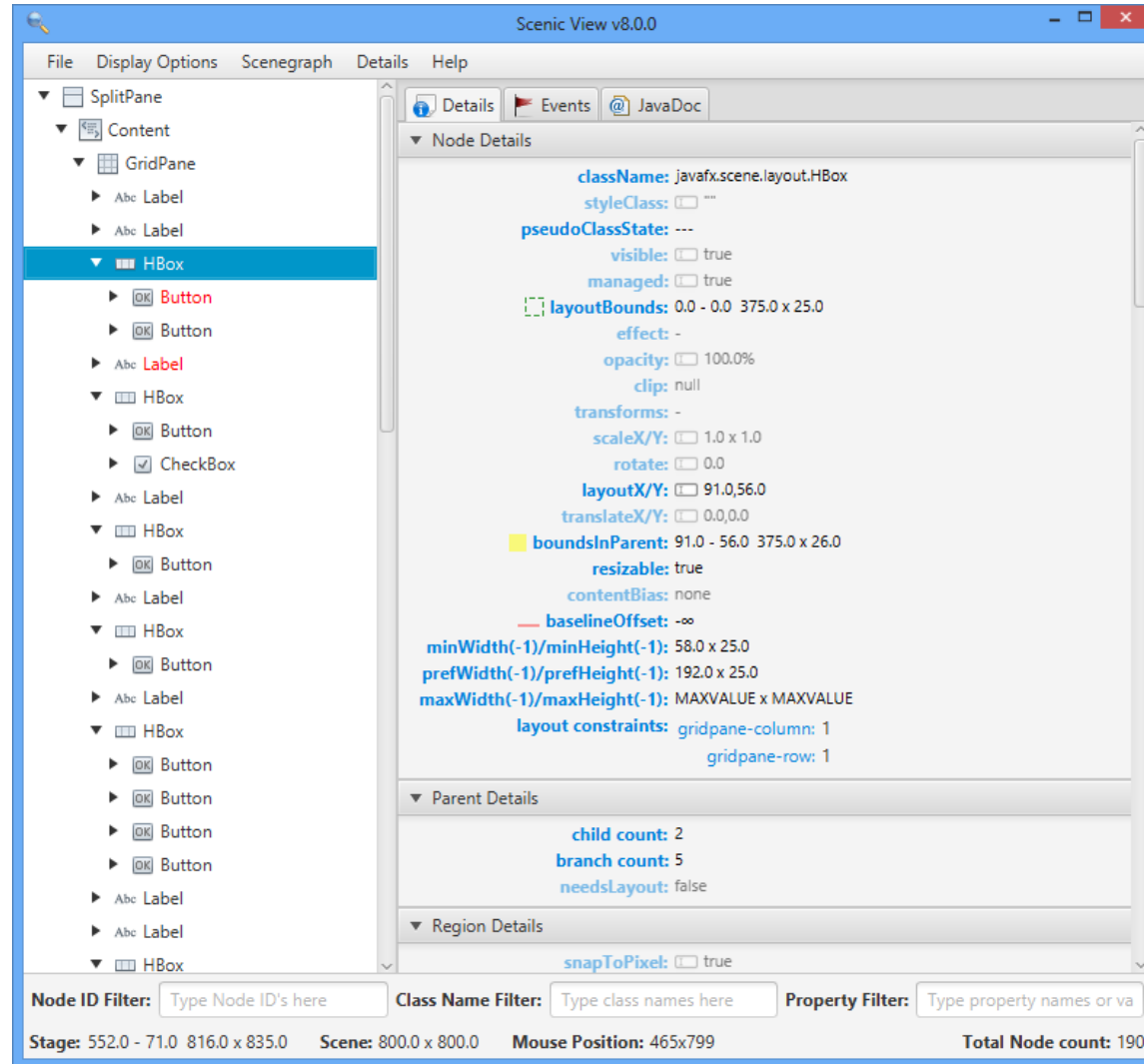
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Warning: This is almost last years BOF.
If you attended last year, you may not want to
stick around...

Scenic View in a Nutshell

Scenic View is to JavaFX what
Firebug is to websites

Scenic View in a Nutshell



Scenic View in a Nutshell

- Scenic View is a free JavaFX scenegraph analyser.
- I develop Scenic View,
 - when time permits,
 - it is not my job!
- Download and find out more about Scenic View here:
<http://www.scenic-view.org>

Scenic View

- What is Scenic View?
 - Originally built by Amy Fowler for diagnosing runtime issues with UI layout
 - It was really simple to use:
just add `ScenicView.show(scene)` in your code
- I took Amy's code and polished the UI considerably before the first public release of Scenic View on May 6, 2012
- It looked a little like this:



Scenic View Pre-1.0.0

Tree showing scenegraph structure of running application

Application overview

The screenshot shows the Scenic View application window. On the left is a tree view of the scenegraph structure. The 'Button' node is selected and highlighted in blue. On the right is the 'Node Details' panel, which displays the properties of the selected node. The properties include styleClass, visible, managed, layoutBounds, effect, opacity, clip, transforms, scaleX/Y, rotate, layoutX/Y, translateX/Y, boundsInParent, resizable, contentBias, baselineOffset, and layout constraints. Below the Node Details panel is the 'Parent Details' panel, which shows child count, branch count, and needsLayout. At the bottom of the window, there is a status bar with the following information: Stage 604.0,-462.0 516.0 x 419.0, Scene: 500.0 x 381.0, and Total Node count: 39.

```
▼ TilePane
  Rectangle
  Rectangle
  Rectangle
  ▼ Group
    Rectangle
  ► Button
  ▼ Button (selected)
    ▼ ButtonSkin
      LabeledText
    ► Button
  ▼ Group
    ► Button
  ▼ Group
    Circle
    Circle
    Circle
  ▼ Group
    Circle
    Circle
    Circle
```

▼ Node Details

- styleClass: "button"
- visible: true
- managed: true
- layoutBounds: 0.0,0.0 89.0 x 32.0
- effect: DropShadow
- opacity: 100.0%
- clip: null
- transforms: -
- scaleX/Y: 1.0 x 1.0
- rotate: 0.0
- layoutX/Y: 146.0,175.0
- translateX/Y: 0.0,0.0
- boundsInParent: 117.0,146.0 147.0 x 91.0
- resizable: true
- contentBias: none
- baselineOffset: 22.42
- minWidth(-1)/minHeight(-1): 42.0 x 31.94
- prefWidth(-1)/prefHeight(-1): 89.0 x 31.94
- maxWidth(-1)/maxHeight(-1): 89.0 x 31.94
- layout constraints:

▼ Parent Details

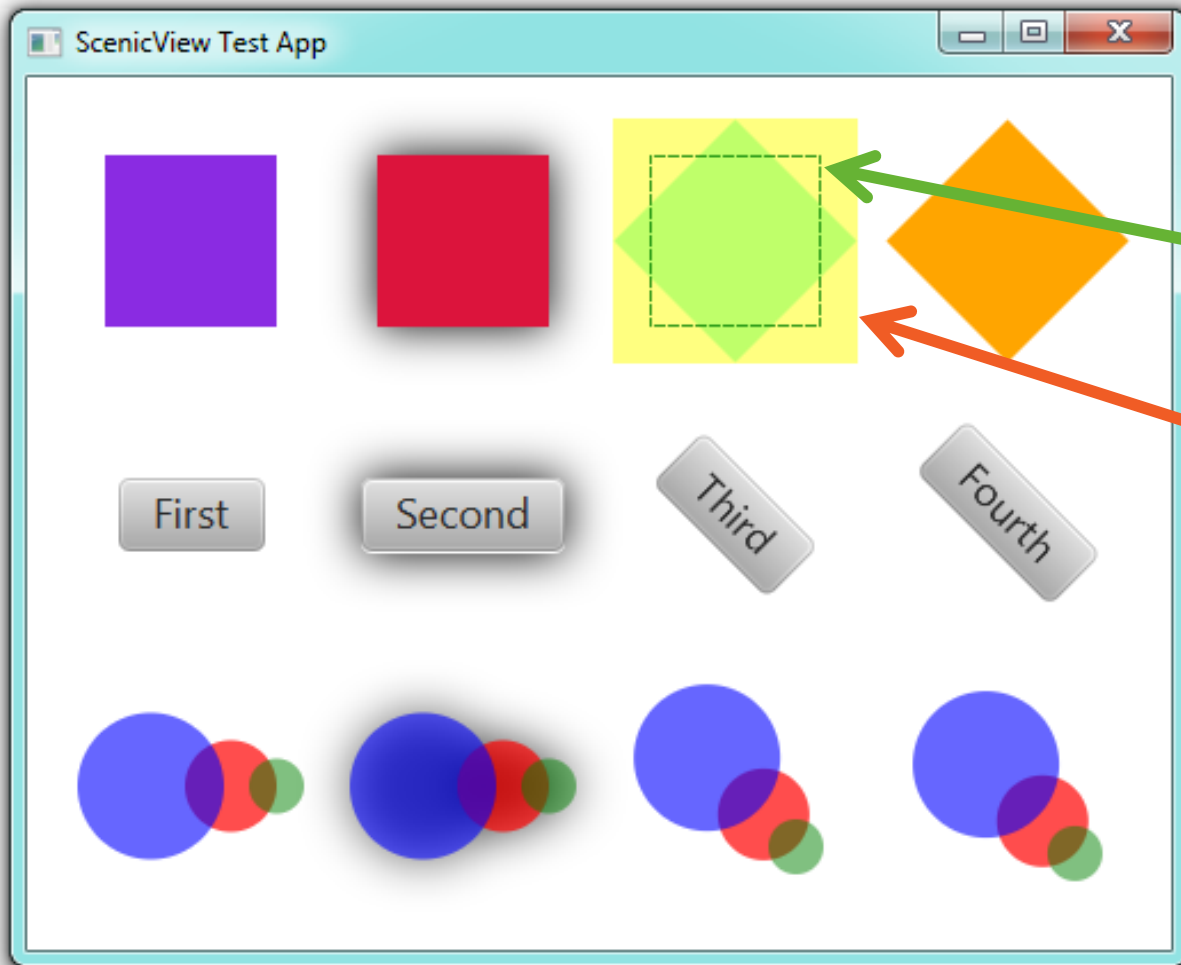
- child count: 1
- branch count: 3
- needsLayout: false

▼ Control Details

Stage 604.0,-462.0 516.0 x 419.0 Scene: 500.0 x 381.0 Total Node count: 39

The most important properties for the selected node

Scenic View



Scenic View can also draw overlays in the application it is observing.

The green dashed rectangle shows the `layoutBounds`, and the yellow filled rectangle shows the `boundsInParent`.

This can be very useful for debugging.

Scenic View

- The response to the public release of Scenic View was extremely positive.
- Most feedback was of the form: “I love it, but it needs to do X”
- X included:
 - Live editing ✓
 - Filtering ✓
 - Selecting nodes by clicking in the UI ✓
 - Event tracing ✓

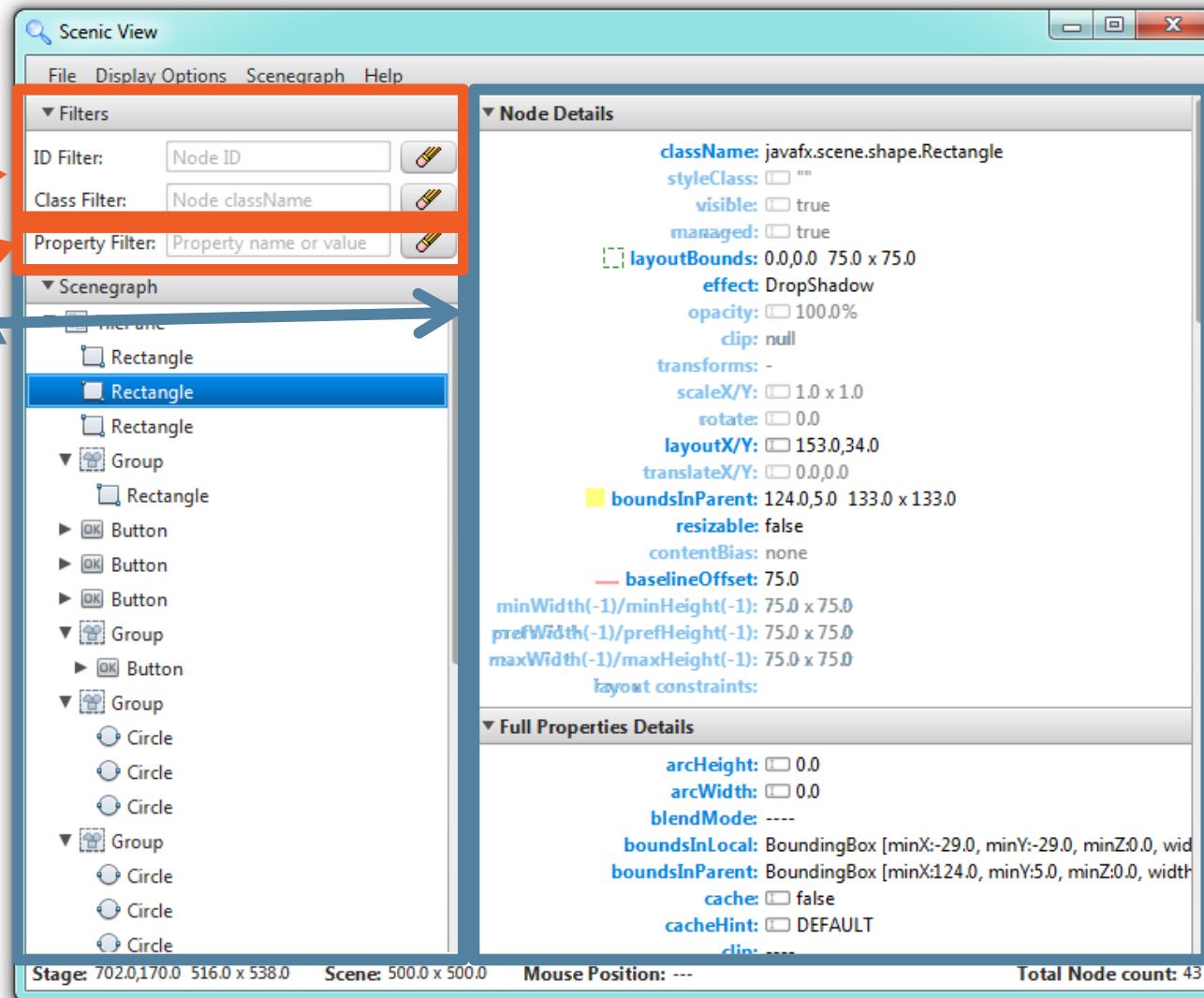
Scenic View 1.0.0

- Ander Ruiz contacted me after the first release, and had a number of ideas.
- Even better, he was keen to help program them!
- Together we made available Scenic View 1.0.0 on June 4, 2012
- The first versioned release of Scenic View
- It looked a little like this:



Scenic View 1.0.0

Ability to filter the tree area by typing in an ID or class name.
Ability to filter the details area by typing in a property name



The Problem with Scenic View 1.0.0

- The major complaint Ander and I heard from users:
 - People did not want to have to modify their code by adding `ScenicView.show(scene)`.
- This proved an interesting (and complex) problem to resolve!
- We needed a way to connect to applications at runtime without any modification of their code.
- We settled on two solutions

Solution 1: Java Agents

- The Java agent API allows for an external library to be called when an application starts.
- Simply add the following when starting your application:
 - `javaagent:ScenicView.jar`
- Scenic View will start when your application starts
- It will automatically discover all stages in your application

- Best approach: in your IDE have two 'run' profiles, one with Scenic View enabled and the other without

Solution 2: Java Attach API

- Java provides the Attach API to discover running Java applications
- We use this to install a small socket server into your application at runtime, through which Scenic View can communicate
- This means that Scenic View can discover all running JavaFX applications and you don't need to do anything!
- To use this solution, simply start Scenic View directly and it'll start in this mode

Scenic View 1.1.0

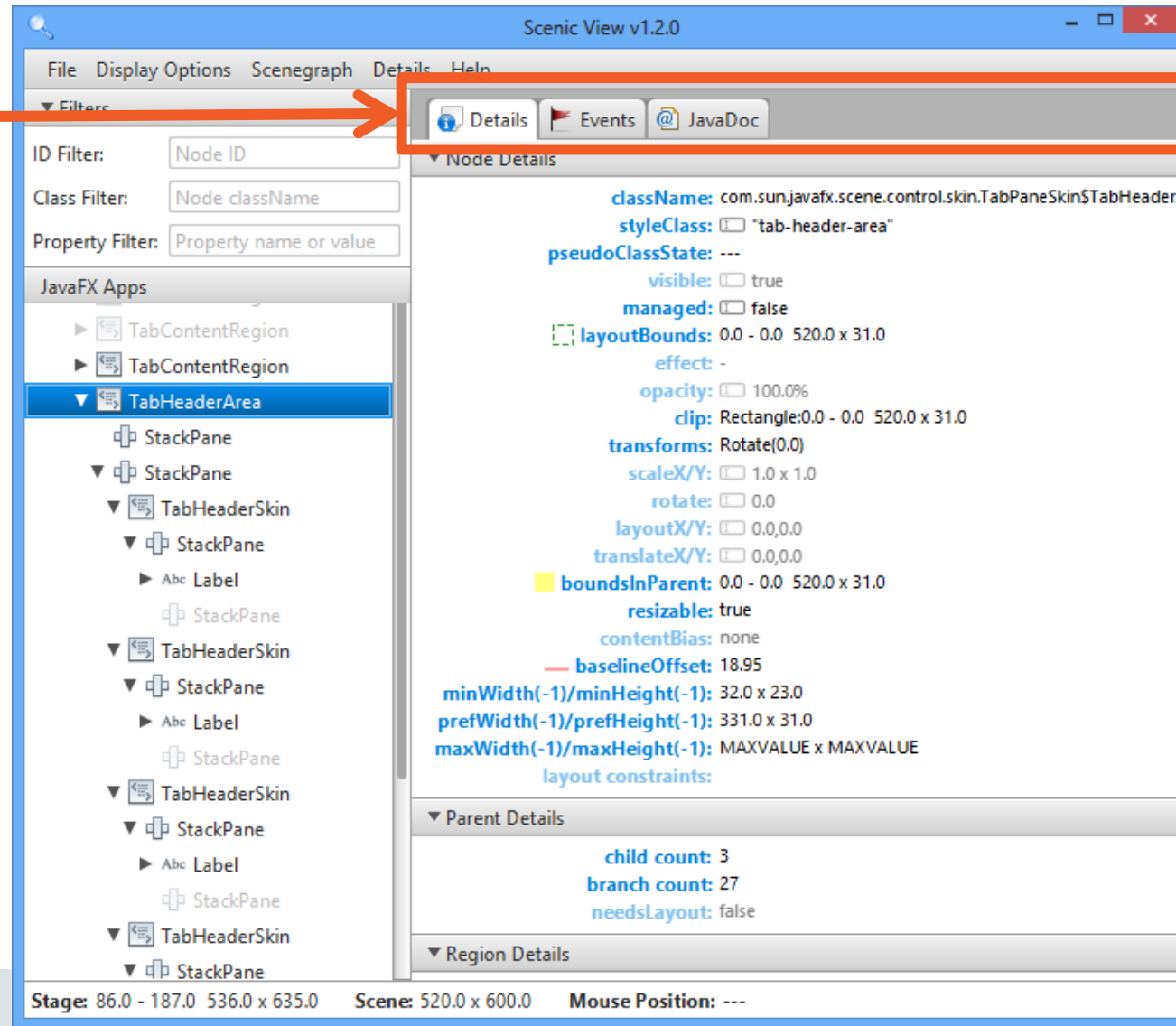
- Scenic View 1.1.0 was released on August 14th, 2012 after much testing and user feedback
- This release required a massive amount of reworking and foundation building.
- We were incredibly relieved to get this working on Windows, Mac OS and Linux!
- However, things are never perfect, and Scenic View 1.1.1 was released on August 16th, 2012
- This improved our ability to debug peoples issues.

Scenic View 1.2.0

- Still, we knew there was more to do, so we carried on and released Scenic View 1.2.0 on September 25th, 2012.
- This release included:
 - Event tracing support
 - JavaDoc browsing support
 - Streamlined menus (context menus)
 - Bug fixes!

Scenic View 1.2.0

Tabbed area for
new
functionality



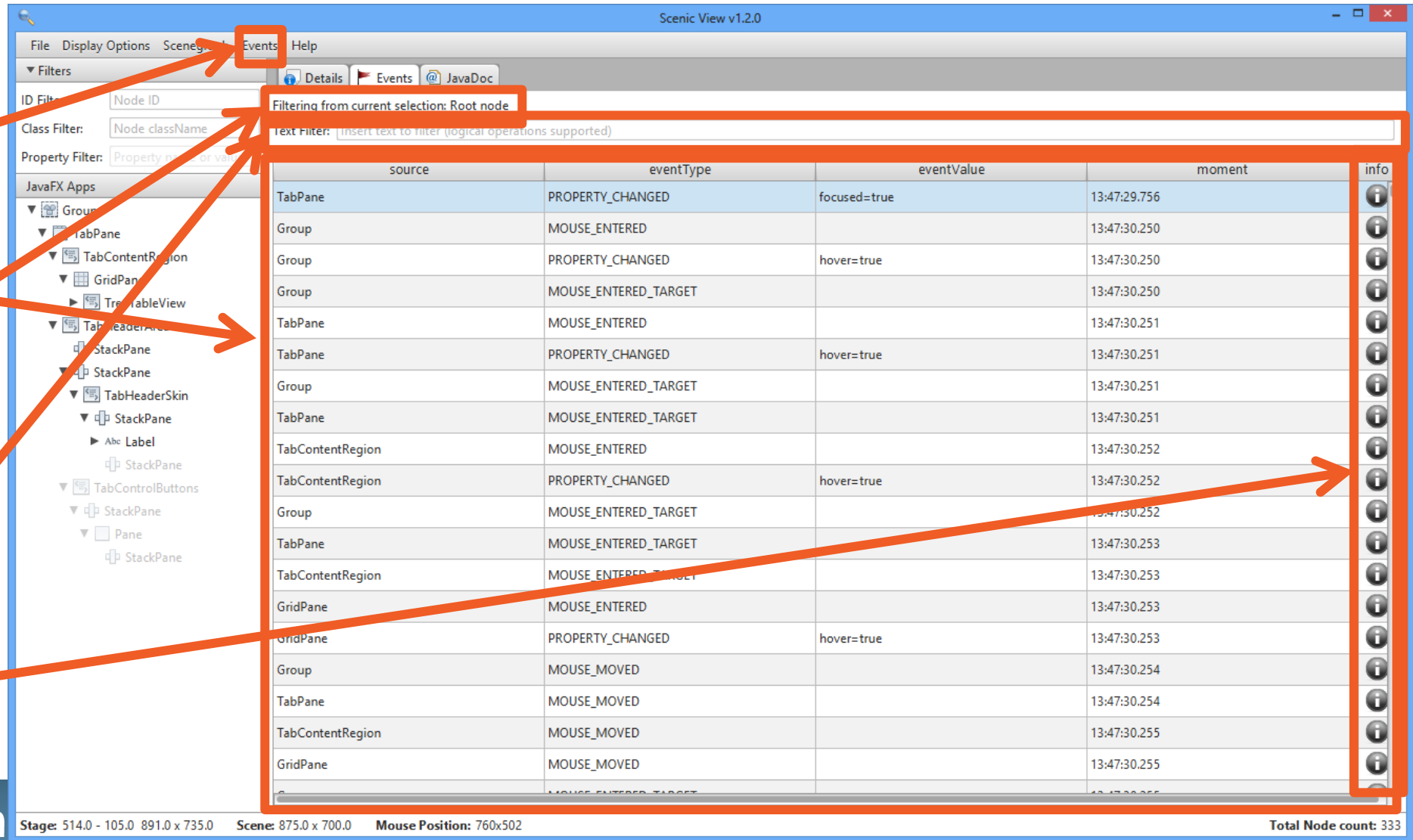
Scenic View 1.2.0: Event Tracing

Event tracing is enabled from the Events menu

Events are all recorded dialogly from selected node down

Events can be searched using boolean statements

Click the info button for the entire stacktrace



The screenshot shows the Scenic View v1.2.0 application window. The 'Events' menu is highlighted in the top menu bar. Below it, the 'Filters' section is visible, including 'ID Filter', 'Class Filter', and 'Property Filter'. The 'Text Filter' is set to 'Filtering from current selection: Root node'. The main area displays a table of events with columns for 'source', 'eventType', 'eventValue', 'moment', and 'info'. The 'info' column contains circular icons for each event. The bottom status bar shows 'Stage: 514.0 - 105.0 891.0 x 735.0', 'Scene: 875.0 x 700.0', 'Mouse Position: 760x502', and 'Total Node count: 333'.

source	eventType	eventValue	moment	info
TabPane	PROPERTY_CHANGED	focused=true	13:47:29.756	i
Group	MOUSE_ENTERED		13:47:30.250	i
Group	PROPERTY_CHANGED	hover=true	13:47:30.250	i
Group	MOUSE_ENTERED_TARGET		13:47:30.250	i
TabPane	MOUSE_ENTERED		13:47:30.251	i
TabPane	PROPERTY_CHANGED	hover=true	13:47:30.251	i
Group	MOUSE_ENTERED_TARGET		13:47:30.251	i
TabPane	MOUSE_ENTERED_TARGET		13:47:30.251	i
TabContentRegion	MOUSE_ENTERED		13:47:30.252	i
TabContentRegion	PROPERTY_CHANGED	hover=true	13:47:30.252	i
Group	MOUSE_ENTERED_TARGET		13:47:30.252	i
TabPane	MOUSE_ENTERED_TARGET		13:47:30.253	i
TabContentRegion	MOUSE_ENTERED_TARGET		13:47:30.253	i
GridPane	MOUSE_ENTERED		13:47:30.253	i
GridPane	PROPERTY_CHANGED	hover=true	13:47:30.253	i
Group	MOUSE_MOVED		13:47:30.254	i
TabPane	MOUSE_MOVED		13:47:30.254	i
TabContentRegion	MOUSE_MOVED		13:47:30.255	i
GridPane	MOUSE_MOVED		13:47:30.255	i
GridPane	MOUSE_ENTERED_TARGET		13:47:30.255	i

Scenic View 1.2.0: JavaDoc Browser

Browser shows
JavaDoc for
currently
selected node

The screenshot shows the Scenic View v1.2.0 interface. The 'JavaDoc' tab is selected in the top navigation bar. The left sidebar shows a tree view of JavaFX classes, with 'TabPane' selected and highlighted in blue. The main content area displays the JavaDoc for 'Class TabPane', including its inheritance hierarchy, implemented interfaces, and source code. The source code is as follows:

```
@DefaultProperty(value="tabs")
public class TabPane
extends Control

A control that allows switching between a group of Tabs. Only one tab is visible at a time. Tabs are added to the TabPane by using the getTabs ().

Tabs in a TabPane can be positioned at any of the four sides by specifying the Side.

A TabPane has two modes floating or recessed. Applying the styleclass STYLE_CLASS_FLOATING will change the TabPane mode to floating.

The tabs width and height can be set to a specific size by setting the min and max for height and width. TabPane default width will be determined by the largest content width in the TabPane. This is the same for the height. If a different size is desired the width and height of the TabPane can be overridden by setting the min, pref and max size.

When the number of tabs do not fit the TabPane a menu button will appear on the right. The menu button is used to select the tabs that are currently not visible.

Example:

TabPane tabPane = new TabPane();
Tab tab = new Tab();
tab.setText("new tab");
tab.setContent(new Rectangle(200,200, Color.LIGHTSTEEELBLUE));
tabPane.getTabs().add(tab);
```

At the bottom of the window, the status bar displays: Stage: 514.0 - 105.0 891.0 x 735.0 Scene: 875.0 x 700.0 Mouse Position: --- Total Node count: 333

Scenic View 1.3.0

- Scenic View 1.3.0 was released on November 12, 2012
- What was new?
 - Massive performance gains
 - Animation tracer
 - Improved CSS support
 - Version update checking
 - Mac native menubar integration
 - Bug fixes and miscellaneous improvements

Beyond Scenic View 1.3.0

- Scenic View 1.3.0 was the last release with support for JavaFX 2.x.
- Unfortunately at this stage Ander had to drop out due to work commitments

Scenic View 8.0.0

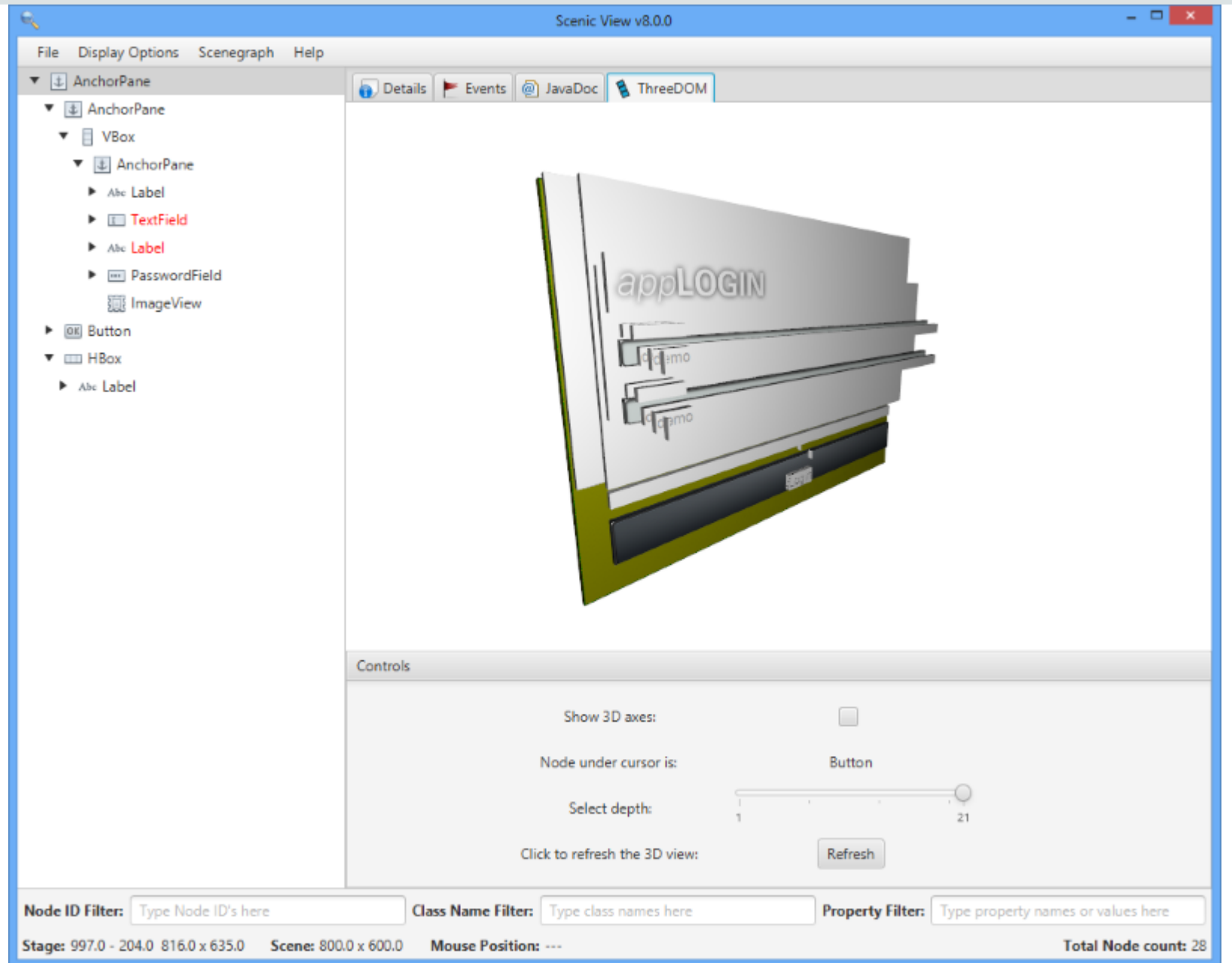
- Released and open sourced at JavaOne 2014
- GPL licensed
- Code repository is open here:
<https://bitbucket.org/scenicview/scenic-view>

Scenic View 8.6.0

- Released September 2nd, 2015
- Primarily consists of two very cool community contributions :
 - ThreeDOM
 - CSSFX

ThreeDOM

3D 'explosion' view of user interface



CSSFX

- Ability to edit / save CSS files and have them be dynamically reloaded at runtime without needing to restart application.

Downloads

- I track downloads of Scenic View.
- The results are pleasing (as of October 20th, 2015):

Version	Release Date	Downloads
1.0.0	4 th June, 2012	208
1.0.1	7 th June, 2012	859
1.1.0	14 th August, 2012	575
1.1.1	16 th August, 2012	539
1.2.0	25 th August, 2012	932
1.3.0	9 th November, 2012	7164
8.0.0	29 th September, 2014	8103
8.6.0	2 nd September, 2015	1591

Scenic View Demo

Getting Started

- In Six Simple Steps!

1 The following software is required:

- Mercurial
- Gradle
- JDK 8

2 You'll need a Bitbucket account

- Accounts are free from <http://bitbucket.org>



Getting Started

- In Six Simple Steps!

3 Fork the repo.

Go here to create your own fork

– <https://bitbucket.org/scenicview/scenic-view/fork>

4 Clone your fork:

– `hg clone https://<username>@bitbucket.org/<username>/<forkname>`

– e.g.

• `hg clone https://jonathangiles@bitbucket.org/jonathangiles/scenic-view`

Getting Started

- In Six Simple Steps!

5 Build your clone. From clone root directory, run:
– gradle clean assemble

6 Run your clone:
– gradle run

The Future of Scenic View

- What else is there left to do?
- Should I pack my bags and stop developing Scenic View now?
- Is there a feature you'd love to have?
 - Tell me!
 - Email me and let me know at jonathan.giles@oracle.com
- Some ideas:
 - Pulse logger support (pulse duration, time since last pulse, etc)
 - Less bugs, faster, better UI, etc

The Future of Scenic View

- Even better – please join in and help me to develop it!
- Fork the project on bitbucket and do pull requests

Thanks for Attending!

It's Discussion Time!

How to contact me:

jonathan@jonathangiles.net

@JonathanGiles

Hardware and Software Engineered to Work Together



JavaOne™

ORACLE®

ORACLE®