

Java FX 2.0

Dr. Stefan Schneider
Oracle Deutschland
Walldorf-Baden

Keywords:

JavaFX, Rich, GUI, Road map.

Introduction

This presentation gives an introduction into JavaFX. It introduces the key features of JavaFX 1.3 and it gives an overview about the new strategy for JavaFX 2.0 which will be fully integrated into Java.

JavaFX combines the power of the rich Java infrastructure with the ease to create state of the art user clients including movies, sound and 3D animation.

The goal with JavaFX 2.0 is to create a fully Java compatible environment which allows to create quickly appealing user front ends by using the latest animation technologies in order to be ready for today's and future customer requirements.

Target Developers

JavaFX 2.0 is aiming at the existing 9 million Java developers. The move is fairly obvious and a change to the strategy applied through Sun before the acquisition.

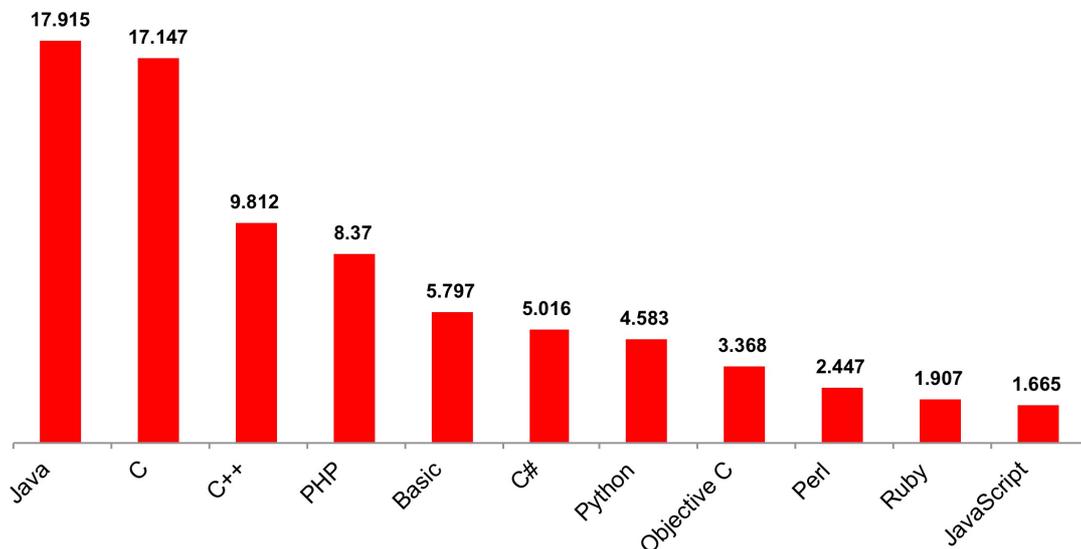


Illustration. 1: TIOBE Index, Popularity of Programming Languages

Most Flex developers have been developing user interfaces earlier on by using Swing. Many of them left the Java platform due to the lack of richness which has been provided by Flash or due to the

potential costs involved in developing Swing applications. JavaFX is overcoming these shortages of Swing while offering a very rich eco system around Java.

JavaFX is targeting as well web developers who are familiar with customization technologies like CSS or the power of html5. JavaFX 2.0 will seamlessly integrate into these environments and help web developers to create rich web applications with JavaFX 2-0 components.

Finally, we want to attract new developers. We want folks who've never written a line of code to be able to build something in JavaFX with only minimal help.

Having a good scripting language is pretty important for this group (otherwise step 1 is to teach OO which is a bit much for somebody who's never written a line of code). We want our APIs to be approachable, for things to just sort of work. We want our system so well designed that it feels like it was the only obvious way to have built the platform, where people don't wonder why we did things the way we did but it just seems "duh".

Value Proposition

JavaFX defines itself as being a well designed, cross platform application development platform. It is fast and flexible. It delivers exceptional value to developers and end users. It is competitive. Most importantly, it leverages the massive existing Java ecosystem. Nobody else offers the complete package – exceptional developer experience, user experience, cross platform development and massive ecosystem.

The advantages of JavaFX are that it is an evolution of the Java graphic user interface technologies. It allows to leverage the existing Java investments in regards of

- available software libraries
- integrated development environment (IDEs)
- existing Java skills in the enterprise and at universities
- full end to end infrastructure of Java web infrastructure

All these benefits make it easier to find developers and it lowers the risk of failure in projects since the total complexity is going down.

JavaFX encourages companies to use Java on the client and the server. A strong and vibrant client platform retards the growth of competing technologies in enterprise server rooms. JavaFX provides a compelling end-to-end Java solution for enterprises and competes well with other technologies.

Programming Languages

Java is the primary programming language for JavaFX. It allows to fully leverage the Java eco system. All future JavaFX APIs will be released as Java APIs. All JavaFX bindings, sequences will be exposed as Java APIs.

JavaFX will embrace all languages supported by the JavaVM and it'll support

- Jruby
- Jython
- Scala
- Groovy
- JavaScript
- etc

The current JavaFX Script compiler will not be updated to work with JavaFX 2.0 APIs. Existing JavaFX Script based applications will continue to run in the new environment.

When JavaFX 2.0 comes out, there will not be any regression in functionality. Indeed, we will be releasing a lot more functionality in addition to this port. There is a chance that not all features will be released in the initial release. The release of all features will be staggered over time.

APIs and Programming Model

JavaFX 2.0 will be a continuation from JavaFX 1.X product line. The scene graph and event model have been working very well in the past. Most APIs will simply be ported directly to Java. Some APIs will be revisited (e.g. layout). Existing Controls will all be ported and JavaFX 2.0 will embrace more web technology:

- Update JavaFX CSS to include more from CSS 3 (eventually to be fully spec compliant)
- Use WAI-ARIA for accessibility API
- Make HTML available for rich text in all Text nodes

Workflows

JavaFX will support the developer workflow as well as the designer workflow.

This means for a developer oriented workstyle:

- Developers write application, produce UI
- Use a RAD tool to produce forms apps quickly
- Import graphics from Illustrator & Photoshop

Designers however style UI using CSS. Their workflow is more the following one:

- Designer produces graphics using professional tools
- Developers produce modules (e.g. access web services)
- Designer aggregates graphics & code modules & scripts it together using JavaScript or another dynamic language

Web Integration

The future web integration is one of the key announcements for JavaFX 2.0.

JavaFX will be able to embed HTML. This will happen through a WebView node in the scene graph which is an embedded browser. This will allow all Text nodes to have HTML content and it offers a seamless DOM integration.

The integration in the opposite direction is possible as well: JavaFX can be embedded in HTML using traditional plugin technology. This offers as well seamless DOM integration. JavaFX is planned to be adopted as well to use HTML 5 local storage, browser history, etc

Other web standards which are planned to be adopted:

- CSS
- Accessibility
- HTML for rich text, ...

Graphics

In JavaFX 1.3 we have support for 3D transforms, and in fact there are a lot of cool things you can do with it. However, we will be adding full 3D support over the next several releases to JavaFX. We don't see JavaFX as primarily a game platform – that is, we aren't competing with Crysis or Havoc or other game engines. We do however see it as a very powerful tool for transitions, marketing campaigns, casual games, data visualization, and so forth.

The focus of the current work is on:

- 2D and 3D scene graph
- Hardware accelerated pipeline
 - DirectX 9 on Windows XP, Windows Vista
 - DirectX 11 on Windows 7
 - OpenGL on Mac, Linux, Embedded
 - Software (Java2D) when necessary
- Embed in Swing
 - c.f. Java3D, JOGL
 - Allows Swing applications to add JavaFX scenes, charts, media, and controls
- High Definition Media

JavaFX Architecture

The JavaFX architecture will leverage the Java VM. All JavaFX APIs exposed will use the Prism toolkit. The tool itself will rely on the operation system Windowing system and graphics card features. All features will be mapped by the HotSpot Java V onto operating system features.

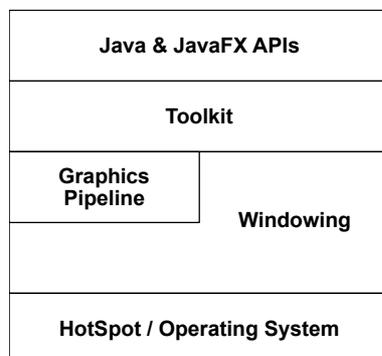


Illustration. 2: JavaFX Architecture

Key concepts of the architecture are:

- to be modular from the start
- design for maximum code reuse
- clean interface separating APIs from implementation
 - improves ability to test the system
 - easier to handle large development team

JavaFX Scene Graph APIs

The JavaFX Scene Graph APIs are being used by

- a full set of rich UI controls
- customizable charts

The JavaFX Scene APIs offer support for

- drag and drop Operations
- mouse, touch, key board input methods
- graphical effects like reflections, shadows, glows etc
- timeline based animations
- painting with gradients and texture paints

- 2D and 3D transformations

Controls	Charts
Scene Graph	
Drag & Drop	Animations
Input Events	Paints
Effects	Transforms

Illustration. 3: JavaFX Scene Graph APIs

Graphics

The prism tool kit is JavaFX's graphic foundation to support the next-generation hardware accelerated graphics including

- Support OpenGL ES 2, DirectX 9, DirectX 11
- DirectX used on Windows due to superiority of drivers
- Falls back to Java2D for software rendering as needed

JavaFX 2.0 will offer full 3D supported for OpenGL and Direct. Some functions however may have to be supported by a software pipe. Some 3D such as transforms supported with software pipe.

Java & JavaFX APIs			
Prism Toolkit			
Prism Render Tree			
Java2D	DirectX	OpenGL	
Glass			

Illustration. 4: Prism Tool Kit Architecture

Product Time Line and Road Map

The current plan is to have JavaFX early access release in Q1 2011 followed by a Beta release and a final GA version in Q3 2011. The latest information about the road map can be found under:

<http://javafx.com/roadmap>

This documents as well all planned features of JavaFX 2.0

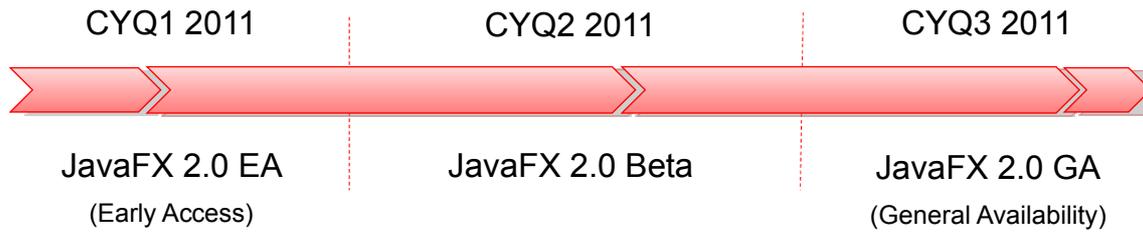


Illustration. 5: JavaFX 2.0 Road Map

Contact address:

Name

Dr. Stefan Schneider
Oracle Deutschland
Altrottstr. 31
69168 Walldorf

Phone: +49(0)6227 356 235
Fax: +49(0)6227 356 222
Email: stefan.schneider@oracle.com
Internet: <http://www.oracle.de>