Learn about the JavaFX 2 and later technology, read a feature summary, explore the sample applications, and follow the high-level steps to create and deploy a JavaFX application.
Contents

1 What Is JavaFX?
JavaFX Applications ........................................................................................... 1-1
Availability ............................................................................................................ 1-2
Key Features .......................................................................................................... 1-2
What Can I Build with JavaFX? ............................................................................. 1-3
How Do I Run a Sample Application? ................................................................. 1-3
How Do I Run a Sample in an IDE? .................................................................. 1-4
How Do I Create a JavaFX Application? ............................................................ 1-4
How Do I Deploy a JavaFX Application? ............................................................ 1-5
Resources ............................................................................................................. 1-5
What Is JavaFX?

JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms.

- JavaFX Applications
- Availability
- Key Features
- What Can I Build with JavaFX?
- How Do I Run a Sample Application?
- How Do I Run a Sample in an IDE?
- How Do I Create a JavaFX Application?
- How Do I Deploy a JavaFX Application?
- Resources

See the JavaFX Architecture document to learn about the JavaFX platform architecture and to get a brief description of the JavaFX APIs for media streaming, web rendering, and user interface styling.

JavaFX Applications

Written as a Java API, JavaFX application code can reference APIs from any Java library. For example, JavaFX applications can use Java API libraries to access native system capabilities and connect to server-based middleware applications.

The look and feel of JavaFX applications can be customized. Cascading Style Sheets (CSS) separate appearance and style from implementation so that developers can concentrate on coding. Graphic designers can easily customize the appearance and style of the application through the CSS. If you have a web design background, or if you would like to separate the user interface (UI) and the back-end logic, then you can develop the presentation aspects of the UI in the FXML scripting language and use Java code for the application logic. If you prefer to design UIs without writing code, then use JavaFX Scene Builder. As you design the UI, Scene Builder creates FXML markup that can be ported to an Integrated Development Environment (IDE) so that developers can add the business logic.
Availability

JavaFX 2.2 and later releases are fully integrated with the Java SE 7 Runtime Environment (JRE) and the Java Development Kit (JDK). Because the JDK is available for all major desktop platforms (Windows, Mac OS X, and Linux), JavaFX applications compiled to JDK 7 and later also run on all the major desktop platforms. The cross-platform compatibility enables a consistent runtime experience for JavaFX applications developers and users. Oracle ensures synchronized releases and updates on all platforms and offers an extensive support program for companies that run mission-critical applications.

On the JDK download page, you can get a zip file of JavaFX sample applications. The sample applications provide many code samples and snippets that show by example how to write JavaFX applications. See “How Do I Run a Sample Application?” for more information.

Key Features

JavaFX 2.2 and later releases have the following features:

- **Java APIs.** JavaFX is a Java library that consists of classes and interfaces that are written in native Java code. The APIs are designed to be a friendly alternative to Java Virtual Machine (Java VM) languages, such as JRuby and Scala.

- **FXML and Scene Builder.** FXML is an XML-based declarative markup language for constructing a JavaFX application user interface. A designer can code in FXML or use JavaFX Scene Builder to interactively design the graphical user interface (GUI). Scene Builder generates FXML markup that can be ported to an IDE where a developer can add the business logic.

- **WebView.** A web component that uses WebKitHTML technology to make it possible to embed web pages within a JavaFX application. JavaScript running in WebView can call Java APIs, and Java APIs can call JavaScript running in WebView.

- **Swing interoperability.** Existing Swing applications can be updated with new JavaFX features, such as rich graphics media playback and embedded Web content.

- **Built-in UI controls and CSS.** JavaFX provides all the major UI controls required to develop a full-featured application. Components can be skinned with standard Web technologies such as CSS.

- **Canvas API.** The Canvas API enables drawing directly within an area of the JavaFX scene that consists of one graphical element (node).

- **Multitouch Support.** JavaFX provides support for multitouch operations, based on the capabilities of the underlying platform.

- **Hardware-accelerated graphics pipeline.** JavaFX graphics are based on the graphics rendering pipeline (Prism). JavaFX offers smooth graphics that render quickly through Prism when it is used with a supported graphics card or graphics processing unit (GPU). If a system does not feature one of the recommended GPUs supported by JavaFX, then Prism defaults to the Java 2D software stack.

- **High-performance media engine.** The media pipeline supports the playback of web multimedia content. It provides a stable, low-latency media framework that is based on the GStreamer multimedia framework.

- **Self-contained application deployment model.** Self-contained application packages have all of the application resources and a private copy of the Java and
JavaFX runtimes. They are distributed as native installable packages and provide the same installation and launch experience as native applications for that operating system. See the Deploying JavaFX Applications document.

What Can I Build with JavaFX?

With JavaFX, you can build many types of applications. Typically, they are network-aware applications that are deployed across multiple platforms and display information in a high-performance modern user interface that features audio, video, graphics, and animation.

Table 1–1 shows images of a few of the sample JavaFX applications that are included with the JavaFX 2.2.n release.

Table 1–1  Sample JavaFX Applications

<table>
<thead>
<tr>
<th>Sample Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="JavaFX Ensemble" /></td>
<td>Ensemble provides a gallery of samples that demonstrate various JavaFX features, such as animation, charts, and controls. You can view the running sample, read a description, copy the source code, and follow links to the API documentation.</td>
</tr>
<tr>
<td><img src="image" alt="Sales Dashboard (DataApp)" /></td>
<td>DataApp is a client-server application for a fictional global automobile company called Henley Car Sales. Automobile sales are simulated on an EJB server using JavaDB, and the data is available via Derby and a RESTful web service. The client demonstrates a variety of data presentations by using a mix of FXML and JavaFX. <strong>Note:</strong> The DataApp sample has multiple NetBeans projects and cannot be run without some additional setup. The DataAppReadme.html file and the NetBeans project file are in the src\DataApp directory.</td>
</tr>
<tr>
<td><img src="image" alt="SwingInterop Sample" /></td>
<td>This Swing application shows how Swing and JavaFX can be combined. It uses JavaFX components to implement a chart and a simple browser. A JTable Swing component is used for the table.</td>
</tr>
</tbody>
</table>

How Do I Run a Sample Application?

The steps in this section explain how to download and run the sample applications that are available as a separate download with the Java Platform (JDK 7).

**Note:** Before you can run a sample JavaFX application, you need to have the JavaFX runtime libraries on your machine. Before you proceed with these steps, either install the latest version of the JDK 7 or the latest version of the JRE.
To download and run the sample applications:
1. Go to the Java SE Downloads page at
2. Scroll down to locate the JDK 7 and JavaFX Demos and Samples section.
3. Click the Demos and Samples Download button to go to the downloads page.
4. On the Java SE Development Kit 7 Downloads page, scroll down to the JavaFX
   Demos and Samples Downloads section.
5. Download the zip file for the correct operating system and extract the files.
   The javafx-samples-2.2.x directory is created and contains the files for the available
   samples. The NetBeans projects for the samples are in the
   javafx-samples-2.2.x\src directory.
6. Double-click one of the executable files for a sample.
   For example, to run the Ensemble sample application, use any of the following
   methods:
   - To run Ensemble as a standalone application, double-click the Ensemble.jar
     file.
   - To run Ensemble in the browser, double-click the Ensemble.html file.
   - To run Ensemble as a Web Start application, either double-click the
     Ensemble.jnlp file or double-click the Ensemble.html file. JavaScript code
     engages the Java Deployment Toolkit to ensure the user environment has the
     required Java and JavaFX runtimes.

How Do I Run a Sample in an IDE?
You can use several Java development IDEs to develop JavaFX applications. The
following steps explain how to view and run the source code in the NetBeans IDE.

To view and run the sample source code in NetBeans IDE:
1. Download the samples as described above and extract the files.
2. From a NetBeans 7.2 or later IDE, load the project for the sample you want to view.
   a. From the File menu, select Open Project.
   b. In the Open Project dialog box, navigate to the directory that lists the samples.
      The navigation path looks something like this:
      
      ..\javafx_samples-2_2_x-<platform>\javafx-samples-2.2.x\src
   c. Select the sample you want to view.
   d. Click the Open Project button.
3. In the Projects window, right click the project you just opened and select Run.
   Notice the Output window is updated and the sample project is run and deployed.

How Do I Create a JavaFX Application?
Because JavaFX applications are written in the Java language, you can use your
favorite editor or any integrated development environment (IDE) that supports the
Java language (such as NetBeans, Eclipse, or IntelliJ IDEA) to create JavaFX
applications.
To create JavaFX applications:

1. Go to the Java SE Downloads page at http://www.oracle.com/technetwork/java/javase/downloads/index.html to download the Oracle® JDK 7 with JavaFX 2.2.n support. Go to the JavaFX Release Documents page for the system requirements and installation instructions for the JavaFX SDK version that you plan to use.

2. Use the Getting Started with JavaFX tutorial to create simple applications that demonstrates how to work with layouts, style sheets, and visual effects.

3. Use JavaFX Scene Builder to design the UI for your JavaFX application without coding. You can drag and drop UI components to a work area, modify their properties, apply style sheets, and integrate the resulting code with their application logic.
   b. Follow the Getting Started with JavaFX Scene Builder the tutorial to learn more.

How Do I Deploy a JavaFX Application?

JavaFX applications can be deployed in any of the following execution modes:

- **Run as a standalone program.** The application package is available on a local drive. Launch it by using a Java launcher, such as java -jar MyApp.jar, or by double-clicking the application JAR file.

- **Launch from a remote server with Web Start.** Click a link in a web page to start the application from a remote web server. Once you have downloaded it, you can also start a Web Start application from a desktop shortcut.

- **Embed in a web page.** JavaFX content is embedded in the web page and hosted on a remote web server.

- **Launch as a self-contained application.** The application is installed on the local drive and runs as a standalone program using a private copy of Java and JavaFX runtimes. You can launch the application in the same way as other native applications for that operating system, for example by using a desktop shortcut or menu entry.

To deploy a JavaFX application:

1. Decide which execution mode you want to use to deploy the application.

2. Read the Deploying JavaFX Application document for detailed instructions.

Resources

Use the following resources to learn more about the JavaFX technology.


- Read the JavaFX Architecture document.

- Browse JavaFX tutorials and articles for developers.

- Review JavaFX 2 Features, the JavaFX FAQ, and the JavaFX Roadmap.