

JavaFX: Getting Started with JavaFX

1 JavaFX Overview

This chapter provides an overview of the types of applications you can build using JavaFX APIs, where to download the JavaFX libraries, and a high level information about the key JavaFX features being delivered.

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?](#)
- [Resources](#)

See the [Understanding the JavaFX Architecture](#) chapter 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

Since the JavaFX library is 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

The JavaFX APIs are available as a fully integrated feature of the Java SE 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. Support for ARM platforms has also been made available with JavaFX 8. JDK for ARM includes the base, graphics and controls components of JavaFX.

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

The following features are included in JavaFX 8 and later releases. Items that were introduced in JavaFX 8 release are indicated accordingly:

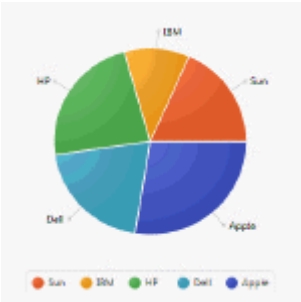
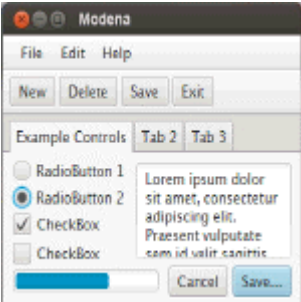

- **Java APIs.** JavaFX is a Java library that consists of classes and interfaces that are written in Java code. The APIs are designed to be a friendly alternative to Java Virtual Machine (Java VM) languages, such as JRuby and Scala.
- **FXML and Scene Builder.** FXML is an XML-based declarative markup 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. Support for additional HTML5 features, including Web Sockets, Web Workers, and Web Fonts, and printing capabilities have been added in JavaFX 8. See [Adding HTML Content to JavaFX Applications](#).
- **Swing interoperability.** Existing Swing applications can be updated with JavaFX features, such as rich graphics media playback and embedded Web content. The `SwingNode` class, which enables you to embed Swing content into JavaFX applications, has been added in JavaFX 8. See the [SwingNode API javadoc](#) and [Embedding Swing Content in JavaFX Applications](#) for more information.
- **Built-in UI controls and CSS.** JavaFX provides all the major UI controls that are required to develop a full-featured application. Components can be skinned with standard Web technologies such as CSS. The `DatePicker` and `TreeTableView` UI controls are now available with the JavaFX 8 release. See [Using JavaFX UI Controls](#) for more information. Also, the `CSS Styleable*` classes have become public API, allowing objects to be styled by CSS.
- **Modena theme.** The Modena theme replaces the Caspian theme as the default for JavaFX 8 applications. The Caspian theme is still available for your use by adding the `setUserAgentStylesheet(STYLESHEET_CASPIAN)` line in your `Application start()` method. For more information, see the [Modena blog](#) at [fxexperience.com](#)
- **3D Graphics Features.** The new API classes for `Shape3D` (`Box`, `Cylinder`, `MeshView`, and `Sphere` subclasses), `SubScene`, `Material`, `PickResult`, `LightBase` (`AmbientLight` and `PointLight` subclasses), and `SceneAntialiasing` have been added to the 3D Graphics library in JavaFX 8. The `Camera` API class has also been updated in this release. For more information, see the [Getting Started with JavaFX 3D Graphics](#) document and the corresponding [API javadoc](#) for `javafx.scene.shape.Shape3D`, `javafx.scene.SubScene`, `javafx.scene.paint.Material`, `javafx.scene.input.PickResult`, and `javafx.scene.SceneAntialiasing`.
- **Canvas API.** The Canvas API enables drawing directly within an area of the JavaFX scene that consists of one graphical element (node).
- **Printing API.** The `javafx.print` package has been added in Java SE 8 release and provides the public classes for the [JavaFX Printing API](#).
- **Rich Text Support.** JavaFX 8 brings enhanced text support to JavaFX, including bi-directional text and complex text scripts, such as Thai and Hindu in controls, and multi-line, multi-style text in text nodes.
- **Multitouch Support.** JavaFX provides support for multitouch operations, based on the capabilities of the underlying platform.
- **Hi-DPI support.** JavaFX 8 now supports Hi-DPI displays.
- **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 software rendering 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.

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 8.n release.

Table 1-1 Sample JavaFX Applications

| Sample Application | Description |
|---|--|
|  | <p>JavaFX Ensemble8</p> <p>Ensemble8 is a gallery of sample applications that demonstrate a large variety of JavaFX features, including animation, charts, and controls. You can view and interact with each running sample on ALL platforms, and read its descriptions. On the desktop platforms, you can copy each sample's source code, adjust the properties of the sample components used in several samples, and follow links to the relevant API documentation when you're connected to the Internet. Ensemble8 also runs with JavaFX for ARM.</p> |
|  | <p>Modena</p> <p>Modena is a sample application that demonstrates the look and feel of UI components using the Modena theme. It gives you the option to contrast Modena and Caspian themes, and explore various aspects of these themes.</p> |
|  | <p>3D Viewer</p> <p>3DViewer is a sample application that allows you to navigate and examine a 3D scene with a mouse or a trackpad. 3DViewer has importers for a subset of the features in OBJ and Maya files. The ability to import animation is also provided for Maya files. (Note that in the case of Maya files, construction history should be deleted on all the objects when saving as a Maya file.)</p> <p>3DViewer also has the ability to export the contents of the scene as Java or FXML files.</p> |

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 8).

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 8](#) or the latest version of the [JRE](#).

To download and run the sample applications:

1. Go to the Java SE Downloads page at <http://www.oracle.com/technetwork/java/javase/downloads/>.

2. Scroll down to locate the JDK 8 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 8 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-8.x` directory is created and contains the files for the available samples. The NetBeans projects for the samples are in the `javafx-samples-8.x\src` directory.

6. Double-click the executable file for a sample.

For example, to run the Ensemble8 pre-built sample application, double-click the `Ensemble8.jar` file.

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 7.4 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-8.x-<platform>\javafx-samples-8.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/> to download the Oracle® JDK 8 with JavaFX 8.n support. Links to the certified system configurations and release notes are also available on that page..
2. Use [Getting Started with JavaFX Sample Applications](#) 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.
 - a. Download the JavaFX Scene Builder from the JavaFX Downloads page at <http://www.oracle.com/technetwork/java/javase/downloads/>.
 - b. Follow the Getting Started with JavaFX Scene Builder tutorial to learn more.

Resources

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

- Download the latest JDK 8 release and the JavaFX samples from the Java SE Downloads page at:
<http://www.oracle.com/technetwork/java/javase/downloads/>.
- Read [Understanding the JavaFX Architecture](#).
- Browse [JavaFX tutorials and articles](#) for developers.