

THE EXPERT'S VOICE® IN JAVA

JavaFX 2.0

Introduction by Example

*DEVELOP HARDWARE-ACCELERATED,
RICH-CLIENT APPLICATIONS USING JAVA*

Carl Dea

Apress®

JavaFX 2.0: Introduction by Example



Carl Dea

Apress®

JavaFX 2.0: Introduction by Example

Copyright © 2011 by Carl Dea

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN 978-1-4302-4257-4

ISBN 978-1-4302-4258-1 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Jonathan Gennick

Technical Reviewer: David Coffin

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Annie Beck

Copy Editor: Nancy Sixsmith

Compositor: Bytheway Publishing Services

Indexer: BIM Indexing & Proofreading Services

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

Contents at a Glance

| | |
|--|-------------|
| ■ About the Author | x |
| ■ About the Technical Reviewer | xi |
| ■ Acknowledgments | xii |
| ■ Introduction | xiii |
| ■ Chapter 1: JavaFX Fundamentals | 1 |
| ■ Chapter 2: Graphics with JavaFX | 69 |
| ■ Chapter 3: Media with JavaFX | 111 |
| ■ Chapter 4: JavaFX on the Web | 141 |
| ■ Index | 175 |

Contents

| | |
|---|-------------|
| ■ About the Author | x |
| ■ About the Technical Reviewer | xi |
| ■ Acknowledgments | xii |
| ■ Introduction | xiii |
| ■ Chapter 1: JavaFX Fundamentals | 1 |
| 1-1. Installing Required Software..... | 2 |
| Problem | 2 |
| Solution | 2 |
| How It Works | 4 |
| 1-2. Creating a Simple User Interface | 5 |
| Problem | 5 |
| Solution #1 | 5 |
| Solution #2 | 7 |
| How It Works | 9 |
| Solution #1 | 9 |
| Solution #2 | 9 |
| 1-3: Drawing Text | 11 |
| Problem | 11 |
| Solution | 11 |
| How It Works | 12 |

| | |
|---|-----------|
| 1-4: Changing Text Fonts..... | 13 |
| Problem | 13 |
| Solution | 13 |
| How It Works | 15 |
| 1-5. Creating Shapes | 16 |
| Problem | 16 |
| Solution | 16 |
| How It Works | 19 |
| 1-6. Assigning Colors to Objects..... | 22 |
| Problem | 22 |
| Solution | 22 |
| How It Works | 25 |
| 1-7. Creating Menus | 27 |
| Problem | 27 |
| Solution | 27 |
| How It Works | 29 |
| 1-8. Adding Components to a Layout | 31 |
| Problem | 31 |
| Solution | 31 |
| How It Works | 32 |
| 1-9. Generating Borders | 33 |
| Problem | 33 |
| Solution | 33 |
| How It Works | 35 |
| 1-10. Binding Expressions | 36 |
| Problem | 36 |
| Solution | 36 |
| How It Works | 39 |

| | |
|---|-----------|
| 1-11. Creating and Working with Observable Lists | 40 |
| Problem | 40 |
| Solution | 41 |
| How It Works | 43 |
| 1-12. Generating a Background Process | 44 |
| Problem | 44 |
| Solution | 44 |
| How It Works | 48 |
| 1-13. Associating Keyboard Sequences to Applications | 49 |
| Problem | 49 |
| Solution | 49 |
| How It Works | 51 |
| 1-14. Creating and Working with Tables | 52 |
| Problem | 52 |
| Solution | 52 |
| How It Works | 56 |
| 1-15. Organizing UI with Split Views | 57 |
| Problem | 57 |
| Solution | 57 |
| How It Works | 59 |
| 1-16. Adding Tabs to the UI | 60 |
| Problem | 60 |
| Solution | 60 |
| How It Works | 62 |
| 1-17. Developing a Dialog Box..... | 63 |
| Problem | 63 |
| Solution | 63 |
| How It Works | 67 |

| | |
|--|------------|
| ■ Chapter 2: Graphics with JavaFX | 69 |
| 2-1. Creating Images | 70 |
| Problem | 70 |
| Solution | 70 |
| How It Works | 75 |
| 2-2. Generating an Animation..... | 79 |
| Problem | 79 |
| Solution | 79 |
| How It Works | 83 |
| 2-3. Animating Shapes Along a Path | 87 |
| Problem | 87 |
| Solution | 87 |
| How It Works | 90 |
| 2-4. Manipulating Layout via Grids..... | 92 |
| Problem | 92 |
| Solution | 92 |
| How It Works | 100 |
| 2-5. Enhancing with CSS | 104 |
| Problem | 104 |
| Solution | 104 |
| How It Works | 108 |
| ■ Chapter 3: Media with JavaFX | 111 |
| 3-1. Playing Audio..... | 111 |
| Problem | 111 |
| Solution | 112 |
| How It Works | 118 |

| | |
|--|------------|
| 3-2. Playing Video | 123 |
| Problem | 123 |
| Solution | 123 |
| How It Works | 128 |
| 3-3. Controlling Media Actions and Events | 132 |
| Problem | 132 |
| Solution | 132 |
| How It Works | 134 |
| 3-4. Marking a Position in a Video | 134 |
| Problem | 134 |
| Solution | 135 |
| How It Works | 136 |
| 3-5. Synchronizing Animation and Media | 137 |
| Problem | 137 |
| Solution | 137 |
| How It Works | 138 |
| ■ Chapter 4: JavaFX on the Web | 141 |
| 4-1. Embedding JavaFX Applications in a Web Page | 141 |
| Problem | 141 |
| Solution | 141 |
| How It Works | 147 |
| 4-2. Displaying HTML5 Content | 149 |
| Problem | 149 |
| Solution | 149 |
| How It Works | 152 |
| 4-3. Manipulating HTML5 Content with Java Code | 153 |
| Problem | 153 |
| Solution | 153 |

| | |
|---|------------|
| How It Works | 157 |
| 4-4. Responding to HTML Events..... | 160 |
| Problem | 160 |
| Solution | 160 |
| How It Works | 162 |
| 4-5. Displaying Content from the Database..... | 163 |
| Problem | 163 |
| Solution | 163 |
| How It Works | 172 |
| ■ Index | 175 |

About the Author



■ **Carl P. Dea** is currently a Software Engineer working for BCT-LLC on projects with high performance computing (HPC) architectures. He has been developing software for 15 years with many clients from Fortune 500 companies to nonprofit organizations. He has written software ranging from mission-critical applications to web applications. Carl has been using Java since the very beginning and he also is a huge JavaFX enthusiast dating back when it used to be called F3. His passion for software development started when his middle school science teacher had shown him the TRS-80 computer. Carl's current software development interests are rich client applications, game programming, Arduino, mobile phones and tablet computers. When he's not working, he and his wife love to watch their daughters perform at gymnastics meets. Carl lives on the East Coast in Pasadena (aka "The Dena"), Maryland, USA.

About the Technical Reviewer

■ **David Coffin** is the author of *Expert Oracle* and *Java Security* from Apress. He is an IT analyst working at the Savannah River Site, a large Department of Energy facility. For more than 30 years, David's expertise has been in multiplatform network integration and systems programming. Before coming to the Savannah River Site, he worked for several defense contractors and served as the technical lead for office and network computing at the National Aerospace Plane Joint Program Office at Wright-Patterson Air Force Base in Ohio. As a perpetual student, he has one master's degree and many hours toward another. As a family man, David has raised eight children. He is a triathlete and distance swimmer who competes in the middle of the pack. He is also a classical guitar player, but he's not quitting his day job.

Acknowledgments

I would like to thank my wife, Tracey, my daughters, Caitlin and Gillian, for their loving support and sacrifices. A special thanks to my daughter Caitlin, who helped with illustrations and brainstorming fun examples. A big thanks to Jim Weaver for recommending me to this project and being so encouraging. I would also thank Josh Juneau for his advice and guidance throughout this journey.

Thanks also to David Coffin with his uncanny ability to know my intentions and provide great feedback. I want to thank the wonderful people at Apress for their professionalism. A special thanks to Jonathan Gennick for believing in me and whipping me into shape.

Thanks to Annie Beck for keeping me on track when things got rough.

Thanks to all who follow me on Twitter especially the ones related to Java Swing, RIA, and JavaFX. Also, thanks to the authors (Jim Weaver, Weiqi Gao, Stephen Chin, and Dean Iverson) of the Pro JavaFX book for allowing me to tech review chapters. Another thanks to Stephen Chin and Keith Combs for heading up the awesome JavaFX User Group. Most importantly all the past JavaFX 1.x book authors that helped inspire me. Thanks goes out to my employer BCT-LLC and their customers who make a lot of things possible. Lastly, I want to give a big kudos and acknowledgment to the people at Oracle who helped me (directly or indirectly) as JavaFX 2.0 was being released: Jonathan Giles, Jasper Potts, Michael Heinrichs, Richard Bair, Amy Fowler, David DeHaven, Nicolas Lorain, Kevin Rushforth, Sheila Cepero, Gail Chappell, Cindy Castillo, Scott Hommel, Joni Gordon, Alexander Kouznetsov, Irina Fedortsova, Dmitry Kostovarov, Alla Redko, Igor Nekrestyanov, Nancy Hildebrandt, and all the Java, JavaFX, and NetBeans teams involved. Whether, then, you eat or drink or whatever you do, do all to the glory of God (1 Corinthians 10:31).

Introduction

JavaFX 2.0 is Java's next generation graphical user interface (GUI) toolkit for developers to rapidly build rich cross-platform applications. Built from the ground up, JavaFX takes advantage of modern GPUs through hardware-accelerated graphics while providing well-designed programming interfaces enabling developers to combine graphics, animation, and UI controls. The new JavaFX 2.0 is a pure Java language application programming interface (API).

The key architectural strategies provided by JavaFX 2.0 API are the reuse of existing Java libraries and the bridging of communication between other languages that run on the JVM (Visage, Jython, Groovy, JRuby, and Scala).

Nandini Ramani of Oracle plainly states the intended direction of JavaFX the platform in the following excerpt from the screencast, *Introducing JavaFX 2.0*:

“The industry is moving toward multi-core/multi-threading [type] platforms with GPUs. JavaFX 2.0 leverages these attributes to improve execution efficiency and UI design flexibility. Our initial goal is to give architects and developers of enterprise applications a set of tools and APIs to help them build better data driven business applications.”

—Nandini Ramani
Oracle Corp.
VP of Development, Java Client Platform

Some History

In 2005, Sun Microsystems acquired the company SeeBeyond, at which a certain software engineer by the name of Chris Oliver created a graphics-rich scripting language known as F3 (Form Follows Function). F3 was later unveiled by Sun Microsystems at the 2007 JavaOne conference as JavaFX.

On April 20, 2009 Oracle Corporation announced the acquisition of Sun Microsystems, making Oracle the new steward of JavaFX. At JavaOne 2010, Oracle announced the JavaFX roadmap. As part of the road map, Oracle revealed its plans to phase out the JavaFX script language and re-create JavaFX for the Java language and platform.

As promised based on the 2010 roadmap, JavaFX 2.0 SDK was released at JavaOne October 3, 2011. Oracle also announced its commitment to take steps to release JavaFX as an open-source product, thus allowing the community to help move the platform forward. Open-sourcing JavaFX will increase its adoption, enable a quicker turnaround time on bug fixes, and generate new enhancements.

Table 0-1 shows the overall history of the major JavaFX releases.

Table 0-1. Historical Timeline of Major JavaFX Releases

| Release Date | Version | Platform | Description |
|-------------------|---------|--------------------------------|---|
| December 4, 2008 | 1.0 | Windows and MacOS | JavaFX Script language, Production Suite, Media Playback |
| February 12, 2009 | 1.1 | Windows and MacOS | New mobile development |
| June 2, 2009 | 1.2 | Windows, MacOS, Linux, Solaris | Skinnable UI controls, Charting API, and performance improvements |
| April 22, 2010 | 1.3 | Windows, MacOS, Linux, Solaris | JavaFX Composer, TV Emulator, Mobile Emulator |
| October 3, 2011 | 2.0 | Windows, MacOS | Rewritten for the Java Language |

Approach in This Book

The title of the book says it all: *JavaFX 2.0 Introduction by Example*. In this book, you will be learning the new JavaFX 2.0 capabilities by following practical recipe examples. These recipes will, in turn, provide you with the knowledge needed to create your own rich client applications. In the same tone with Java’s mantra “Write once, run anywhere,” JavaFX also preserves this same sentiment. Because JavaFX 2.0 is written entirely in Java the language, you will feel right at home.

Most of the recipes can be compiled and run under Java 6. However, some recipes will take advantage of Java 7’s language enhancements, so Java 7 will be required. While working through this book with JavaFX 2.0 and Java 7, you will realize that the new APIs and language enhancements will help you to become a more productive developer. Having said this, I encourage you to explore all of Java 7’s new capabilities. To delve deeper into the new capabilities of Java 7, I recommend the book, *Java 7 Recipes*. On an added note, the recipes in this book can also be found in *Java 7 Recipes*.

This book covers JavaFX 2.0’s fundamentals, graphics and animations, audio and video, and the Web. The fundamentals include how to install prerequisite software (JavaFX 2.0, NetBeans 7.1) and create simple user interfaces. You will also learn the basics of the scene graph, text nodes and font styles, shapes, colors, layouts, menus, UI controls, simple styling (CSS styling), binding expressions, background processes, keyboard shortcuts, and dialog boxes. Next, in graphics and animations you will encounter image handling, drag-and-drop operations, animation APIs, and UI theming (Look ‘n’ Feel). After graphics and animations, you will learn audio and video. This section will include creating an MP3 player, using a video player, responding to media events, handling media marker events, and synchronizing an animation with media events. Finally, you will be using JavaFX 2.0 to interoperate with web technologies such as HTML5, JavaScript, and XML. In this section, you will be learning how to embed JavaFX into a web page, rendering and dynamically manipulating HTML5 content, creating a weather application to respond to HTML events, and creating an RSS feed application using an embedded database (Derby).

Who This Book Is For

If you are a Java developer who desires to take your client-side applications to the next level, you will find this book your guide to help you begin creating usable and aesthetically pleasing user interfaces. If you want a particular platform that is not listed in the preceding table, don't be too concerned because by the time you read this, JavaFX 2.0 should be available on your favorite OS.

How This Book Is Structured

This book is arranged in a natural progression that moves forward from beginner- to intermediate-level concepts. For the Java developer, none of the concepts mentioned in this book should be extremely difficult to figure out. This book's recipes are presented in a problem-solution format. After a brief description of a practical and real-world problem, a step-by-step solution will explain which techniques will be best suited to solve the problem. Each recipe can be easily adapted to meet your own needs when developing a game, media player, or your usual enterprise application. The more experienced Java UI developer you are, the more freedom you have to jump around to different chapters and recipes throughout the book. However, any Java developer can naturally progress through the book and learn the skills needed to enhance everyday GUI applications.

Downloading the Code

Source code is available for the examples in this book. You can download that code from this book's catalog page on the Apress web site. The URL is <http://www.apress.com/9781430242574>. The code will be in a .zip file that is organized by chapter.

References

Following are some online resources that will prove helpful as you begin your journey:

Introducing JavaFX 2.0, by Nandini Ramani:

<http://medianetwork.oracle.com/video/player/1191127359001>

Chris Oliver's weblog: <http://blogs.oracle.com/chrisoliver/entry/f3>

JavaFX Roadmap: <http://javafx.com/roadmap/>

OpenJDK Discussion About JavaFX, by Richard Bair: <http://fxexperience.com/2011/10/openjdk-discussion-about-javafx/>

JavaFX on Wikipedia: <http://en.wikipedia.org/wiki/JavaFX>

JavaFX Fundamentals

The JavaFX 2.0 API is Java's next generation GUI toolkit for developers to build rich cross-platform applications. JavaFX 2.0 is based on a scene graph paradigm (retained mode) as opposed to the traditional immediate mode style rendering. JavaFX's scene graph is a tree-like data structure that maintains vector-based graphic nodes. The goal of JavaFX is to be used across many types of devices such as mobile devices, smartphones, TVs, tablet computers, and desktops.

Before the creation of JavaFX, the development of rich Internet applications (RIAs) involved the gathering of many separate libraries and APIs to achieve highly functional applications. These separate libraries include Media, UI controls, Web, 3D, and 2D APIs. Because integrating these APIs together can be rather difficult, the talented engineers at Sun Microsystems (now Oracle) created a new set of JavaFX libraries that roll up all the same capabilities under one roof. JavaFX is the Swiss Army Knife of GUIs (see Figure 1-1). JavaFX 2.0 is a pure Java (language) API that allows developers to leverage existing Java libraries and tools.

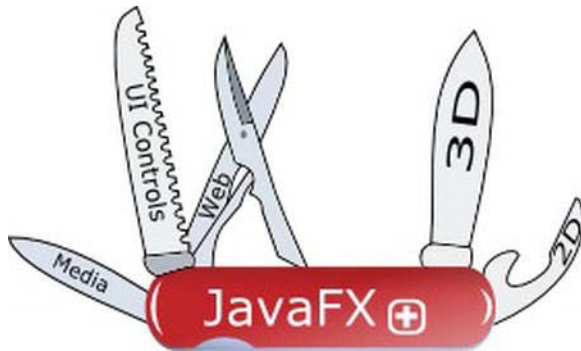


Figure 1-1. JavaFX

Depending on who you talk to, you will likely encounter different definitions of “user experience” (or in the UI world, UX). But one fact still remains; the users will always demand better content and increased usability from GUI applications. In light of this fact, developers and designers often work together to craft applications to fulfill this demand. JavaFX provides a toolkit that helps both the developer and designer (in some cases, they happen to be the same person) to create functional yet esthetically pleasing applications. Another thing to acknowledge is that if you are developing a game, media player, or the usual enterprise application, JavaFX will not only assist in developing richer UIs but you'll also find that the APIs are extremely well designed to greatly improve developer productivity (I'm all about the user of the API's perspective).

Although this book doesn't go through an exhaustive study of all of JavaFX 2.0's capabilities, you will find common use cases that can help you build richer applications. Hopefully, these recipes can lead

you in the right direction by providing practical and real-world examples. I also would like to encourage you to explore other resources to gain further insight into JavaFX. I highly recommend the book *Pro JavaFX Platform* (Apress, 2009) and the soon to be released *Pro JavaFX 2.0 Platform* (Apress, 2012), which is an invaluable resource. These books go in depth to help you create professional grade applications.

So without further ado, let's get started, shall we?

1-1. Installing Required Software

Problem

You want to start developing JavaFX applications, but you don't know what software is required to be installed.

Solution

You'll need to install the following software in order to get started with JavaFX:

- Java 7 JDK or greater
- JavaFX 2.0 SDK
- NetBeans IDE 7.1 or greater

■ **Note** As of this writing, things are subject to change. To see additional requirements, refer to http://download.oracle.com/javafx/2.0/system_requirements/jfxpub-system_requirements.htm.

As of this writing, things are subject to change. By the time you read this; you will likely find JavaFX able to run on your favorite OS. For this recipe, I assume that Java 7 is already installed so I won't detail those installation steps. Following are steps to install all other required software components:

1. Download JavaFX 2.0 and NetBeans IDE 7.1.x from the following locations:
 - JavaFX 2.0 SDK: <http://www.oracle.com/technetwork/java/javafx/downloads/index.html>
 - NetBeans 7.1 beta SDK: <http://netbeans.org>
2. Install JavaFX 2.0 SDK. The screen in Figure 1-2 will appear once you've launched the JavaFX SDK Setup executable.

Once you have launched the JavaFX SDK setup executable you will see the start of the wizard in Figure 1-2.

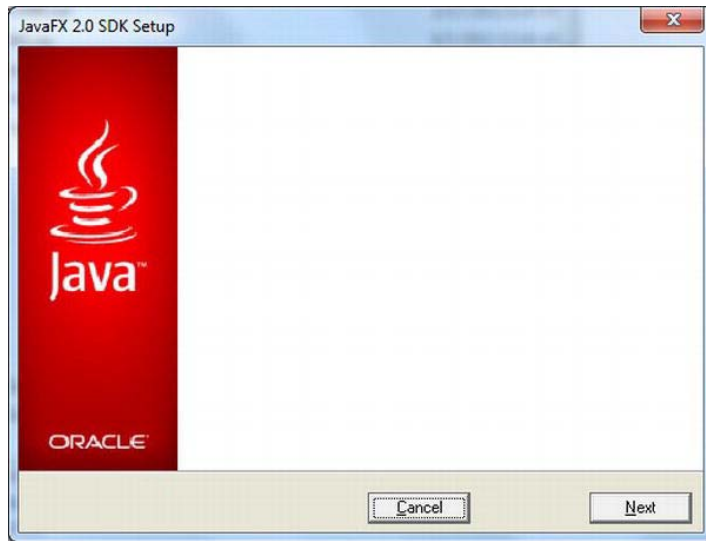


Figure 1-2. JavaFX 2.0 SDK Setup Wizard

3. Next, you can specify the home directory of the JavaFX SDK by clicking the Browse button. Figure 1-3 shows the default location for the JavaFX SDK's home directory. You might want to jot this location down in order to configure your CLASSPATH in Step 6.

Figure 1-3 displays Setup Options, which allow you to specify the JavaFX 2.0 SDK's home directory.

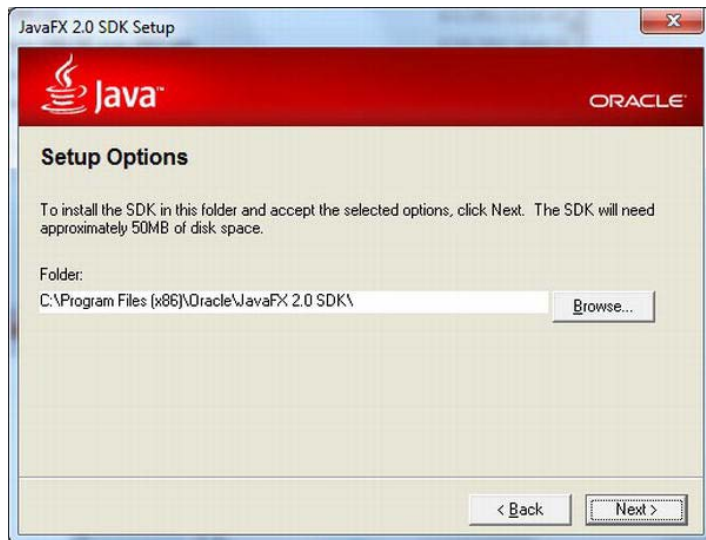


Figure 1-3. JavaFX SDK home directory

4. After you click Next, the components will install and the screen shown in Figure 1-4 will appear.

Figure 1-4 displays the progress indicator installing the last components before completing.

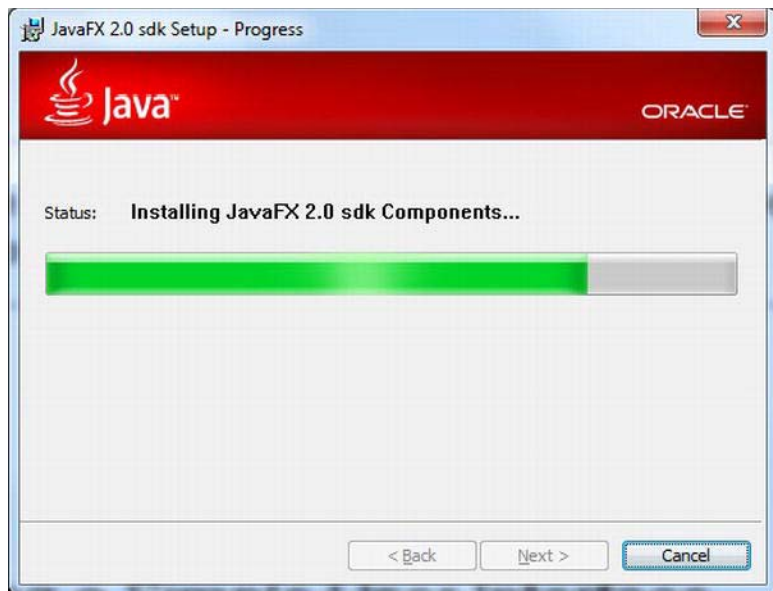


Figure 1-4. Completing the install

5. Install the NetBeans IDE, which includes the JavaFX 2.0 plug-in.

When installing, you will follow the default wizard screens. For additional instructions, you may refer to <http://netbeans.org/community/releases/71/install.html>.

6. Configuring your environment variable CLASSPATH to include the JavaFX runtime library. The name and location of the runtime library is at `<JavaFX SDK Home directory>\rt\lib\jfxrt.jar`. (Linux uses the forward slash: /).

How It Works

This recipe shows how to install Java FX 2.0 and the NetBeans IDE onto the Windows platform. You may need to modify your steps slightly when installing JavaFX 2.0 on other operating systems as they become available. Although the steps described here are for NetBeans, you can also develop using other IDEs such as Eclipse, IntelliJ, or vi. While most of the example recipes were created using the NetBeans IDE, you can also compile and run JavaFX applications using the command-line prompt.

To compile and run JavaFX applications using the command-line prompt you will need to configure your CLASSPATH. After you have followed the wizards to install the prerequisite software you will need to set your environment's CLASSPATH variable to include the JavaFX runtime library `<JavaFX SDK Home directory>\rt\lib\jfxrt.jar` (Step 6). Setting this library will later assist in compiling and running JavaFX-based applications on the command-line. The following code configures your CLASSPATH environment variable based on your platform: