Before You Start

IF you haven't written and run programs using the Java platform, then please close this book and go to the "Getting Started" trail of *The Java Tutorial*. That trail will introduce the Java programming language and platform and lead you through writing and running a basic application and applet. The trail is included on this book's CD at: `JavaTutorial/ getStarted/index.html`.

This book is divided into two distinct parts. The first six chapters describe general Swing concepts and introduce you to progressively more difficult code examples. The last four chapters are composed of how-to reference sections.

Our goal was to make it easy to skip around this book, choosing exactly which sections you want to read. That's why the how-to sections in the reference chapters are self-contained and tabbed. The Appendix contains several handy troubleshooting guides. We also provide the following navigational aids:

- The contents (page v) lists all of the chapters in this book, and the major sections within them.

- The beginning of each chapter describes what you can expect to learn and has its own table of contents that lists the sections within it.

- Within each chapter, you can expect to find links to related information whenever appropriate.

- The Index (page 763) is a traditional book index.

At the back of this book, The Java Tutorial CD description lists the contents of the CD-ROM. In particular, it includes *The Java Tutorial* in HTML and all of the source code for the examples in this book. The CD also includes most of the resources referenced in this book, such as the API documentation and the online magazine—*The Swing Connection*.

BEFORE YOU START

It's easy to translate the URLs in the book to their locations on the CD-ROM; for example, the URL for `JApplet` API documentation is:

```
http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/JApplet.html
```

It can be found on the CD at:

```
Docs/j2sdk14-api/docs/api/javax/swing/JApplet.html
```

If you need some information that's not in this book or in the API documentation, we recommend that you consult these resources:

- The JavaDesktop community Web site, especially its articles and forums at: `http://javadesktop.org/` and `http://javadesktop.org/forums/`.
- The online version of The Java Tutorial: `http://java.sun.com/docs/books/tutorial/index.html`.

In addition, you can find other resources listed on the following page:

```
http://java.sun.com/docs/books/tutorial/information/resources.html
```

1

Getting Started with Swing

THIS chapter gives you a brief introduction to using the Java™ Foundation Classes (JFC) Swing packages. After telling you about JFC and Swing, it helps you get the necessary software and walks you through how to compile and run a program that uses the Swing packages. Next, it shows you how to run programs using Java Web Start.

The next chapter, Learning Swing by Example (page 11), will build on these first steps to help you create several increasingly more complex examples. For now, let's start with the basics.

About the JFC and Swing

JFC is short for Java Foundation Classes, which encompass a group of features for building graphical user interfaces (GUIs[1]) and adding rich graphics functionality and interactivity to Java applications. JFC was first announced at the 1997 JavaOne[SM] developer conference. It is defined as containing the features shown in Table 1.[2]

Table 1: Features of the Java Foundation Classes

Feature	Description
Swing GUI Components	Includes everything from buttons to split panes to tables. See screenshots of all the components in A Visual Index to Swing Components (page 37) in Chapter 3.
Pluggable Look-and-Feel Support	Gives any program that uses Swing components a choice of look and feel. For example, the same program can use either the Java or the Windows look and feel. Many more look-and-feel packages are available from various sources. As of v1.4.2, the Java platform supports the GTK+ look and feel, which makes hundreds of existing look and feels available to Swing programs.
Accessibility API	Enables assistive technologies, such as screen readers and Braille displays, to get information from the user interface.
Java 2D[TM] API	Enables developers to easily incorporate high-quality 2D graphics, text, and images in applications and applets. Java 2D includes extensive APIs for generating and sending high-quality output to printing devices.
Drag-and-Drop Support	Provides the ability to drag and drop between Java applications and native applications.
Internationalization	Allows developers to build applications that can interact with users worldwide in their own languages and cultural conventions. With the input method framework developers can build applications that accept text in languages that use thousands of different characters, such as Japanese, Chinese, or Korean.

This book concentrates on the Swing components. We help you choose the appropriate components for your GUI, tell you how to use them, and give you the background information you need to use them effectively. We also discuss other JFC features as they apply to Swing components.

---

[1] The acronym for graphical user interface, GUI, is pronounced "gooey."
[2] See the JFC home page for more information on JFC features: `http://java.sun.com/products/jfc/index.html`.

ABOUT THE JFC AND SWING

> **Note:** "Swing" was the code name of the project that developed the new components. Although unofficial, it's frequently used to refer to the new components and related API. "Swing" is immortalized in the package names for the Swing API, which begin with `javax.swing`.

Which Releases Contain the Swing API?

The short answer is that the Swing API has been included in the Java 2 platform, Standard Edition (J2SE™) since its initial release (1.2). A 1.4.2 release of the Java 2 platform is included on the CD that accompanies this book. You can also download the latest release from the Sun Microsystems Web site at: `http://java.sun.com/j2se/`.

This book concentrates on the Swing API in the Java 2 platform, Standard Edition, v1.4.2. Except where noted, the code in this book works, without change, with earlier J2SE releases.[1] We also include notes about a few important changes expected in 1.5.

Which Swing Packages Should I Use?

The Swing API is powerful, flexible—and immense. In release 1.4 of the Java platform, the Swing API has 17 public packages:

```
javax.accessibility      javax.swing.plaf        javax.swing.text.html

javax.swing              javax.swing.plaf.basic  javax.swing.text.parser

javax.swing.border       javax.swing.plaf.metal  javax.swing.text.rtf

javax.swing.colorchooser javax.swing.plaf.multi  javax.swing.tree

javax.swing.event        javax.swing.table       javax.swing.undo

javax.swing.filechooser  javax.swing.text
```

Fortunately, most programs use only a small subset of the API. This book sorts out the API for you, giving you examples of common code and pointing you to methods and classes you're likely to need. Most of the code in this book uses only one or two Swing packages:

- `javax.swing`
- `javax.swing.event` (not always required)

---

[1] We recommend that you use the latest version of the Java 2 platform. However, it is possible to use the Swing components with releases as early as 1.1. We give tips for using earlier releases where necessary.

Compiling and Running Swing Programs

This section explains how to compile and run a Swing application. The compilation instructions work for all Swing programs—applets, as well as applications. The following are the steps you need to follow:

1. Install the latest release of the Java 2 platform, if you haven't already done so.
2. Create a program that uses Swing components.
3. Compile the program.
4. Run the program.

Note: If you're new to writing and compiling programs that use the Java platform, read *The Java Tutorial* trail "Getting Started" first; it's available online and on the CD at: `JavaTutorial/ getStarted/index.html`. That trail also explains common errors that users may encounter.

Install the Latest Release of the Java 2 Platform

You can download the latest release of the J2SE SDK for free from `http://java.sun.com/ j2se`. Version 1.4.2 of the J2SE SDK is included on the CD that accompanies this book.

Create a Program That Uses Swing Components

You can use a simple program we provide, called `HelloWorldSwing`, that brings up the GUI shown in Figure 1. The program is in a single file, `HelloWorldSwing.java`.[1] When you save this file, you must match the spelling and capitalization of its name exactly. If you prefer to type in the file's code yourself, you can find the full code starting on page 12 in Chapter 2.

Figure 1   The `HelloWorldSwing` application.

Compile the Program

Your next step is to compile the program. Here's an example of compiling `HelloWorld-Swing.java`:

---

[1] You can find `HelloWorldSwing.java` here: `JavaTutorial/uiswing/14start/example-1dot4/ HelloWorldSwing.java`.

COMPILING AND RUNNING SWING PROGRAMS

```
javac HelloWorldSwing.java
```

If you can't compile, make sure you're using the compiler in a recent release of the Java platform, such as 1.4.2 included on this book's CD. Once you've updated your SDK, you should be able to use the programs in this book without changes. Another common mistake is installing the Java Runtime Environment (JRE) and not the full Software Development Kit (SDK) needed to compile these programs. Refer to the "Getting Started" trail of *The Java Tutorial* to help you solve any compiling problems you encounter.[1] Another installation troubleshooting guide for the Java 2 platform is online at: `http://servlet.java.sun.com/help/installation/`.

Run the Program

After you compile the program successfully, you can run it. Assuming that your program uses a standard look and feel—such as the Java, Windows, or GTK+ look and feel—you can use the interpreter to run the program without adding anything to your class path. For example:

```
java HelloWorldSwing
```

For programs that use a nonstandard look and feel or any other nonstandard code package, you must make sure that the necessary classes are in the class path. For example:

Solaris:
```
java -classpath.:/home/me/lnfdir/newlnf.jar HelloWorldSwing
```

Microsoft Windows:
```
java -classpath .;C:\java\lnfdir\newlnf.jar HelloWorldSwing
```

Alternatively, you can launch your program from a Web browser using Java Web Start.

Running Programs Using Java Web Start

Java Web Start is a technology that simplifies the distribution of applications. With a click on a Web page link, you can launch full-featured applications without a complicated download and installation process. The first time you launch an application, Java Web Start automatically downloads all necessary files. It then caches them on your computer so the application can be quickly relaunched from a desktop shortcut or from a Web browser. Java Web Start checks for updates each time you run a remote application and downloads any updated files automatically.

---

[1] The "Getting Started" trail is available online and on this book's CD at: `JavaTutorial/getStarted/index.html`.

In 1.4.1 and later releases of the Java platform, Java Web Start is shipped as part of the plat-form. So, if you have J2SE or JRE v1.4.1 or higher, you already have Java Web Start installed. When you install a version of the Java platform with Java Web Start, on most plat-forms you'll see a Java Web Start shortcut on your desktop. (See Figure 2.)

Figure 2   The Java Web Start desktop shortcut.

To test whether your browser can launch an application using Java Web Start, point it to one of the following places and click the "Launch Java™ Web Start" link:

- `JavaTutorial/tutorial/uiswing/14start/HelloJWS.html` (on this book's CD)
- `http://java.sun.com/docs/books/tutorial/uiswing/14start/HelloJWS.html`

You should see a Java Web Start splash screen. (See Figure 3.)

Figure 3   The Java Web Start splash screen.

The application is then downloaded, and you should see what's shown in Figure 4.

You're running an application using Java™ Web Start!

Figure 4   The `HelloJWS`[1] application launched from a Web browser using Java Web Start.

If you're unable to launch this application, you should check the Java Web Start Trouble-shooting (page 733) guide in the Appendix.

Questions and Exercises

Questions

1. Which package is required for all Swing applications?
2. What is Java Web Start and where can you get it?

Exercises

1. Show how you would set the class path when launching an application.
2. Test if your browser is Java Web Start enabled by following the instructions listed in Running Programs Using Java Web Start (page 7).

Answers

You can find answers to these Questions and Exercises online:

```
http://java.sun.com/docs/books/tutorial/uiswing/QandE/answers.html
```

Example Programs

You can find source files for all the examples from this chapter on the CD and online:

```
JavaTutorial/uiswing/14start/example-1dot4/index.html
```

```
http://java.sun.com/docs/books/tutorial/uiswing/14start/
example-1dot4/index.html
```

---

[1] You can find the `HelloJWS.java` source files here: `JavaTutorial/uiswing/14start/example-1dot4/index.html#HelloJWS`.
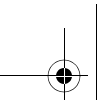
The preceding URLs take you to an index that has links to the files required by each example. You can go directly to the entry for a particular example by adding *#ExampleName* to the URL. Most examples have a "Run" link in the example index which executes the example using Java Web Start technology.

Example	Where Described	Notes
`HelloWorldSwing`	page 6	Demonstrates the basic code in every Swing program.
`HelloJWS`	page 9	Tests whether you can launch a Java Web Start-enabled application.