Trail: Creating a GUI With JFC/Swing

Lesson: Getting Started with Swing

Examples Index

This lesson gives you a brief introduction to using Swing. After telling you about Swing, it walks you through how to compile and run a program that uses the Swing packages.

The following lesson, Learning Swing with the NetBeans IDE will build on these first steps to help you create several progressively more complex examples. For now, let's start with the basics.

About the JFC and Swing

This section gives you an overview of Swing.

Compiling and Running Swing Programs

This section gives you detailed instructions on where to obtain the latest JDK and how to create, compile and run a program that uses Swing components.

If you are interested in using the JavaFX package to create your UI, see the JavaFX Documentation. To deploy, see Java Platform, Standard Edition Deployment Guide.

Previous page: Table of Contents
Next page: About the JFC and Swing

Trail: Creating a GUI With JFC/Swing
Lesson: Getting Started with Swing

About the JFC and Swing

JFC is short for Java Foundation Classes, which encompass a group of features for building graphical user interfaces (GUIs) and adding rich graphics functionality and interactivity to Java applications. It is defined as containing the features shown in the table below.

Feature	Description
Swing GUI Components	Includes everything from buttons to split panes to tables. Many components are capable of sorting, printing, and drag and drop, to name a few of the supported features.
Pluggable Look-and-Feel Support	The look and feel of Swing applications is pluggable, allowing a choice of look and feel. For example, the same program can use either the Java or the Windows look and feel. Additionally, the Java platform supports the GTK+ look and feel, which makes hundreds of existing look and feels available to Swing programs. Many more look-and-feel packages are available from various sources.
Accessibility API	Enables assistive technologies, such as screen readers and Braille displays, to get information from the user interface.
Java 2D 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.
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 trail 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 features as they apply to Swing components.

Which Swing Packages Should I Use?

The Swing API is powerful, flexible — and immense. The Swing API has 18 public packages:

```
javax.accessibility              javax.swing.plaf              javax.swing.text
javax.swing                      javax.swing.plaf.basic        javax.swing.text.html
javax.swing.border               javax.swing.plaf.metal        javax.swing.text.html.parser
javax.swing.colorchooser         javax.swing.plaf.multi        javax.swing.text.rtf
javax.swing.event                javax.swing.plaf.synth        javax.swing.tree
javax.swing.filechooser          javax.swing.table             javax.swing.undo
```

Fortunately, most programs use only a small subset of the API. This trail 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 trail uses only one or two Swing packages:

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

Previous page: Getting Started with Swing
Next page: Compiling and Running Swing Programs

Trail: Creating a GUI With JFC/Swing
Lesson: Getting Started with Swing

Compiling and Running Swing Programs

This section explains how to compile and run a Swing application from the command line. For information on compiling and running a Swing application using NetBeans IDE, see Running Tutorial Examples in NetBeans IDE. The compilation instructions work for all Swing programs — applets, as well as applications. Here are the steps you need to follow:

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

Install the Latest Release of the Java SE Platform

You can download the latest release of the JDK for free from http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Create a Program That Uses Swing Components

You can use a simple program we provide, called HelloWorldSwing, that brings up the GUI shown in the figure below. The program is in a single file, `HelloWorldSwing.java`. When you save this file, you must match the spelling and capitalization of its name exactly.

The `HelloWorldSwing.java` example, like all of our Swing tutorial examples, is created inside a package. If you look at the source code, you see the following line at the beginning of the file:

```
package start;
```

This means you must put the `HelloWorldSwing.java` file inside of a `start` directory. You compile and run the example from the directory above the `start` directory. The tutorial examples from the *Using Swing Components* lesson are inside of a `components` package and the examples from the *Writing Event Listeners* lesson are inside a `events` package, and so on. For more information, you might want to see the `Packages` lesson.

Compile the Program

Your next step is to compile the program. To compile the example, from the directory above the `HelloWorldSwing.java` file:

```
javac start/HelloWorldSwing.java
```

If you prefer, you may compile the example from within the `start` directory:

```
javac HelloWorldSwing.java
```

but you must remember to leave the `start` directory to execute the program.

If you are unable to compile, make sure you are using the compiler in a recent release of the Java platform. You can verify the version of your compiler or Java Runtime Environment (JRE) using these commands

```
javac -version
java -version
```

Once you've updated your JDK, you should be able to use the programs in this trail without changes. Another common mistake is installing the JRE and not the full Java Development Kit (JDK) needed to compile these programs. Refer to the Getting Started trail to help you solve any compiling problems you encounter. Another resource is the Troubleshooting Guide for Java™ SE 6 Desktop Technologies.

Run the Program

After you compile the program successfully, you can run it. From the directory above the `start` directory:

```
java start.HelloWorldSwing
```

Previous page: About the JFC and Swing
Next page: Learning Swing with the NetBeans IDE