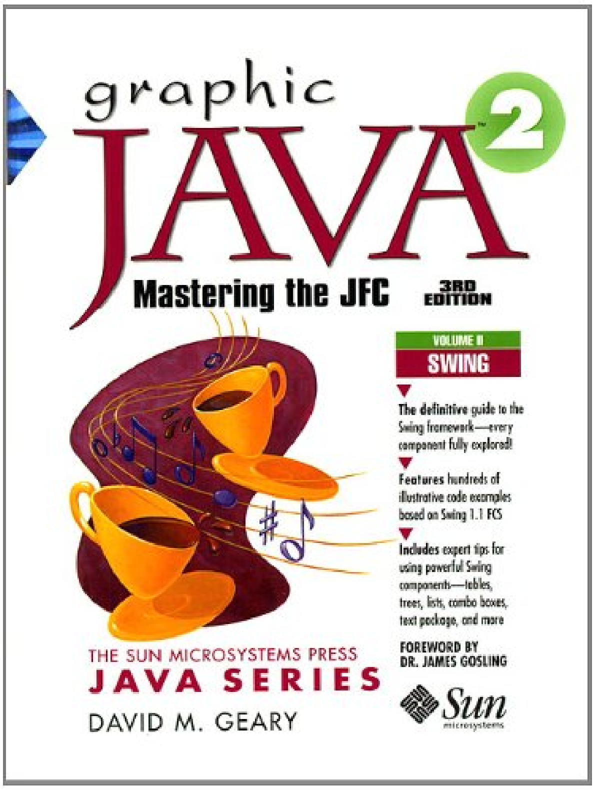
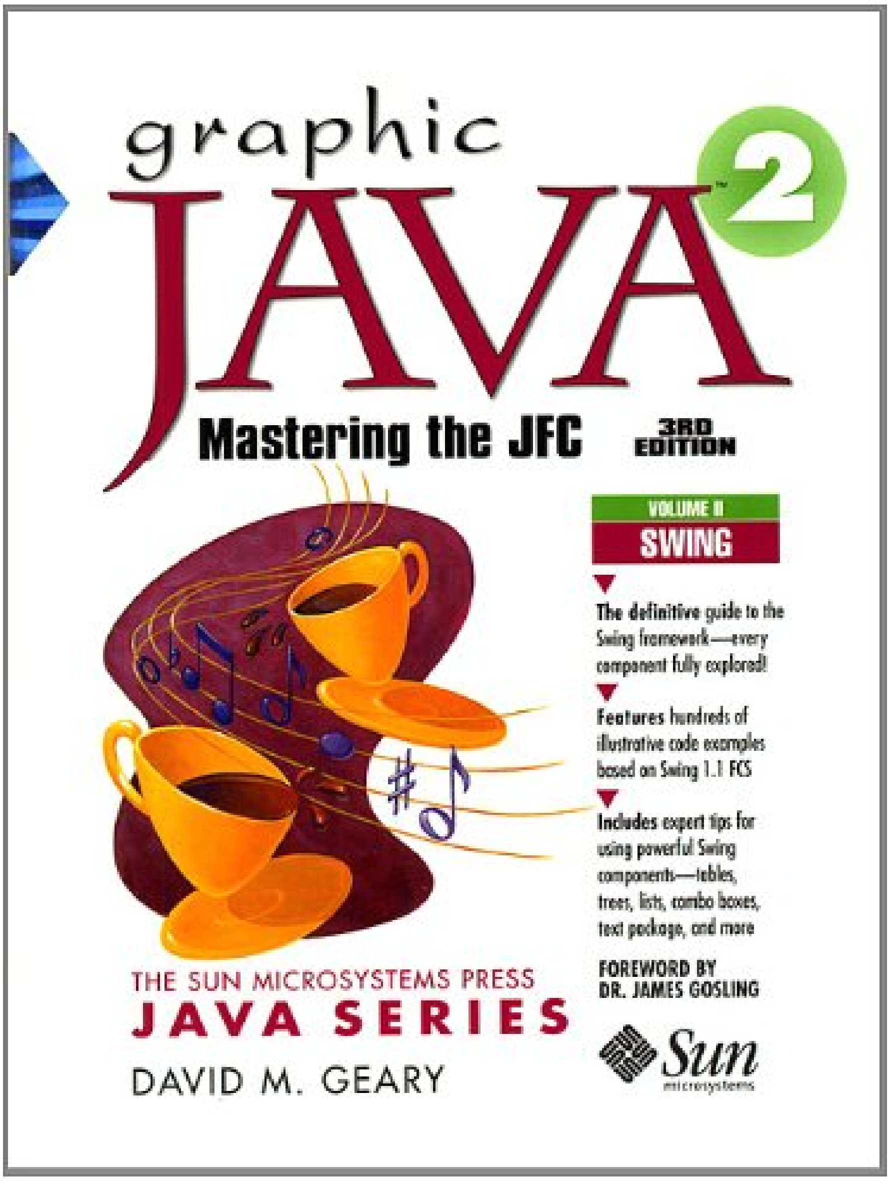
GRAPHIC JAVA 2: MASTERING THE JFC, BY GEARY, 3RD EDITION, VOLUME 2: SWING BY DAVID GEARY


graphic
JAVA 2
Mastering the JFC
3RD EDITION
VOLUME II
SWING
The definitive guide to the Swing framework—every component fully explored!
Features hundreds of illustrative code examples based on Swing 1.1 FCS
Includes expert tips for using powerful Swing components—tables, trees, lists, combo boxes, text package, and more
FOREWORD BY DR. JAMES GOSLING
THE SUN MICROSYSTEMS PRESS
JAVA SERIES
DAVID M. GEARY
Sun microsystems


graphic
JAVA 2
Mastering the JFC
3RD EDITION
VOLUME II
SWING
The definitive guide to the Swing framework—every component fully explored!
Features hundreds of illustrative code examples based on Swing 1.1 FCS
Includes expert tips for using powerful Swing components—tables, trees, lists, combo boxes, text package, and more
FOREWORD BY DR. JAMES GOSLING
THE SUN MICROSYSTEMS PRESS
JAVA SERIES
DAVID M. GEARY
Sun microsystems

Click link bellow and free register to download ebook:
GRAPHIC JAVA 2: MASTERING THE JFC, BY GEARY, 3RD EDITION, VOLUME 2: SWING BY DAVID GEARY

DOWNLOAD FROM OUR ONLINE LIBRARY

GRAPHIC JAVA 2: MASTERING THE JFC, BY GEARY, 3RD EDITION, VOLUME 2: SWING BY DAVID GEARY PDF

As one of the book collections to suggest, this *Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary* has some strong factors for you to review. This book is very suitable with just what you need currently. Besides, you will additionally enjoy this book Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary to review because this is among your referred publications to read. When going to get something brand-new based on encounter, entertainment, and also other lesson, you could utilize this book Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary as the bridge. Beginning to have reading routine can be undergone from various ways as well as from variant sorts of books

Amazon.com Review
If you're developing software that will be used by a large group of people, you need to give it a good-looking front-end--in Java 2, that means you have to use Swing. An excellent resource, Graphic Java 2: Mastering the JFC, Third Edition (Volume 2: Swing) takes on the Swing components one at a time and shows you how to incorporate them into attractive, efficient programs.

In many ways, Graphic Java 2 is a cookbook. You search the table of contents or index for a reference to the kind of problem you want to solve, then examine the author's examples for the solution (or at least some clues to it). This is the book to turn to if you're wondering how to implement the JComboBox.KeySelectionManager interface (which enables users to select items in a combo box) or compare the various ways of making the JTree component into a file browser. Those are just two of hundreds of examples in David Geary's book.

While most examples don't serve any practical purpose by themselves, they do clearly illustrate how a specific aspect of Swing works. It's easy to adapt the details presented here into your own programs. Geary shows consideration for the reader by presenting all his examples as programs that can be compiled and including them on the enclosed CD-ROM. --David Wall

Review
"At over 1500 pages, Graphic Java 2 is a monolith, heavy with details and weighted with valuable information for all Java UI developers. The coverage is meticulous and fluid, instructive in countless ways, unveiling things you won't find anywhere else. Yet it remains accessible to even a novice Swing developer. The net effect is that the price tag seems trivial given the value you can get from this book...

The coverage is not just impressive in its sheer volume; it's also an easy read with clear, concise explanations and useful examples...

All the fundamentals--in eloquent prose and incredible detail--are laid out before the reader in logical chapters that build on each other to unveil the most complete picture I've seen in print. I can hardly wait to read Volume III." -- DevX Book Reviews, July 29, 1999

"Thinking about making that jump from the AWT to Swing? Not sure where to start or how to go about it? Your search stops with David Geary's Graphic Java 2: Mastering the JFC...

Part 2 covers all the Swing components and their properties in detail; each component allocated a chapter. Two of the most complex components--Table and Tree--are covered in great detail and Chapters 19 and 20, respectively. These two chapters alone are worth the price of the book for serious Swing developers...

All in all, Geary has done such a thorough job in Graphic Java 2 that it can truly be considered the Swing bible. Don't let the size of the book intimidate you, it provides a top-level view as well as microscopic details on all Swing components. It will definitely prove to be an asset in the Swing developer's arsenal." -- Dr. Dobbs Journal, September, 1999

From the Inside Flap

Preface

This book has been my full-time passion for more than a year. After Sun closed its office in Colorado Springs in August of 1997, I decided to forego traditional employment to give the Graphic Java series my complete attention. I was determined that the Swing volume would be the definitive guide to Swing, which meant that it had to be the most comprehensive, accurate and insightful Swing book available. I believe that to be the case, but of course, you will be the final judge.

Before I began writing, I spent a considerable amount of effort designing a book that would be beneficial to Swing novices in addition to developers with some Swing experience. As a software developer who has used numerous object oriented GUI frameworks, I realized that the fastest way to climb a framework's learning curve is to study code examples that illustrate specific concepts. As a result, code examples are the foundation upon which this book is built; nearly 300 code examples are discussed in detail--an average of approximately one code example every 5 pages.

However, code examples in and of themselves, cannot suffice as a reference for developers who have advanced along the Swing learning curve. As a result, every Swing component discussion is accompanied by a component summary that includes class diagrams, an examination of the component's properties and events and a class summary that discusses the public and protected methods implemented by the component.

The first part of this book explores fundamental aspects of Swing including:

Swing applets and applications

Swing and multithreading

Swing's Model-View-Controller architecture

Pluggable Look And Feel

The JComponent class

Borders, icons and actions

Miscellaneous features including timers and Swing utilities

The second part of the book examines Swing's components in detail. Numerous code examples are discussed for every Swing component, from labels and buttons to tables, trees and the text package. For example, more than 150 pages containing 25 code examples are dedicated to the table component, and 120 pages and 20 code examples illustrate how to get the most out of the tree component.

Audience

This book assumes that the reader has a good grasp of the Java language including recent additions to the language such as inner classes. This book also assumes a rudimentary understanding of the AWT; specifically, the delegation event model and the Component and Container classes upon which all Swing components are based. See "Graphic Java 2 Mastering the JFC Volume I: AWT" for a thorough investigation of AWT infrastructure and components.

How To Use This Book

Before diving into the Swing components, it is useful to have an understanding of fundamental concepts such as Swing's Model-View-Controller (MVC) design and pluggable look and feel. The former is discussed in "Swing Component Architecture" on page 71 and the latter is covered in the "Pluggable Look and Feel" on page 317. It is also important to understand the services provided by the JComponent class, which is the ultimate superclass of all lightweight Swing components. The JComponent class is examined in "The JComponent Class" on page 123.

Component Summaries

Each Swing component is introduced with numerous code examples that illustrate various component features. Code examples are followed by component summaries, such as the JScrollPane component summary that is partially listed below.

Component summaries begin with a listing of the component's model, UI delegate, renderer and editor, in addition to the events fired by the component. If a component is a replacement for an AWT component, the AWT component is also listed.

JScrollPane

----

javax.swing.plaf.basic.BasicScrollPaneUI

PropertyChangeEvent

java.awt.ScrollPane

The static relationships that a component maintains with other objects are illustrated with a class diagram, such as the class diagram for the JScrollPane class shown above. Class diagrams are introduced in Appendix A, Class Diagrams .

Properties

Property tables, such as the property table for the JScrollPane class shown below, are used to communicate a component's properties. Property tables include property names, the property's data type and whether a property is boolean, bound (a property change event is fired when the property is changed), simple (no events are fired when the property is changed), constrained (changes to the property can be vetoed), or indexed (a parameter, usually an integer , is used to access the property).

Some properties can be specified when a component is instantiated, and a component may provide setter and getter methods for a property. The manner in which a property can be specified is listed in the Access column of a property table.

Property tables also include a column for property default values.

Property tables are followed by a short description of each of the properties listed in the table.

JScrollPane Properties

Property Name	Data Type	Property Type	Access	Default
columnHeader	JViewport	B	SG	null
columnHeaderView	Component	B	S	null
corner				

Component

IB

SG

null

horizontalScrollbar

JScrollBar

B

SG

----

horizontalScrollbarPolicy

int

B

CSG

As needed

rowHeader

JViewport

B

SG

null

rowHeaderView

Component

B

S

null

verticalScrollbar

JScrollBar

B

SG

----

verticalScrollbarPolicy

int

B

CSG

As needed

viewport

JViewport

B

SG

JViewport

viewportBorder

Border

B

SG

null

viewportView

Component

B

CSG

null

columnHeader

-- An instance of JViewport for the column header.

columnHeaderView

-- An instance of Component used as the column header viewport's view.

corner

-- A component that is displayed in one of the

scrollpane's corners. The corner is specified by one of the following strings:

ScrollPaneConstants.UPPER_LEFT_CORNER

ScrollPaneConstants.LOWER_LEFT_CORNER

ScrollPaneConstants.UPPER_RIGHT_CORNER

ScrollPaneConstants.LOWER_RIGHT_CORNER

horizontalScrollbar

-- The horizontal scrollbar used by the

scrollpane. The scrollbar is an instance of JScrollPane.ScrollBar , an extension of JScrollBar that takes into account whether the view contained in the scrollpane implements the Scrollable interface.

The remaining property descriptions for the JScrollPane class are omitted.

Events

Code examples are presented that illustrate event handling for a component. For example, the Tree chapter provides five code examples that illustrate handling of tree mouse, editing, selection, and expansion events.

Class Summaries

Each component summary concludes with a class summary that provides descriptions of the component's constructors and methods, such as the class summary for the JScrollPane class listed below.

JScrollPane

JComponent

ScrollPaneConstants, javax.accessibility.Accessible

public JScrollPane()

public JScrollPane(int vsbPolicy, int hsbPolicy)

public JScrollPane(Component view)

public JScrollPane(Component view, int vsbPolicy, int hsbPolicy)

JScrollPane provides four constructors. The integer values passed to JScrollPane

constructors represent the vertical and horizontal scrollbar display policies, in that order. The component passed to the constructors is used as the viewport's view, meaning it is the component that is scrolled by the

scrollpane.

The no-argument constructor constructs a scrollpane with a null component for the viewport's view, and scrollbar display policies that display both the horizontal and vertical scrollbars as needed.

Creation Methods

public JScrollBar createHorizontalScrollBar()

public JScrollBar createVerticalScrollBar()

protected JViewport createViewport()

Like most Swing components, JScrollPane

provides create... methods that create its subcomponents. Unlike most Swing components however, JScrollPane implements the methods that create its scrollbars as public instead of protected because the methods are invoked from the scrollpane's UI delegate.

The createHorizontalScrollBar and createVerticalScrollBar methods both return instances of JScrollPane.ScrollBar , which is an extension of JScrollBar that takes into account whether the scrollpane's view implements the Scrollable

interface. See Interface Summary 13-2 on page 779 for more information on the Scrollable interface.

The createViewport method returns an instance of JViewport that is used as the

scrollpane's default viewport.

As with all Swing components that implement create... methods, the methods can be overridden in extension classes to replace the default sub-components with custom versions.

The remaining method descriptions for the JScrollPane class are omitted.

Swing Bugs

Swing has come a long way in terms of quality. The early beta releases contained numerous bugs, many of which have subsequently been fixed. However, like any software, Swing still has its share of bugs. Throughout this book, I have tried to point out as many bugs as possible so that developers can avoid the frustration of debugging code only to discover that a problem is due to a Swing bug.

It is also important to keep in mind that this book is based on Swing 1.1 FCS. As we went to press, a 1.1.1 version of Swing was released that was mostly a bug fix release. Therefore, it is a certainty that some of the bugs cited in this book will have been fixed by the time this book is on the shelves.

Using the CD-ROM

The CD in the back of the book contains the following:

1.1.7 JDK and Swing 1.1.1

1.2 JDK and Swing 1.1 FCS

Code examples from this book

Swing and the JDK

As mentioned previously, Swing 1.1.1 is mostly a bug fix release. Swing 1.1.1 does not work with the 1.2 JDK, and therefore the 1.1.7 JDK is included on the CD. Swing 1.1 FCS works with both the 1.1.7 and 1.2 versions of the JDK.

Code Examples from the Book

All of the code examples in this book that are accompanied by a CD-ROM icon are included on the CD in the back of the book. Figure P-1 shows the directory structure for the code examples. A directory exists for every chapter in the book, and subdirectories are included for each example, which should make it trivial to locate examples. For instance, Example 3-1 on page 91 can be found in the chapters/3/1 directory.

Every example is compiled and ready to run. Some of the directories representing code examples contain a README.txt file if the examples exhibit bugs or have been modified from their listing in the book.

The CD also contains two versions of the GridBagLab application that comes with "Graphic Java 2 Mastering the JFC Volume I: AWT." One version uses external windows and another uses Swing internal frames. The application provides an example of a fairly complete Swing application and illustrates the use of Swing internal frames. Additionally, the application can be used to explore the intricacies of the GridBagLayout layout manager.

Applets Vs. Applications

Most of the code examples discussed in this book are applets, but a fair percentage of examples are applications. Applications are typically implemented instead of applets either because files are manipulated or dialogs are shown--applets have restricted access to files, and under JDK 1.2 dialogs displayed from an applet contain a warning string. Sometimes applications are implemented instead of applets simply for the sake of variety. Nearly all of the applications discussed in this book that do not manipulate files can easily be rewritten as applets.

Conventions Used in This Book

Table P-1 shows the coding conventions used in this book. Coding Conventions

Convention

Example

Class names have initial capital letters.

public class ClassName

Method names have initial lower case and the rest of the words have an initial capital letter.

getLength()

Variable names have initial lower case and the rest of the words have an initial capital letter.

private int length

private int bufferLength

Note that, for the most part, methods are referred to without their arguments; however, arguments are included when the discussion warrants them.

Table P-2 shows the typographic conventions used in this book.

Typographic Conventions

Typeface or Symbol

Description

Indicates that the accompanying code, command, or file is available on the CD that accompanies this book.

courier

Indicates a command, file name, class name, method, argument, Java keyword, HTML tag, file content, or code excerpt.

bold courier

Indicates a sample command-line entry.

italics

Indicates definitions, emphasis, a book title, or a variable that should be replaced with a valid value.

GRAPHIC JAVA 2: MASTERING THE JFC, BY GEARY, 3RD EDITION, VOLUME 2: SWING BY DAVID GEARY PDF

Download: GRAPHIC JAVA 2: MASTERING THE JFC, BY GEARY, 3RD EDITION, VOLUME 2: SWING BY DAVID GEARY PDF

Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary. A task might obligate you to consistently improve the knowledge and also encounter. When you have no adequate time to boost it directly, you could get the experience and understanding from checking out the book. As everyone understands, publication Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary is preferred as the home window to open the globe. It means that reviewing publication Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary will certainly offer you a new means to locate everything that you require. As guide that we will provide here, Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary

Also the price of an e-book *Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary* is so economical; many individuals are really stingy to reserve their money to purchase the e-books. The other factors are that they feel bad and also have no time to head to guide company to search the e-book Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary to read. Well, this is contemporary age; so lots of publications could be got quickly. As this Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary and also much more publications, they could be obtained in really fast means. You will certainly not should go outside to obtain this e-book Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary

By seeing this page, you have actually done the ideal staring factor. This is your start to pick guide Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary that you really want. There are bunches of referred books to read. When you intend to obtain this Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary as your book reading, you can click the link web page to download and install Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary In few time, you have actually owned your referred publications as your own.

GRAPHIC JAVA 2: MASTERING THE JFC, BY GEARY, 3RD EDITION, VOLUME 2: SWING BY DAVID GEARY PDF

Master the Java Foundation Classes with Graphic Java 2, the most complete and comprehensive guide to the AWT, Swing, and the 2D API VOLUME 2: SWING Swing gives Java application programmers world-class tools for building professional, customizable cross-platform GUIs. This comprehensive reference by former Sun engineer David Geary shows experienced programmers how to take full advantage of Swing's power. Practical explanations and robust code examples provide the resources you need to build Java applications with sophisticated graphical user interfaces. Graphic Java is the one exhaustive reference that contains everything you need to know about Swing. In depth explanations are coupled with class diagrams and code examples for all of the key components, including: * Buttons and labels * Progress bars and sliders * Frames, windows, and dialogs * Internal frames and desktop panes * Color and file choosers * Menus and toolbars * Lists and combo boxes * Text components * Tables and trees You'll discover the key design considerations associated with Swing development, including Swing's object-oriented idioms and design patterns, and the pluggable look and feel architecture. You will under

- Sales Rank: #2463171 in Books
- Brand: Prentice Hall
- Published on: 1999-03-22
- Original language: German
- Number of items: 1
- Dimensions: 9.00" h x 3.30" w x 7.00" l, 5.46 pounds
- Binding: Paperback
- 1622 pages

Features

- Great product!

Amazon.com Review
If you're developing software that will be used by a large group of people, you need to give it a good-looking front-end--in Java 2, that means you have to use Swing. An excellent resource, Graphic Java 2: Mastering the JFC, Third Edition (Volume 2: Swing) takes on the Swing components one at a time and shows you how to incorporate them into attractive, efficient programs.

In many ways, Graphic Java 2 is a cookbook. You search the table of contents or index for a reference to the kind of problem you want to solve, then examine the author's examples for the solution (or at least some clues to it). This is the book to turn to if you're wondering how to implement the JComboBox.KeySelectionManager interface (which enables users to select items in a combo box) or compare the various ways of making the JTree component into a file browser. Those are just two of hundreds of examples in David Geary's book.

While most examples don't serve any practical purpose by themselves, they do clearly illustrate how a specific aspect of Swing works. It's easy to adapt the details presented here into your own programs. Geary shows consideration for the reader by presenting all his examples as programs that can be compiled and including them on the enclosed CD-ROM. --David Wall

Review

"At over 1500 pages, Graphic Java 2 is a monolith, heavy with details and weighted with valuable information for all Java UI developers. The coverage is meticulous and fluid, instructive in countless ways, unveiling things you won't find anywhere else. Yet it remains accessible to even a novice Swing developer. The net effect is that the price tag seems trivial given the value you can get from this book...

The coverage is not just impressive in its sheer volume; it's also an easy read with clear, concise explanations and useful examples...

All the fundamentals--in eloquent prose and incredible detail--are laid out before the reader in logical chapters that build on each other to unveil the most complete picture I've seen in print. I can hardly wait to read Volume III." -- DevX Book Reviews, July 29, 1999

"Thinking about making that jump from the AWT to Swing? Not sure where to start or how to go about it? Your search stops with David Geary's Graphic Java 2: Mastering the JFC...

Part 2 covers all the Swing components and their properties in detail; each component allocated a chapter. Two of the most complex components--Table and Tree--are covered in great detail and Chapters 19 and 20, respectively. These two chapters alone are worth the price of the book for serious Swing developers...

All in all, Geary has done such a thorough job in Graphic Java 2 that it can truly be considered the Swing bible. Don't let the size of the book intimidate you, it provides a top-level view as well as microscopic details on all Swing components. It will definitely prove to be an asset in the Swing developer's arsenal." -- Dr. Dobbs Journal, September, 1999

From the Inside Flap

Preface

This book has been my full-time passion for more than a year. After Sun closed its office in Colorado Springs in August of 1997, I decided to forego traditional employment to give the Graphic Java series my complete attention. I was determined that the Swing volume would be the definitive guide to Swing, which meant that it had to be the most comprehensive, accurate and insightful Swing book available. I believe that to be the case, but of course, you will be the final judge.

Before I began writing, I spent a considerable amount of effort designing a book that would be beneficial to Swing novices in addition to developers with some Swing experience. As a software developer who has used numerous object oriented GUI frameworks, I realized that the fastest way to climb a framework's learning curve is to study code examples that illustrate specific concepts. As a result, code examples are the foundation upon which this book is built; nearly 300 code examples are discussed in detail--an average of approximately one code example every 5 pages.

However, code examples in and of themselves, cannot suffice as a reference for developers who have advanced along the Swing learning curve. As a result, every Swing component discussion is accompanied by a component summary that includes class diagrams, an examination of the component's properties and events and a class summary that discusses the public and protected methods implemented by the component.

The first part of this book explores fundamental aspects of Swing including:

Swing applets and applications

Swing and multithreading

Swing's Model-View-Controller architecture

Pluggable Look And Feel

The JComponent class

Borders, icons and actions

Miscellaneous features including timers and Swing utilities

The second part of the book examines Swing's components in detail. Numerous code examples are discussed for every Swing component, from labels and buttons to tables, trees and the text package. For example, more than 150 pages containing 25 code examples are dedicated to the table component, and 120 pages and 20 code examples illustrate how to get the most out of the tree component.

Audience

This book assumes that the reader has a good grasp of the Java language including recent additions to the language such as inner classes. This book also assumes a rudimentary understanding of the AWT; specifically, the delegation event model and the Component and Container classes upon which all Swing components are based. See "Graphic Java 2 Mastering the JFC Volume I: AWT" for a thorough investigation of AWT infrastructure and components.

How To Use This Book

Before diving into the Swing components, it is useful to have an understanding of fundamental concepts such as Swing's Model-View-Controller (MVC) design and pluggable look and feel. The former is discussed in "Swing Component Architecture" on page 71 and the latter is covered in the "Pluggable Look and Feel" on page 317. It is also important to understand the services provided by the JComponent class, which is the ultimate superclass of all lightweight Swing components. The JComponent class is examined in "The JComponent Class" on page 123.

Component Summaries

Each Swing component is introduced with numerous code examples that illustrate various component features. Code examples are followed by component summaries, such as the JScrollPane component summary that is partially listed below.

Component summaries begin with a listing of the component's model, UI delegate, renderer and editor, in addition to the events fired by the component. If a component is a replacement for an AWT component, the AWT component is also listed.

JScrollPane

----

javax.swing.plaf.basic.BasicScrollPaneUI

PropertyChangeEvent

java.awt.ScrollPane

The static relationships that a component maintains with other objects are illustrated with a class diagram, such as the class diagram for the JScrollPane class shown above. Class diagrams are introduced in Appendix A, Class Diagrams .

Properties

Property tables, such as the property table for the JScrollPane class shown below, are used to communicate a component's properties. Property tables include property names, the property's data type and whether a property is boolean, bound (a property change event is fired when the property is changed), simple (no events are fired when the property is changed), constrained (changes to the property can be vetoed), or indexed (a parameter, usually an integer , is used to access the property).

Some properties can be specified when a component is instantiated, and a component may provide setter and getter methods for a property. The manner in which a property can be specified is listed in the Access column of a property table.

Property tables also include a column for property default values.

Property tables are followed by a short description of each of the properties listed in the table.

JScrollPane Properties

Property Name

Data Type

Property Type

Access

Default

columnHeader

JViewport

B

SG

null

columnHeaderView

Component

B

S

null

corner

Component

IB

SG

null

horizontalScrollbar

JScrollBar

B

SG

----

horizontalScrollbarPolicy

int

B

CSG

As needed

rowHeader

JViewport

B

SG

null

rowHeaderView

Component

B

S

null

verticalScrollbar

JScrollBar

B

SG

----

verticalScrollbarPolicy

int

B

CSG

As needed

viewport

JViewport

B

SG

JViewport

viewportBorder

Border

B

SG

null

viewportView

Component

B

CSG

null

columnHeader

-- An instance of JViewport for the column header.

columnHeaderView

-- An instance of Component used as the column header viewport's view.

corner

-- A component that is displayed in one of the

scrollpane's corners. The corner is specified by one of the following strings:

ScrollPaneConstants.UPPER_LEFT_CORNER

ScrollPaneConstants.LOWER_LEFT_CORNER

ScrollPaneConstants.UPPER_RIGHT_CORNER

ScrollPaneConstants.LOWER_RIGHT_CORNER

horizontalScrollbar

-- The horizontal scrollbar used by the

scrollpane. The scrollbar is an instance of JScrollPane.ScrollBar , an extension of JScrollBar that takes into account whether the view contained in the scrollpane implements the Scrollable interface.

The remaining property descriptions for the JScrollPane class are omitted.

Events

Code examples are presented that illustrate event handling for a component. For example, the Tree chapter provides five code examples that illustrate handling of tree mouse, editing, selection, and expansion events.

Class Summaries

Each component summary concludes with a class summary that provides descriptions of the component's constructors and methods, such as the class summary for the JScrollPane class listed below.

JScrollPane

JComponent

ScrollPaneConstants, javax.accessibility.Accessible public JScrollPane()

public JScrollPane(int vsbPolicy, int hsbPolicy)

public JScrollPane(Component view)

public JScrollPane(Component view, int vsbPolicy, int hsbPolicy)

JScrollPane provides four constructors. The integer values passed to JScrollPane

constructors represent the vertical and horizontal scrollbar display policies, in that order. The component passed to the constructors is used as the viewport's view, meaning it is the component that is scrolled by the

scrollpane.

The no-argument constructor constructs a scrollpane with a null component for the viewport's view, and scrollbar display policies that display both the horizontal and vertical scrollbars as needed.

Creation Methods

public JScrollBar createHorizontalScrollBar()

public JScrollBar createVerticalScrollBar()

protected JViewport createViewport()

Like most Swing components, JScrollPane

provides create... methods that create its subcomponents. Unlike most Swing components however, JScrollPane implements the methods that create its scrollbars as public instead of protected because the methods are invoked from the scrollpane's UI delegate.

The createHorizontalScrollBar and createVerticalScrollBar methods both return instances of JScrollPane.ScrollBar , which is an extension of JScrollBar that takes into account whether the scrollpane's view implements the Scrollable

interface. See Interface Summary 13-2 on page 779 for more information on the Scrollable interface.

The createViewport method returns an instance of JViewport that is used as the

scrollpane's default viewport.

As with all Swing components that implement create... methods, the methods can be overridden in extension classes to replace the default sub-components with custom versions.

The remaining method descriptions for the JScrollPane class are omitted.

Swing Bugs

Swing has come a long way in terms of quality. The early beta releases contained numerous bugs, many of which have subsequently been fixed. However, like any software, Swing still has its share of bugs. Throughout this book, I have tried to point out as many bugs as possible so that developers can avoid the frustration of debugging code only to discover that a problem is due to a Swing bug.

It is also important to keep in mind that this book is based on Swing 1.1 FCS. As we went to press, a 1.1.1 version of Swing was released that was mostly a bug fix release. Therefore, it is a certainty that some of the bugs cited in this book will have been fixed by the time this book is on the shelves.

Using the CD-ROM

The CD in the back of the book contains the following:

1.1.7 JDK and Swing 1.1.1

1.2 JDK and Swing 1.1 FCS

Code examples from this book

Swing and the JDK

As mentioned previously, Swing 1.1.1 is mostly a bug fix release. Swing 1.1.1 does not work with the 1.2 JDK, and therefore the 1.1.7 JDK is included on the CD. Swing 1.1 FCS works with both the 1.1.7 and 1.2 versions of the JDK.

Code Examples from the Book

All of the code examples in this book that are accompanied by a CD-ROM icon are included on the CD in the back of the book. Figure P-1 shows the directory structure for the code examples. A directory exists for every chapter in the book, and subdirectories are included for each example, which should make it trivial to locate examples. For instance, Example 3-1 on page 91 can be found in the chapters/3/1 directory.

Every example is compiled and ready to run. Some of the directories representing code examples contain a README.txt file if the examples exhibit bugs or have been modified from their listing in the book.

The CD also contains two versions of the GridBagLab application that comes with "Graphic Java 2 Mastering the JFC Volume I: AWT." One version uses external windows and another uses Swing internal frames. The application provides an example of a fairly complete Swing application and illustrates the use of Swing internal frames. Additionally, the application can be used to explore the intricacies of the GridBagLayout layout manager.

Applets Vs. Applications

Most of the code examples discussed in this book are applets, but a fair percentage of examples are applications. Applications are typically implemented instead of applets either because files are manipulated or dialogs are shown--applets have restricted access to files, and under JDK 1.2 dialogs displayed from an applet contain a warning string. Sometimes applications are implemented instead of applets simply for the sake of variety. Nearly all of the applications discussed in this book that do not manipulate files can easily be rewritten as applets.

Conventions Used in This Book

Table P-1 shows the coding conventions used in this book. Coding Conventions

Convention

Example

Class names have initial capital letters.

public class ClassName

Method names have initial lower case and the rest of the words have an initial capital letter.

getLength()

Variable names have initial lower case and the rest of the words have an initial capital letter.

private int length

private int bufferLength

Note that, for the most part, methods are referred to without their arguments; however, arguments are included when the discussion warrants them.

Table P-2 shows the typographic conventions used in this book.

Typographic Conventions

Typeface or Symbol

Description

Indicates that the accompanying code, command, or file is available on the CD that accompanies this book.

courier

Indicates a command, file name, class name, method, argument, Java keyword, HTML tag, file content, or code excerpt.

bold courier

Indicates a sample command-line entry.

italics

Indicates definitions, emphasis, a book title, or a variable that should be replaced with a valid value.

Most helpful customer reviews

1 of 1 people found the following review helpful.
Caveat Emptor
By KellyRek
Please be aware that if you received this book (from the shipper) with the index missing, you probably have only half the book! Please check your front cover. If it says **PART A** in the lower center, you are missing **PART B**. Also check the last page number; if it says Page 788, then you are missing pages 789-1622.

The book "Graphic Java2, Volume 2, Swing (3rd Edition)" comes in two formats. One version is the big monster -- all 1622 pages in one binding (plus cd). The other version is the two separate books, **PART A** and **PART B** (but the cd is not included here).

So once you have the complete book, you should be impressed with David M. Geary's comprehensive guide to the Swing framework. However, if you are new to Swing, I recommend you first read a tutorial on Swing in some beginner's book (such as "Java All-In-One Desk Reference for Dummies" by Doug Lowe and Barry Burd pages 527-627). Then when you return back to Geary's book, you dig deeper into Swing (at a more intermediate level).

I also have Kim Topley's book "Core Java Foundation Classes." Comparing the two books, Geary uses more diagrams which helps me to learn faster and to visualize the structure of Swing. As a concrete example, in Geary's book on page 1076 is the diagram of Figure 18-2 of "Combo Box Models." He has the tree structure of the different classes and interfaces, visually showing you the Combo Box Model types, what they implement and what they extend. On page 612 of Topley's book is only text, no diagram. Nevertheless, I use both Geary's and Topley's books; they complement one another. Sometimes Topley explains a concept better than Geary; other times it's vice versa.

Geary's book serves a dual purpose. It is both for learning Swing and for use as a reference. The problem with O'Reilly's "Java Foundation Classes in a Nutshell" is that as a reference, it doesn't adequately explain what the different methods (of the various classes) actually do; it merely lists them. Whereas with Geary, the methods are explained -- hence the 1622 pages of material. (The Nutshell book is 731 pages; it's still good as a quick reference, though.)

I also recommend Geary's first book of the series, "Graphic Java_1.2, Volume 1, AWT." It is fun reading and only 878 pages.

Please be aware that Geary's purported books "Graphic Java2, Volume III, Advanced Swing" and "Graphic Java2, Volume IV, 2D API" do not exist. If they had at one time existed, then they have somehow disappeared. I wonder if they were even ever written? It is a shame; David M. Geary is a good writer and I would have enjoyed his other volumes.

1 of 1 people found the following review helpful.
Lots of details, no big picture, and no index
By Aleksandar Totic
I have never used Swing before. It seemed like a right choice for my latest app, whose main target is a small set of cross-platform developers. Because of the audience, I did not have to worry about getting that platform look-and-feel just right, and it is a 6 week project.

I've done a lot of GUI work, so coding a new one is mostly a matter of figuring out "how do they implement this feature that I've done in 3 other toolkits". I like an overview, describinh general philosophy and main features: what's the frame hierarchy, menus, command chains, graphics model.

I did not get this out of this book. There was a lot of detail, but no big picture, no "how do I get things done". The navigation was awkward, especially since there is no index. The index is in Volume III, which I do not have. That alone is just silly. It is as if they wrote one book, and then just chopped it into 2 parts.

I got what I needed from Core Java(TM), Volume I--Fundamentals and Core Java, Vol. 2: Advanced Features. Surprisingly, their swing coverage was enough to get me on the right track, the rest I can google.

6 of 6 people found the following review helpful.
approachable and thorough
By Technomage
I found this book to be a thorough start-to-finish explanation of Swing, with detailed coverage of each component. There is a wealth of online information about Java and Swing, and there are also other Swing books, but I never saw the big picture until I bought this book. A MUST READ for any Java GUI developer. Other reviewers have stated that this book was too complex for beginner and intermediate developers. I strongly disagree. Swing itself is complex and powerful, and to leverage its full potential requires advanced programming skill. However I do not believe it, or this book, could be made any simpler and remain as useful and powerful. If you want simplicity, get a book on VB!

See all 53 customer reviews...

GRAPHIC JAVA 2: MASTERING THE JFC, BY GEARY, 3RD EDITION, VOLUME 2: SWING BY DAVID GEARY PDF

As a result of this publication Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary is offered by on-line, it will ease you not to publish it. you can obtain the soft data of this Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary to conserve in your computer system, gadget, and more gadgets. It depends on your willingness where and also where you will certainly check out Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary One that you have to always keep in mind is that reviewing publication **Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary** will certainly endless. You will certainly have eager to read various other publication after finishing a publication, and it's continually.

Amazon.com Review
If you're developing software that will be used by a large group of people, you need to give it a good-looking front-end--in Java 2, that means you have to use Swing. An excellent resource, Graphic Java 2: Mastering the JFC, Third Edition (Volume 2: Swing) takes on the Swing components one at a time and shows you how to incorporate them into attractive, efficient programs.

In many ways, Graphic Java 2 is a cookbook. You search the table of contents or index for a reference to the kind of problem you want to solve, then examine the author's examples for the solution (or at least some clues to it). This is the book to turn to if you're wondering how to implement the JComboBox.KeySelectionManager interface (which enables users to select items in a combo box) or compare the various ways of making the JTree component into a file browser. Those are just two of hundreds of examples in David Geary's book.

While most examples don't serve any practical purpose by themselves, they do clearly illustrate how a specific aspect of Swing works. It's easy to adapt the details presented here into your own programs. Geary shows consideration for the reader by presenting all his examples as programs that can be compiled and including them on the enclosed CD-ROM. --David Wall

Review
"At over 1500 pages, Graphic Java 2 is a monolith, heavy with details and weighted with valuable information for all Java UI developers. The coverage is meticulous and fluid, instructive in countless ways, unveiling things you won't find anywhere else. Yet it remains accessible to even a novice Swing developer. The net effect is that the price tag seems trivial given the value you can get from this book...

The coverage is not just impressive in its sheer volume; it's also an easy read with clear, concise explanations and useful examples...

All the fundamentals--in eloquent prose and incredible detail--are laid out before the reader in logical chapters that build on each other to unveil the most complete picture I've seen in print. I can hardly wait to read Volume III." -- DevX Book Reviews, July 29, 1999

"Thinking about making that jump from the AWT to Swing? Not sure where to start or how to go about it? Your search stops with David Geary's Graphic Java 2: Mastering the JFC...

Part 2 covers all the Swing components and their properties in detail; each component allocated a chapter. Two of the most complex components--Table and Tree--are covered in great detail and Chapters 19 and 20, respectively. These two chapters alone are worth the price of the book for serious Swing developers...

All in all, Geary has done such a thorough job in Graphic Java 2 that it can truly be considered the Swing bible. Don't let the size of the book intimidate you, it provides a top-level view as well as microscopic details on all Swing components. It will definitely prove to be an asset in the Swing developer's arsenal." -- Dr. Dobbs Journal, September, 1999

From the Inside Flap

Preface

This book has been my full-time passion for more than a year. After Sun closed its office in Colorado Springs in August of 1997, I decided to forego traditional employment to give the Graphic Java series my complete attention. I was determined that the Swing volume would be the definitive guide to Swing, which meant that it had to be the most comprehensive, accurate and insightful Swing book available. I believe that to be the case, but of course, you will be the final judge.

Before I began writing, I spent a considerable amount of effort designing a book that would be beneficial to Swing novices in addition to developers with some Swing experience. As a software developer who has used numerous object oriented GUI frameworks, I realized that the fastest way to climb a framework's learning curve is to study code examples that illustrate specific concepts. As a result, code examples are the foundation upon which this book is built; nearly 300 code examples are discussed in detail--an average of approximately one code example every 5 pages.

However, code examples in and of themselves, cannot suffice as a reference for developers who have advanced along the Swing learning curve. As a result, every Swing component discussion is accompanied by a component summary that includes class diagrams, an examination of the component's properties and events and a class summary that discusses the public and protected methods implemented by the component.

The first part of this book explores fundamental aspects of Swing including:

Swing applets and applications

Swing and multithreading

Swing's Model-View-Controller architecture

Pluggable Look And Feel

The JComponent class

Borders, icons and actions

Miscellaneous features including timers and Swing utilities

The second part of the book examines Swing's components in detail. Numerous code examples are discussed for every Swing component, from labels and buttons to tables, trees and the text package. For example, more than 150 pages containing 25 code examples are dedicated to the table component, and 120 pages and 20 code examples illustrate how to get the most out of the tree component.

Audience

This book assumes that the reader has a good grasp of the Java language including recent additions to the language such as inner classes. This book also assumes a rudimentary understanding of the AWT; specifically, the delegation event model and the Component and Container classes upon which all Swing components are based. See "Graphic Java 2 Mastering the JFC Volume I: AWT" for a thorough investigation of AWT infrastructure and components.

How To Use This Book

Before diving into the Swing components, it is useful to have an understanding of fundamental concepts such as Swing's Model-View-Controller (MVC) design and pluggable look and feel. The former is discussed in "Swing Component Architecture" on page 71 and the latter is covered in the "Pluggable Look and Feel" on page 317. It is also important to understand the services provided by the JComponent class, which is the ultimate superclass of all lightweight Swing components. The JComponent class is examined in "The JComponent Class" on page 123.

Component Summaries

Each Swing component is introduced with numerous code examples that illustrate various component features. Code examples are followed by component summaries, such as the JScrollPane component summary that is partially listed below.

Component summaries begin with a listing of the component's model, UI delegate, renderer and editor, in addition to the events fired by the component. If a component is a replacement for an AWT component, the AWT component is also listed.

JScrollPane

----

javax.swing.plaf.basic.BasicScrollPaneUI

PropertyChangeEvent

java.awt.ScrollPane

The static relationships that a component maintains with other objects are illustrated with a class diagram, such as the class diagram for the JScrollPane class shown above. Class diagrams are introduced in Appendix A, Class Diagrams .

Properties

Property tables, such as the property table for the JScrollPane class shown below, are used to communicate a component's properties. Property tables include property names, the property's data type and whether a property is boolean, bound (a property change event is fired when the property is changed), simple (no events are fired when the property is changed), constrained (changes to the property can be vetoed), or indexed (a parameter, usually an integer , is used to access the property).

Some properties can be specified when a component is instantiated, and a component may provide setter and getter methods for a property. The manner in which a property can be specified is listed in the Access column of a property table.

Property tables also include a column for property default values.

Property tables are followed by a short description of each of the properties listed in the table.

JScrollPane Properties

Property Name

Data Type

Property Type

Access

Default

columnHeader

JViewport

B

SG

null

columnHeaderView

Component

B

S

null

corner

Component

IB

SG

null

horizontalScrollbar

JScrollBar

B

SG

----

horizontalScrollbarPolicy

int

B

CSG

As needed

rowHeader

JViewport

B

SG

null

rowHeaderView

Component

B

S

null

verticalScrollbar

JScrollBar

B

SG

----

verticalScrollbarPolicy

int

B

CSG

As needed

viewport

JViewport

B

SG

JViewport

viewportBorder

Border

B

SG

null

viewportView

Component

B

CSG

null

columnHeader

-- An instance of JViewport for the column header.

columnHeaderView

-- An instance of Component used as the column header viewport's view.

corner

-- A component that is displayed in one of the

scrollpane's corners. The corner is specified by one of the following strings:

ScrollPaneConstants.UPPER_LEFT_CORNER

ScrollPaneConstants.LOWER_LEFT_CORNER

ScrollPaneConstants.UPPER_RIGHT_CORNER

ScrollPaneConstants.LOWER_RIGHT_CORNER

horizontalScrollbar

-- The horizontal scrollbar used by the

scrollpane. The scrollbar is an instance of JScrollPane.ScrollBar , an extension of JScrollBar that takes into account whether the view contained in the scrollpane implements the Scrollable interface.

The remaining property descriptions for the JScrollPane class are omitted.

Events

Code examples are presented that illustrate event handling for a component. For example, the Tree chapter provides five code examples that illustrate handling of tree mouse, editing, selection, and expansion events.

Class Summaries

Each component summary concludes with a class summary that provides descriptions of the component's constructors and methods, such as the class summary for the JScrollPane class listed below.

JScrollPane

JComponent

ScrollPaneConstants, javax.accessibility.Accessible

public JScrollPane()

public JScrollPane(int vsbPolicy, int hsbPolicy)

public JScrollPane(Component view)

public JScrollPane(Component view, int vsbPolicy, int hsbPolicy)

JScrollPane provides four constructors. The integer values passed to JScrollPane

constructors represent the vertical and horizontal scrollbar display policies, in that order. The component passed to the constructors is used as the viewport's view, meaning it is the component that is scrolled by the

scrollpane.

The no-argument constructor constructs a scrollpane with a null component for the viewport's view, and scrollbar display policies that display both the horizontal and vertical scrollbars as needed.

Creation Methods

public JScrollBar createHorizontalScrollBar()

public JScrollBar createVerticalScrollBar()

protected JViewport createViewport()

Like most Swing components, JScrollPane

provides create... methods that create its subcomponents. Unlike most Swing components however, JScrollPane implements the methods that create its scrollbars as public instead of protected because the methods are invoked from the scrollpane's UI delegate.

The createHorizontalScrollBar and createVerticalScrollBar methods both return instances of JScrollPane.ScrollBar , which is an extension of JScrollBar that takes into account whether the scrollpane's view implements the Scrollable

interface. See Interface Summary 13-2 on page 779 for more information on the Scrollable interface.

The createViewport method returns an instance of JViewport that is used as the

scrollpane's default viewport.

As with all Swing components that implement create... methods, the methods can be overridden in extension classes to replace the default sub-components with custom versions.

The remaining method descriptions for the JScrollPane class are omitted.

Swing Bugs

Swing has come a long way in terms of quality. The early beta releases contained numerous bugs, many of which have subsequently been fixed. However, like any software, Swing still has its share of bugs. Throughout this book, I have tried to point out as many bugs as possible so that developers can avoid the frustration of debugging code only to discover that a problem is due to a Swing bug.

It is also important to keep in mind that this book is based on Swing 1.1 FCS. As we went to press, a 1.1.1 version of Swing was released that was mostly a bug fix release. Therefore, it is a certainty that some of the bugs cited in this book will have been fixed by the time this book is on the shelves.

Using the CD-ROM

The CD in the back of the book contains the following:

1.1.7 JDK and Swing 1.1.1

1.2 JDK and Swing 1.1 FCS

Code examples from this book

Swing and the JDK

As mentioned previously, Swing 1.1.1 is mostly a bug fix release. Swing 1.1.1 does not work with the 1.2 JDK, and therefore the 1.1.7 JDK is included on the CD. Swing 1.1 FCS works with both the 1.1.7 and 1.2 versions of the JDK.

Code Examples from the Book

All of the code examples in this book that are accompanied by a CD-ROM icon are included on the CD in the back of the book. Figure P-1 shows the directory structure for the code examples. A directory exists for every chapter in the book, and subdirectories are included for each example, which should make it trivial to locate examples. For instance, Example 3-1 on page 91 can be found in the chapters/3/1 directory.

Every example is compiled and ready to run. Some of the directories representing code examples contain a README.txt file if the examples exhibit bugs or have been modified from their listing in the book.

The CD also contains two versions of the GridBagLab application that comes with "Graphic Java 2 Mastering the JFC Volume I: AWT." One version uses external windows and another uses Swing internal frames. The application provides an example of a fairly complete Swing application and illustrates the use of Swing internal frames. Additionally, the application can be used to explore the intricacies of the GridBagLayout layout manager.

Applets Vs. Applications

Most of the code examples discussed in this book are applets, but a fair percentage of examples are applications. Applications are typically implemented instead of applets either because files are manipulated or dialogs are shown--applets have restricted access to files, and under JDK 1.2 dialogs displayed from an applet contain a warning string. Sometimes applications are implemented instead of applets simply for the sake of variety. Nearly all of the applications discussed in this book that do not manipulate files can easily be rewritten as applets.

Conventions Used in This Book

Table P-1 shows the coding conventions used in this book. Coding Conventions

Convention

Example

Class names have initial capital letters.

public class ClassName

Method names have initial lower case and the rest of the words have an initial capital letter.

getLength()

Variable names have initial lower case and the rest of the words have an initial capital letter.

private int length

private int bufferLength

Note that, for the most part, methods are referred to without their arguments; however, arguments are included when the discussion warrants them.

Table P-2 shows the typographic conventions used in this book.

Typographic Conventions

Typeface or Symbol

Description

Indicates that the accompanying code, command, or file is available on the CD that accompanies this book.

courier

Indicates a command, file name, class name, method, argument, Java keyword, HTML tag, file content, or code excerpt.

bold courier

Indicates a sample command-line entry.

italics

Indicates definitions, emphasis, a book title, or a variable that should be replaced with a valid value.

As one of the book collections to suggest, this *Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary* has some strong factors for you to review. This book is very suitable with just what you need currently. Besides, you will additionally enjoy this book Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary to review because this is among your referred publications to read. When going to get something brand-new based on encounter, entertainment, and also other lesson, you could utilize this book Graphic Java 2: Mastering The Jfc, By Geary, 3Rd Edition, Volume 2: Swing By David Geary as the bridge. Beginning to have reading routine can be undergone from various ways as well as from variant sorts of books