

# **The Definitive Guide to Java Swing**

Third Edition

JOHN ZUKOWSKI

Apress®

## **The Definitive Guide to Java Swing, Third Edition**

**Copyright © 2005 by John Zukowski**

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN (pbk): 1-59059-447-9

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

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

Lead Editor: Steve Anglin

Technical Reviewer: Robert Castaneda

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis, Jason Gilmore, Jonathan Hassell, Chris Mills, Dominic Shakeshaft, Jim Sumser

Assistant Publisher: Grace Wong

Project Manager: Beth Christmas

Copy Edit Manager: Nicole LeClerc

Copy Editor: Marilyn Smith

Production Manager: Kari Brooks-Copony

Production Editor: Ellie Fountain

Compositor: Susan Glinert

Proofreaders: Linda Seifert, Liz Welch

Indexer: Michael Brinkman

Artist: Kinetic Publishing Services, LLC

Cover Designer: Kurt Krames

Manufacturing Manager: Tom Debolski

Distributed to the book trade in the United States by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013, and outside the United States by Springer-Verlag GmbH & Co. KG, Tiergartenstr. 17, 69112 Heidelberg, Germany.

In the United States: phone 1-800-SPRINGER, fax 201-348-4505, e-mail [orders@springer-ny.com](mailto:orders@springer-ny.com), or visit <http://www.springer-ny.com>. Outside the United States: fax +49 6221 345229, e-mail [orders@springer.de](mailto:orders@springer.de), or visit <http://www.springer.de>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Downloads section.

# Contents at a Glance

About the Author .....	xix
About the Technical Reviewers .....	xxi
Acknowledgments .....	xxiii
Introduction .....	xxv
<b>CHAPTER 1</b> Swing Overview .....	1
<b>CHAPTER 2</b> Event Handling with the Swing Component Set .....	17
<b>CHAPTER 3</b> The Model-View-Controller Architecture .....	59
<b>CHAPTER 4</b> Core Swing Components .....	67
<b>CHAPTER 5</b> Toggle Buttons .....	115
<b>CHAPTER 6</b> Swing Menus and Toolbars .....	151
<b>CHAPTER 7</b> Borders .....	211
<b>CHAPTER 8</b> Root Pane Containers .....	235
<b>CHAPTER 9</b> Pop-Ups and Choosers .....	267
<b>CHAPTER 10</b> Layout Managers .....	343
<b>CHAPTER 11</b> Advanced Swing Containers .....	377
<b>CHAPTER 12</b> Bounded Range Components .....	419
<b>CHAPTER 13</b> List Model Controls .....	451
<b>CHAPTER 14</b> Spinner Model Controls .....	509
<b>CHAPTER 15</b> Basic Text Components .....	521
<b>CHAPTER 16</b> Advanced Text Capabilities .....	585
<b>CHAPTER 17</b> Trees .....	623
<b>CHAPTER 18</b> Tables .....	675
<b>CHAPTER 19</b> Drag-and-Drop Support .....	729
<b>CHAPTER 20</b> The Pluggable Look and Feel Architecture .....	741
<b>CHAPTER 21</b> The Undo Framework .....	783

<b>CHAPTER 22</b>	<b>Accessibility</b> .....	<b>805</b>
<b>APPENDIX</b>	<b>UI Manager Properties</b> .....	<b>813</b>
<b>INDEX</b> .....		<b>847</b>

# Contents

About the Author .....	xix
About the Technical Reviewers .....	xxi
Acknowledgments .....	xxiii
Introduction .....	xxv
<b>CHAPTER 1 Swing Overview .....</b>	<b>1</b>
Getting to Know the Swing Components .....	2
AWT Component Replacements .....	3
Non-AWT Upgraded Components .....	5
Event Handling and Layout Management .....	10
Undo Framework .....	11
SwingSet Demonstration .....	11
Swing Component to Chapter Mapping .....	12
Summary .....	15
<b>CHAPTER 2 Event Handling with the Swing Component Set .....</b>	<b>17</b>
Delegation-Based Event Handling .....	17
Event Delegation Model .....	17
Event Listeners As Observers .....	19
Multithreaded Swing Event Handling .....	21
Using SwingUtilities for Mouse Button Identification .....	23
Using Property Change Listeners As Observers .....	26
Managing Listener Lists .....	29
Timer Class .....	34
Swing-Specific Event Handling .....	37
Action Interface .....	37
AbstractAction Class .....	38
KeyStroke Class .....	41
Using Mnemonics and Accelerators .....	46

Swing Focus Management .....	46
Moving the Focus .....	48
Examining Focus Cycles .....	50
FocusTraversalPolicy Class .....	52
KeyboardFocusManager Class .....	55
Verifying Input During Focus Traversal .....	56
Summary .....	57
<b>CHAPTER 3 The Model-View-Controller Architecture .....</b>	<b>59</b>
Understanding the Flow of MVC .....	59
MVC Communication .....	59
UI Delegates for Swing Components .....	60
Sharing Data Models .....	61
Understanding the Predefined Data Models .....	63
Summary .....	65
<b>CHAPTER 4 Core Swing Components .....</b>	<b>67</b>
JComponent Class .....	67
Component Pieces .....	69
JComponent Properties .....	74
Handling JComponent Events .....	80
JToolTip Class .....	84
Creating a JToolTip .....	84
Creating Customized JToolTip Objects .....	84
Displaying Positional Tooltip Text .....	85
Customizing a JToolTip Look and Feel .....	86
ToolTipManager Class .....	86
ToolTipManager Properties .....	87
JLabel Class .....	88
Creating a JLabel .....	89
JLabel Properties .....	89
JLabel Event Handling .....	91
Customizing a JLabel Look and Feel .....	92
Interface Icon .....	92
Creating an Icon .....	93
Using an Icon .....	94
ImageIcon Class .....	94
GrayFilter Class .....	97

AbstractButton Class	98
AbstractButton Properties	98
Handling AbstractButton Events	102
JButton Class	104
Creating a JButton	104
JButton Properties	105
Handling JButton Events	106
Customizing a JButton Look and Feel	108
JPanel Class	110
Creating a JPanel	110
Using a JPanel	110
Customizing a JPanel Look and Feel	112
Summary	112
<b>CHAPTER 5 Toggle Buttons</b>	<b>115</b>
ToggleButtonModel Class	115
ButtonGroup Class	116
JToggleButton Class	119
Creating JToggleButton Components	119
JToggleButton Properties	120
Handling JToggleButton Selection Events	121
Customizing a JToggleButton Look and Feel	124
JCheckBox Class	125
Creating JCheckBox Components	126
JCheckBox Properties	127
Handling JCheckBox Selection Events	130
Customizing a JCheckBox Look and Feel	133
JRadioButton Class	134
Creating JRadioButton Components	135
JRadioButton Properties	136
Grouping JRadioButton Components in a ButtonGroup	136
Handling JRadioButton Selection Events	139
Customizing a JRadioButton Look and Feel	147
Summary	149
<b>CHAPTER 6 Swing Menus and Toolbars</b>	<b>151</b>
Working with Menus	152
Menu Class Hierarchy	156
JMenuBar Class	157

SingleSelectionModel Interface .....	161
JMenuItem Class .....	162
JMenu Class .....	168
JSeparator Class .....	175
JPopupMenu Class .....	176
JCheckBoxMenuItem Class .....	184
JRadioButtonMenuItem Class .....	189
Creating Custom MenuElement Components:	
The MenuElement Interface .....	195
Working with Pop-Ups: The Popup Class .....	200
Creating Pop-Up Components .....	200
A Complete Popup/PopupFactory Usage Example .....	200
Working with Toolbars: The JToolBar Class .....	202
Creating JToolBar Components .....	202
Adding Components to a JToolBar .....	202
JToolBar Properties .....	203
Handling JToolBar Events .....	205
Customizing a JToolBar Look and Feel .....	205
A Complete JToolBar Usage Example .....	206
JToolBar.Separator Class .....	208
Summary .....	208

<b>CHAPTER 7</b>	<b>Borders</b> .....	211
Some Basics on Working with Borders .....	211	
Exploring the Border Interface .....	212	
Introducing BorderFactory .....	215	
Starting with AbstractBorder .....	216	
Examining the Predefined Borders .....	218	
EmptyBorder Class .....	218	
LineBorder Class .....	219	
BevelBorder Class .....	220	
SoftBevelBorder Class .....	222	
EtchedBorder Class .....	223	
MatteBorder Class .....	224	
CompoundBorder Class .....	226	
TitledBorder Class .....	227	
Creating Your Own Borders .....	232	
Summary .....	234	



<b>CHAPTER 8</b>	<b>Root Pane Containers</b> .....	235
	JRootPane Class .....	235
	Creating a JRootPane .....	236
	JRootPane Properties .....	236
	Customizing a JRootPane Look and Feel .....	238
	RootPaneContainer Interface .....	239
	JLayeredPane Class .....	239
	JFrame Class .....	242
	Creating a JFrame .....	243
	JFrame Properties .....	243
	Adding Components to a JFrame .....	245
	Handling JFrame Events .....	245
	Extending JFrame .....	246
	JWindow Class .....	247
	Creating a JWindow .....	247
	JWindow Properties .....	248
	Handling JWindow Events .....	248
	Extending JWindow .....	248
	JDialog Class .....	248
	Creating a JDialog .....	248
	JDialog Properties .....	250
	Handling JDialog Events .....	250
	Extending JDialog .....	252
	JApplet Class .....	252
	Working with a Desktop .....	252
	JInternalFrame Class .....	253
	JDesktopPane Class .....	262
	Summary .....	266
<b>CHAPTER 9</b>	<b>Pop-Ups and Choosers</b> .....	267
	JOptionPane Class .....	267
	Creating a JOptionPane .....	268
	Displaying a JOptionPane .....	271
	Automatically Creating a JOptionPane in a Pop-Up Window .....	274
	JOptionPane Properties .....	280
	Customizing a JOptionPane Look and Feel .....	287

ProgressMonitor Class .....	291
Creating a ProgressMonitor .....	292
Using a ProgressMonitor .....	293
ProgressMonitor Properties .....	296
Customizing a ProgressMonitor Look and Feel .....	297
ProgressMonitorInputStream Class .....	297
Creating a ProgressMonitorInputStream .....	297
Using a ProgressMonitorInputStream .....	298
ProgressMonitorInputStream Properties .....	299
JColorChooser Class .....	300
Creating a JColorChooser .....	301
Using JColorChooser .....	302
JColorChooser Properties .....	307
Customizing a JColorChooser Look and Feel .....	320
JFileChooser Class .....	322
Creating a JFileChooser .....	323
Using JFileChooser .....	323
JFileChooser Properties .....	326
Working with File Filters .....	328
Customizing a JFileChooser Look and Feel .....	336
Summary .....	341

## ■ CHAPTER 10 **Layout Managers** ..... 343

Layout Manager Responsibilities .....	343
LayoutManager Interface .....	344
Exploring the LayoutManager Interface .....	344
Exploring the LayoutManager2 Interface .....	345
FlowLayout Class .....	345
BorderLayout Class .....	347
GridLayout Class .....	349
GridBagLayout Class .....	350
GridBagLayout Rows and Columns .....	353
GridBagConstraints Class .....	353
CardLayout Class .....	357
BoxLayout Class .....	357
Creating a BoxLayout .....	358
Laying Out Components .....	359
OverlayLayout Class .....	365
SizeRequirements Class .....	370

ScrollPaneLayout Class .....	370
ViewportLayout Class .....	371
SpringLayout Class .....	371
Summary .....	375
<b>CHAPTER 11 Advanced Swing Containers .....</b>	<b>377</b>
Box Class .....	377
Creating a Box .....	378
Box Properties .....	379
Working with Box.Filler .....	380
Creating Areas That Grow .....	380
Creating Rigid Areas .....	382
JSplitPane Class .....	383
Creating a JSplitPane .....	384
JSplitPane Properties .....	385
Listening for JSplitPane Property Changes .....	390
Customizing a JSplitPane Look and Feel .....	393
JTabbedPane Class .....	394
Creating a JTabbedPane .....	395
Adding and Removing Tabs .....	397
JTabbedPane Properties .....	398
Listening for Changing Tab Selection .....	399
Customizing a JTabbedPane Look and Feel .....	401
JScrollPane Class .....	403
Creating a JScrollPane .....	404
Changing the Viewport View .....	406
Scrollable Interface .....	406
JScrollPane Properties .....	407
Customizing a JScrollPane Look and Feel .....	410
JViewport Class .....	412
Creating a JViewport .....	412
JViewport Properties .....	412
Customizing a JViewport Look and Feel .....	417
Summary .....	417
<b>CHAPTER 12 Bounded Range Components .....</b>	<b>419</b>
BoundedRangeModel Interface .....	419
DefaultBoundedRangeModel Class .....	420

JScrollBar Class .....	421
Creating JScrollBar Components .....	422
Handling Scrolling Events .....	423
JScrollBar Properties .....	426
Customizing a JScrollBar Look and Feel .....	427
JSlider Class .....	428
Creating JSlider Components .....	428
Handling JSlider Events .....	430
JSlider Properties .....	431
Customizing a JSlider Look and Feel .....	435
JSlider Client Properties .....	438
JProgressBar Class .....	439
Creating JProgressBar Components .....	439
JProgressBar Properties .....	440
Handling JProgressBar Events .....	445
Customizing a JProgressBar Look and Feel .....	446
JTextField Class and BoundedRangeModel Interface .....	447
Summary .....	449

## CHAPTER 13 List Model Controls .....

ListModel Interface .....	451
AbstractListModel Class .....	452
DefaultListModel Class .....	453
Listening for ListModel Events with a ListDataListener .....	454
ComboBoxModel Interface .....	460
MutableComboBoxModel Interface .....	460
DefaultComboBoxModel Class .....	460
JList Class .....	463
Creating JList Components .....	463
JList Properties .....	464
Scrolling JList Components .....	466
Rendering JList Elements .....	468
Selecting JList Elements .....	473
Displaying Multiple Columns .....	479
Customizing a JList Look and Feel .....	480
Creating a Dual List Box .....	481
Adding Element-Level Tooltips to List Items .....	488

JComboBox Class .....	490
Creating JComboBox Components .....	491
JComboBox Properties .....	491
Rendering JComboBox Elements .....	493
Selecting JComboBox Elements .....	493
Editing JComboBox Elements .....	497
Customizing a JComboBox Look and Feel .....	503
Sharing the Data Model for a JComboBox and JList .....	506
Summary .....	508
<b>CHAPTER 14 Spinner Model Controls .....</b>	<b>509</b>
JSpinner Class .....	509
Creating JSpinner Components .....	510
JSpinner Properties .....	510
Listening for JSpinner Events with a ChangeListener .....	511
Customizing a JSpinner Look and Feel .....	512
SpinnerModel Interface .....	513
AbstractSpinnerModel Class .....	513
SpinnerDateModel Class .....	514
SpinnerListModel Class .....	515
SpinnerNumberModel Class .....	516
Custom Models .....	517
JSpinner Editors .....	518
JSpinner.DefaultEditor Class .....	518
JSpinner.DateEditor Class .....	519
JSpinner.ListEditor Class .....	519
JSpinner.NumberEditor Class .....	520
Summary .....	520
<b>CHAPTER 15 Basic Text Components .....</b>	<b>521</b>
Overview of the Swing Text Components .....	521
JTextComponent Class .....	523
JTextComponent Properties .....	523
JTextComponent Operations .....	526
JTextField Class .....	526
Creating a JTextField .....	527
Using JLabel Mnemonics .....	527
JTextField Properties .....	529

JTextComponent Operations with a JTextField . . . . .	530
Document Interface . . . . .	537
DocumentListener and DocumentEvent Interfaces . . . . .	546
Caret and Highlighter Interfaces . . . . .	547
CaretListener Interface and CaretEvent Class . . . . .	550
NavigationFilter Class . . . . .	552
Keymap Interface . . . . .	554
JTextComponent.KeyBinding Class . . . . .	556
Handling JTextField Events . . . . .	556
Customizing a JTextField Look and Feel . . . . .	562
JPasswordField Class . . . . .	563
Creating a JPasswordField . . . . .	563
JPasswordField Properties . . . . .	564
Customizing a JPasswordField Look and Feel . . . . .	565
JFormattedTextField Class . . . . .	566
Creating a JFormattedTextField . . . . .	566
JFormattedTextField Properties . . . . .	567
Customizing a JFormattedTextField Look and Feel . . . . .	569
JTextArea Class . . . . .	570
Creating a JTextArea . . . . .	570
JTextArea Properties . . . . .	571
Handling JTextArea Events . . . . .	572
Customizing a JTextArea Look and Feel . . . . .	572
JEditorPane Class . . . . .	574
Creating a JEditorPane . . . . .	575
JEditorPane Properties . . . . .	575
Handling JEditorPane Events . . . . .	576
Customizing a JEditorPane Look and Feel . . . . .	579
JTextPane Class . . . . .	580
Creating a JTextPane . . . . .	580
JTextPane Properties . . . . .	580
Customizing a JTextPane Look and Feel . . . . .	581
Loading a JTextPane with Content . . . . .	582
Summary . . . . .	584
<b>CHAPTER 16 Advanced Text Capabilities . . . . .</b>	<b>585</b>
Using Actions with Text Components . . . . .	585
Listing Actions . . . . .	586
Using Actions . . . . .	589
Finding Actions . . . . .	591

Creating Styled Text .....	595
StyledDocument Interface and DefaultStyledDocument Class ...	595
AttributeSet Interface .....	597
MutableAttributeSet Interface .....	597
SimpleAttributeSet Class .....	597
StyleConstants Class .....	601
TabStop and TabSet Classes .....	603
Style Interface .....	606
StyleContext Class .....	606
The Editor Kits .....	607
Loading HTML Documents .....	607
Iterating Through HTML Documents .....	608
JFormattedTextField Formats .....	612
Dates and Numbers .....	612
Input Masks .....	618
DefaultFormatterFactory Class .....	620
Summary .....	621

## CHAPTER 17 Trees .....

Introducing Trees .....	623
JTree Class .....	624
Creating a JTree .....	624
Scrolling Trees .....	627
JTree Properties .....	628
Customizing a JTree Look and Feel .....	630
TreeCellRenderer Interface .....	634
DefaultTreeCellRenderer Class .....	635
DefaultTreeCellRenderer Properties .....	635
Creating a Custom Renderer .....	637
Working with Tree Tooltips .....	641
Editing Tree Nodes .....	643
CellEditor Interface .....	644
TreeCellEditor Interface .....	644
DefaultCellEditor Class .....	645
DefaultTreeCellEditor Class .....	647
Creating a Proper ComboBox Editor for a Tree .....	648
Creating an Editor Just for Leaf Nodes .....	648
CellEditorListener Interface and ChangeEvent Class .....	650
Creating a Better Check Box Node Editor .....	650

Working with the Nodes of the Tree .....	659
TreeNode Interface .....	659
MutableTreeNode Interface .....	660
DefaultMutableTreeNode Class .....	661
Traversing Trees .....	664
JTree.DynamicUtilTreeNode Class .....	666
TreeModel Interface .....	667
DefaultTreeModel Class .....	667
TreeModelListener Interface and TreeModelEvent Class .....	668
TreeSelectionModel Interface .....	668
DefaultTreeSelectionModel Class .....	670
TreeSelectionListener Interface and TreeSelectionEvent Class .....	671
TreePath Class .....	671
Additional Expansion Events .....	672
TreeExpansionListener Interface and TreeExpansionEvent Class .....	672
TreeWillExpandListener Interface and ExpandVetoException Class .....	673
Summary .....	674
<b>CHAPTER 18 Tables .....</b>	<b>675</b>
Introducing Tables .....	675
JTable Class .....	677
Creating a JTable .....	677
Scrolling JTable Components .....	678
Manually Positioning the JTable View .....	679
Removing Column Headers .....	680
JTable Properties .....	680
Rendering Table Cells .....	686
Handling JTable Events .....	689
Customizing a JTable Look and Feel .....	689
TableModel Interface .....	690
AbstractTableModel Class .....	691
DefaultTableModel Class .....	696
Sorting JTable Elements .....	700
TableColumnModel Interface .....	707
DefaultTableColumnModel Class .....	708
Listening to JTable Events with a TableColumnModelListener .....	709
TableColumn Class .....	712



JTableHeader Class	715
Creating a JTableHeader	716
JTableHeader Properties	716
Using Tooltips in Table Headers	716
Customizing a JTableHeader Look and Feel	717
Editing Table Cells	718
TableCellEditor Interface and DefaultCellEditor Class	718
Creating a Simple Cell Editor	718
Creating a Complex Cell Editor	722
Printing Tables	724
Summary	728
<b>CHAPTER 19 Drag-and-Drop Support</b>	<b>729</b>
Built-in Drag-and-Drop Support	729
TransferHandler Class	731
Drag-and-Drop Support for Images	733
Summary	740
<b>CHAPTER 20 The Pluggable Look and Feel Architecture</b>	<b>741</b>
LookAndFeel Class	741
Listing the Installed Look and Feel Classes	742
Changing the Current Look and Feel	743
Customizing the Current Look and Feel	747
Creating a New Look and Feel	767
Using the WindowsLookAndFeel on a Non-Windows Machine	767
Adding UI Delegates	771
Working with Metal Themes	772
MetalTheme Class	772
DefaultMetalTheme and OceanTheme Classes	774
Using an Auxiliary Look and Feel	776
SynthLookAndFeel Class	777
Configuring Synth	777
Default Synth Properties	780
Working with Synth Images	780
Summary	781

<b>CHAPTER 21</b>	<b>The Undo Framework</b>	783
	Working with the Undo Framework	783
	Using the Undo Framework with Swing Text Components	784
	The Command Design Pattern	788
	Undo Framework Components	789
	UndoableEdit Interface	789
	AbstractUndoableEdit Class	791
	CompoundEdit Class	791
	UndoManager Class	792
	UndoableEditListener Interface and UndoableEditEvent Class	794
	UndoableEditSupport Class	794
	A Complete Undoable Program Example	795
	Using an Outside Object to Manage Undo States	800
	StateEditable Interface	800
	StateEdit Class	801
	A Complete StateEditable/StateEdit Example	801
	Summary	804
<b>CHAPTER 22</b>	<b>Accessibility</b>	805
	Accessibility Classes	805
	Accessible Interface	806
	AccessibleContext Class	806
	Creating Accessible Components	807
	Working with the Java Access Bridge	808
	Summary	811
<b>APPENDIX</b>	<b>UI Manager Properties</b>	813
<b>INDEX</b>		847

# About the Author



**JOHN ZUKOWSKI** has been involved with the Java platform since it was just called Java, pushing ten years now. He currently writes a monthly column for Sun's Core Java Technologies Tech Tips (<http://java.sun.com/developer/JDCTechTips/>) and IBM's developerWorks (<http://www-136.ibm.com/developerworks/java/>). He has contributed content to numerous other sites, including jGuru (<http://www.jguru.com>), DevX (<http://www.devx.com/>), Intel (<http://www.intel.com/>), and JavaWorld (<http://www.javaworld.com/>).

He is the author of many other popular titles on Java, including *Java AWT Reference* (O'Reilly and Associates), *Mastering Java 2* (Sybex), *Borland's JBuilder: No Experience Required* (Sybex), *Learn Java with JBuilder 6* (Apress), *Java Collections* (Apress), and *Definitive Guide to Swing for Java 2* (Apress).

# About the Technical Reviewers

**T**his book was technically reviewed by Daren Klamer, David Vittor, Hido Hasimbegovic, Charlie Castaneda, and Robert Castaneda, who are all part of the CustomWare Asia Pacific team working on numerous Java and integration-based projects in Australia and the Asia Pacific region. Their web site is <http://www.customware.net>.

# Acknowledgments

**T**his book has been a long time coming, with various starts and stops, and getting sidetracked a few times along the way. Now that it is all done, I need to thank those who helped.

For starters, I want to thank everyone at Apress who hung in there and had patience when dealing with me throughout the project, especially project manager Beth Christmas, who I'm sure I drove nuts at times, and editor Steve Anglin, who kept nudging me along. On the production side, I'd like to thank Marilyn Smith for all the input and updates, Ellie Fountain for her hard work at ensuring little changes got done right, and, of course, my technical reviewer Rob Castaneda and the team at CustomWare for all the input on my rough work. Congrats on that marriage thing.

Some of the images used in the sample programs were made by Deb Felts, who ran a web site called the Image Addict's Attic. The site doesn't seem to be online any more, but the images are used with permission and she does retain copyright on them. Sun also maintains the Java Look and Feel Graphics Repository at <http://java.sun.com/developer/techDocs/hi/repository/>, with its own set of images to be used for Java applications.

For all the readers out there, thanks for asking me to do the update. Without your continued support, you wouldn't be holding this book in your hands.

For their continued encouragement along the way, I'd like to personally thank the following: Joe Sam Shirah, thanks for doing that long drive to visit while I was in Florida for the conference; my Aunt Mary Hamfeldt, congrats on your first grandchild; our Realtor Nancy Moore, thanks for putting up with us for so long; Miguel Muniz, thanks for all the bug reports at SavaJe; Matthew B. Doar, thanks for JDiff (<http://www.jdiff.org/>), a great little Java doclet for reporting API differences. Happy tenth birthday, Duke and Java.

I am forever grateful to my wife, Lisa, for her support, and our dog, Jaeger, for his playfulness. Thanks to Dad, too. Good luck at the casinos.

# Introduction

**W**elcome to *Learn Java 5.0 Swing in a Nutshell for Dummies in 21 Days*. Since the beginning of Java time (1995), the component libraries have been actively evolving. What began as a small set of nine AWT components, plus menus and containers, has grown to a more complete and complex set of around 50 Swing components—all just to create graphical user interfaces (GUIs) for your Java client-side programs. That’s where this book comes in. Its purpose is to make your life easier in creating those GUIs.

Earlier editions of this book took the approach that if the class wasn’t found in the `javax.swing` package, it wasn’t covered in the book. This third edition takes a more complete view of creating GUIs. For instance, instead of just describing the Swing layout managers, there is also material on the AWT layout managers, since you’re likely to be using them.

The first edition of this book was written for a mix of the Java 1.1 and 1.2 developer. The second edition hit the 1.3 platform. This edition is wholly for the 5.0 developer. Almost all the programs will not work on a 1.4 platform, though with a little tweaking, they can be made to do so.

In this book, you’ll find a tutorial-like approach to learning about the Swing libraries and related capabilities. It is not an API reference book, nor is it a primer that describes how to install the Java Development Kit (JDK), compile your programs, or run them. If you need help in those areas, consider using an integrated development environment (IDE)—such as IntelliJ IDEA, Eclipse, or Borland’s JBuilder—or get one of Apress’s other books, such as *Beginning Java Objects*, by Jacquie Barker.

Is this book for you? If you are new to the Java platform, you might want to start with a more introductory text first, before jumping on the Swing bandwagon. On the other hand, if you’ve been working with Java for a while and have decided it’s time to start using the Swing component set, you’ll find this book extremely useful. With this book, you won’t have to drudge through the countless Swing classes for a way to accomplish that impossible task. You’ll become much more productive more quickly, and you’ll be able to make the most of the many reusable components and techniques available with Swing.

## Book Structure

This book can be read from cover to cover, but it doesn’t have to be done that way. It’s true that later sections of the book assume you’ve absorbed knowledge from the earlier sections. However, if you want to find something on a topic covered in a later chapter, you don’t need to read all the chapters that precede it first. If you come across something that’s unfamiliar to you, you can always go back to the earlier chapter or search the index to locate the information you need.

The contents of this book are grouped into three logical sections:

Chapters 1 through 4 provide general knowledge that will prove to be useful as you read through the remainder of the book. In Chapter 1, you'll find an overview of the Swing component set. Chapter 2 details event handling with the Swing component set. It describes the delegation-based event model and focus management policies used by Swing. In Chapter 3, you'll learn about the Model-View-Controller (MVC) architecture. You can avoid using MVC if you wish, but to take full advantage of everything that Swing has to offer, it helps to have a good grasp of MVC concepts. In Chapter 4, you'll find the beginning coverage of the specific Swing components. All Swing components share many of the same attributes, and in Chapter 4, you'll learn the foundation for those common behaviors.

In Chapters 5 through 15, you'll discover the many aspects of the reusable Swing components. You'll find out about menus, toolbars, borders, high-level containers, pop-up dialogs, layout managers, advanced Swing containers, bounded range components, toggle components, list model components, spinners, and text components. Most of what you'll want to accomplish with the Swing libraries is discussed in these chapters.

In Chapters 16 through 22, some of the more advanced Swing topics are covered. These tend to be the areas that even the experienced developers find the most confusing. Chapter 16 goes beyond the basics of text component handling found in Chapter 15. Chapters 17 and 18 deal with the Swing tree and table components. These components allow you to display hierarchical or tabular data. In Chapter 19, you'll learn about drag-and-drop support in Swing. Chapter 20 explores how to customize the appearance of your application. Because the Swing libraries are completely Java-based, if you don't like the way something is done or how it appears, you can change it. In Chapter 21, you'll learn about the undo framework, which offers undo and redo support for your applications. Finally, in Chapter 22, you finish off with a look into the accessibility framework offered by Swing, such as support for screen readers and magnifying glasses to help those needing assistive technologies.

The Appendix contains a list of about 1,000 settable properties the user interface manager employs to configure the appearance of the Swing components for the current look and feel. The Swing components manage various defaults, such as colors and fonts applied to components, so you don't need to subclass a component in order to customize its appearance. Appendix A gathers all of the property settings listed throughout the chapters into one comprehensive list for easy reference.

## Support

You can head to many places online to get technical support for Swing and answers to general Java questions. Here's a list of some of the more useful places around:

- The Java Ranch at <http://www.javaranch.com/> offers forums for just about everything in the Big Moose Saloon.
- Java Forums at <http://forums.java.sun.com/> are Sun's online forums for Java development issues.

- developerWorks at <http://www.ibm.com/developerworks/java/> is the IBM's developer community for Java with forums and tutorials.
- jGuru at <http://www.jguru.com> offers a series of FAQs and forums for finding answers.
- Marcus Green's Java Certification Exam Discussion Forum at <http://www.jchq.net/discus/> provides support for those going the certification route.

While I would love to be able to answer all reader questions, I get swamped with e-mail and real-life responsibilities. Please consider using these resources to get help.

## About Java

Java is one of 13,000 islands that makes up Indonesia, whose capital is Jakarta. It is home to about 120 million people with an area about 50,000 square miles (132,000 square kilometers). While on the island, you can hear traditional music such as gamelan or angklung and enjoy Java's main export, a coffee that is considered spicy and full-bodied, with a strong, slightly acidic flavor. The island also has a dangerous volcano named Merapi, which makes up part of the Pacific "Ring of Fire." In 1891, on the island, Eugene Dubois discovered fossils from *Pithecanthropus erectus*, better known as Java man (*homo javanensis*).

For more information, see [http://encyclopedia.lockergnome.com/s/b/Java\\_\(island\)](http://encyclopedia.lockergnome.com/s/b/Java_(island)).





# Swing Overview

**A**ccording to *Encyclopedia Britannica*, Swing was a popular music in the United States, circa 1930–1945. Okay, maybe not in the Java sense. Instead, on May 23, 1995, John Gage, then director of the Science Office for Sun, introduced Java to the world. With its birth came something called the Abstract Window Toolkit, or AWT. In turn, with AWT came native widgets, and with this early native widget set came . . . trouble.

The original component set that came with the Java platform, AWT, was dependent on too many idiosyncrasies of the underlying platform. Instead of providing a mature-looking component set, Java offered the lowest common denominator version. If a feature wasn't available on all Java platforms, it wasn't available on any Java platform. And then you had to deal with all the browser/platform differences. Each Java runtime environment relied on how the component set was connected with the underlying platform-specific native widget set. If there were issues with the connection, first, they were specific to the platform (and/or browser) and second, you had to code around these problems so your programs could be write-once, run anywhere (WORA), the Java mantra of the time.

As Java technologies became more popular, users realized AWT was extremely slow and unreliable, and you couldn't really do much with the provided components. Very few of them were available, and you couldn't use them in a visual programming environment. So, new technologies were introduced, such as just-in-time (JIT) compilers to improve performance and, with Borland's help, JavaBeans for a component-based development.

With these new technologies came more and more widget sets, for the AWT component set itself was very basic. So, applet download times grew and grew, because these new widget sets weren't part of the core Java platform, and Java archive (JAR) files were introduced to improve delivery time. Eventually, each of the major browser vendors added its favorite component library to its virtual machine—AFC, IFC, and WFC, to name just a few. Yet all the libraries used different design models, and there were no true cross-browser standards.

Eventually, Sun Microsystems teamed up with Netscape Communication and other partners to create yet another library called the Java Foundation Classes, or JFC. Part of JFC is something called the Swing component set. This Swing component set is what this book is all about.

---

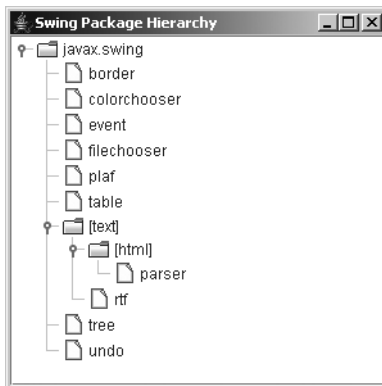
**Note** Later technologies were introduced to help people use the Swing components within a browser and with web-based application delivery. These include the Java Plug-in (<http://java.sun.com/products/plugin/>) and Java Web Start (<http://java.sun.com/products/javawebstart/>). Alternatives to Swing, like the SWT component set with Eclipse (<http://www.eclipse.org/swt/>), have also been created. These are not discussed here.

---

This chapter will familiarize you with the various Swing pieces. For starters, there is the component set. Without these, there is no Swing. Next, you'll peek at the world of event handling and layout management common to both AWT and Swing components. After that, you'll take a quick look at the undo/redo framework available within the Swing architecture. Then you'll explore the SwingSet2 demonstration provided with the Java 2 Platform Standard Edition 5.0 Development Kit (JDK 5.0) so that you can see some of the capabilities. Lastly, I'll point out where in the book all these capabilities are discussed in detail.

## Getting to Know the Swing Components

The book will serve as a guide to development using the Swing component set. Over the course of its pages, you'll look at every package in the `javax.swing` package hierarchy, as shown in Figure 1-1.



**Figure 1-1.** *The Swing package hierarchy*

---

**Note** The `javax.swing.plaf` package contains several subpackages and related packages, some of which are located outside the `javax.swing` package hierarchy. *Plaf* stands for pluggable look and feel—a Swing concept that will be described more fully in Chapter 20.

---

The Swing component set is one big group of components. While the JDK 5.0 release didn't add any new Swing components to the mix, logically, you can think of them as those with duplicate components within AWT and those without.

## AWT Component Replacements

The Swing component set was originally created because the basic AWT components that came with the original version of the Java libraries were insufficient for real-world, forms-based applications. All the basic components were there, but the existing set was too small and far too restrictive. For instance, you couldn't even put an image on a button. To alleviate this situation, the Swing component set offers replacements for each of the AWT components. The Swing components support all the capabilities of the original set and offer a whole lot more besides. As such, you should never need to deal with any of the basic AWT components.

---

**Note** Although the Swing components replace the AWT components, you'll still need to understand several basic AWT concepts, such as layout managers, event handling, and drawing support. In addition, you'll need to grasp the concept that all of Swing is built on top of the core AWT libraries.

---

The basic distinction between the Swing and equivalent AWT components is, in most cases, the Swing component class names begin with a *J* and the AWT ones don't. Swing's `JButton` is a replacement for the AWT `Button` component. One exception is the `JComboBox`, which replaces the AWT `Choice` component.

At the application programming interface (API) level, the Swing components are almost always a superset of the features the AWT components support. While they support additional capabilities, the basic AWT capabilities are there for everything but the `JList` component, whose API is completely unlike that of the AWT `List` component. Table 1-1 maps the original AWT components to their replacement Swing components.

**Table 1-1.** *AWT to Swing Component Mapping*

AWT Component	Nearest Swing Replacement
Button	<code>JButton</code>
Canvas	<code>JPanel</code>
Checkbox	<code>JCheckBox</code>
Checkbox in <code>CheckboxGroup</code>	<code>JRadioButton</code> in <code>ButtonGroup</code>
Choice	<code>JComboBox</code>
Component	<code>JComponent</code>
Container	<code>JPanel</code>
Label	<code>JLabel</code>
List	<code>JList</code>

**Table 1-1.** *AWT to Swing Component Mapping (Continued)*

<b>AWT Component</b>	<b>Nearest Swing Replacement</b>
Menu	JMenu
MenuBar	JMenuBar
MenuItem	JMenuItem
Panel	JPanel
PopupMenu	JPopupMenu
Scrollbar	JScrollBar
ScrollPane	JScrollPane
TextArea	JTextArea
TextField	JTextField

---

**Note** For most people, the fact that the Swing components replace AWT components is irrelevant. Just treat the Swing components as an *independent* component set, and you'll be perfectly okay.

---

To help you understand how to use the Swing components, you'll examine each of the components in this book. For instance, Chapter 4 looks at how the JButton component works, with just a single line of text as its label, like an AWT Button, but adds capabilities, such as using image icons on buttons and working with multiple lines of text. To find out where each component is discussed in this book, see the "Swing Component to Chapter Mapping" section later in this chapter.

In addition to replacing each of the basic components, the Swing component set has a replacement for the higher-level window objects. Although the only change in most of the components' names is the beginning *J*, you'll discover in Chapter 8 how the high-level container objects are much different in the Swing world. Swing's replacement for the old FileDialog object differs even more and is discussed in Chapter 9. Table 1-2 maps the high-level window objects from the AWT component world to the Swing universe.

**Table 1-2.** *AWT to Swing Window Mapping*

<b>AWT Window</b>	<b>Nearest Swing Replacement</b>
Applet	JApplet
Dialog	JDialog
FileDialog	JFileChooser
Frame	JFrame
Window	JWindow

Whereas the AWT components rely on the user's operating system to provide the actual component to a Java program, Swing components are all controlled from within the Java runtime. The AWT approach is called either the *heavyweight* or the *peered* approach; most Swing components are *lightweight* or *peerless*. You'll explore the basics of this approach in Chapter 4 with the `JComponent`. Additional features for customizing the look and feel of components are discussed in Chapter 20.

## Non-AWT Upgraded Components

In addition to offering replacements for all the basic AWT components, the Swing component set includes twice as many new components.

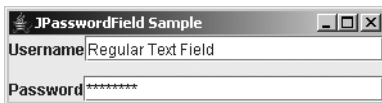
---

**Note** If you're new to Java, just think of all of these components—both the AWT component replacements and those that were not in the AWT—as one big set of components, versus two distinct sets.

---

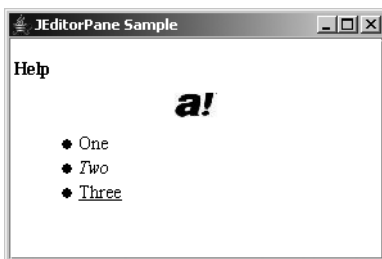
Here's a look at those components that didn't originate in the AWT world:

- `JPasswordField`: This specialized text field is for password entry, as shown in Figure 1-2. You cannot use cut or copy operations within the component, but you can paste text into it.



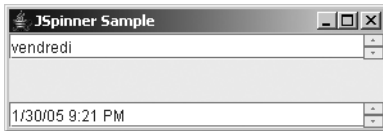
**Figure 1-2.** *The Swing `JPasswordField`*

- `JEditorPane` and `JTextPane`: These two components provide support for displaying and editing multiple-attributed content, such as an HTML and RTF viewer. Figure 1-3 shows a `JEditorPane` component.



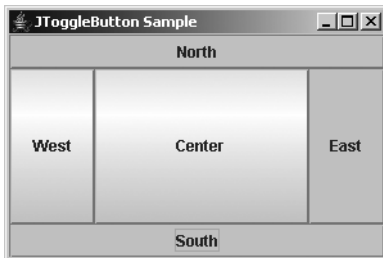
**Figure 1-3.** *The Swing `JEditorPane`*

- **JSpinner:** This component, shown in Figure 1-4, provides selection from an ordered set of predefined values, offering arrows to scroll through the next and previous choices. The predefined values can be an array of strings, a sequential set of numbers, or a date.



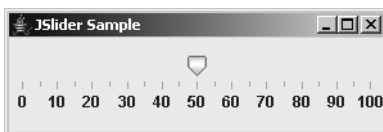
**Figure 1-4.** *The Swing JSpinner*

- **JToggleButton:** This component offers a button that stays depressed when selected. In the example shown in Figure 1-5, the North, East, and South buttons are depressed.



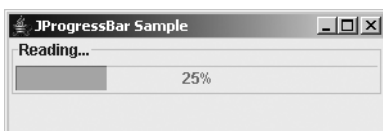
**Figure 1-5.** *The Swing JToggleButton*

- **JSlider:** This component is like the Scrollbar component of AWT (or JScrollBar in the Swing component set). However, its purpose in Swing is for user input. It offers various clues to help the user choose a value. Figure 1-6 shows an example of a JSlider component.



**Figure 1-6.** *The Swing JSlider*

- **JProgressBar:** This component allows the user to visually see the progress of an activity. Some options available include showing static text or percentage done, as shown in Figure 1-7.



**Figure 1-7.** *The Swing JProgressBar*