
Cross-Platform GUI Programming with wxWidgets

BRUCE PERENS' OPEN SOURCE SERIES

<http://www.phptr.com/perens>

- ◆ Java™ Application Development on Linux®
Carl Albing and Michael Schwarz
- ◆ C++ GUI Programming with Qt 3
Jasmin Blanchette and Mark Summerfield
- ◆ Managing Linux Systems with Webmin: System Administration and Module Development
Jamie Cameron
- ◆ The Linux Book
David Elboth
- ◆ Understanding the Linux Virtual Memory Manager
Mel Gorman
- ◆ PHP 5 Power Programming
Andi Gutmans, Stig Bakken, and Derick Rethans
- ◆ Linux® Quick Fix Notebook
Peter Harrison
- ◆ Linux Desk Reference, Second Edition
Scott Hawkins
- ◆ Implementing CIFS: The Common Internet File System
Christopher Hertel
- ◆ Open Source Security Tools: A Practical Guide to Security Applications
Tony Howlett
- ◆ Apache Jakarta CommonsReusable Java™ Components
Will Iverson
- ◆ Embedded Software Development with eCos
Anthony Massa
- ◆ Rapid Application Development with Mozilla
Nigel McFarlane
- ◆ Subversion Version Control: Using the Subversion Version Control System in Development Projects
William Nagel
- ◆ Linux Assembly Language Programming
Bob Neveln
- ◆ Intrusion Detection with SNORT: Advanced IDS Techniques Using SNORT, Apache, MySQL, PHP, and ACID
Rafeeq Ur Rehman
- ◆ Cross-Platform GUI Programming with wxWidgets
Julian Smart and Kevin Hock with Stefan Csomor
- ◆ Samba-3 by Example: Practical Exercises to Successful Deployment
John H. Terpstra
- ◆ The Official Samba-3 HOWTO and Reference Guide
John H. Terpstra and Jelmer R. Vernooij, Editors
- ◆ Real World Linux Security, Second Edition
Bob Toxen

Cross-Platform GUI Programming with wxWidgets

Julian Smart and Kevin Hock
with Stefan Csomor



Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris
Madrid • Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U. S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the U. S., please contact:

International Sales
international@pearsoned.com



This Book Is Safari Enabled

The Safari Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days. Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information when ever and wherever you need it.

To gain 45-day Safari Enabled access to this book:

- Go to <http://www.awprofessional.com/safarienabled>
- Complete the brief registration form
- Enter the coupon code 33G2-XGSI-ZFYZ-D7IV-B9HE

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail customer-service@safaribooksonline.com.

Visit us on the Web: www.phptr.com

Library of Congress Catalog Number: 2005924108

Copyright © 2006 Pearson Education, Inc.

Printed in the United States of America.

This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

ISBN 0-13-147381-6

Text printed in the United States on recycled paper at R.R. Donnelley & Sons in Crawfordsville, Indiana.
First printing, July 2005

About Prentice Hall Professional Technical Reference

With origins reaching back to the industry's first computer science publishing program in the 1960s, and formally launched as its own imprint in 1986, Prentice Hall Professional Technical Reference (PH PTR) has developed into the leading provider of technical books in the world today. Our editors now publish over 200 books annually, authored by leaders in the fields of computing, engineering, and business.

Our roots are firmly planted in the soil that gave rise to the technical revolution. Our bookshelf contains many of the industry's computing and engineering classics: Kernighan and Ritchie's *C Programming Language*, Nemeth's *UNIX System Administration Handbook*, Horstmann's *Core Java*, and Johnson's *High-Speed Digital Design*.

PH PTR acknowledges its auspicious beginnings while it looks to the future for inspiration. We continue to evolve and break new ground in publishing by providing today's professionals with tomorrow's solutions.



Contents

Foreword by Mitch Kaporxxiii
Prefacexxv
Acknowledgmentsxxxiii
About the Authorsxxxv
1 Introduction1
What Is wxWidgets?1
Why Use wxWidgets?2
A Brief History of wxWidgets5
The wxWidgets Community6
wxWidgets and Object-Oriented Programming7
License Considerations7
The wxWidgets Architecture8
wxMSW8
wxGTK8
wxX119
wxMotif10
wxMac10
wxCocoa10
wxWinCE10
wxPalmOS12
wxOS212
wxMGL12
Internal Organization12
Summary13
2 Getting Started15
A Small wxWidgets Sample15
The Application Class16

The Frame Class	17
The Event Handlers	18
The Frame Constructor	19
The Whole Program	20
Compiling and Running the Program	22
Program Flow	23
Summary	23
3 Event Handling	25
Event-Driven Programming	25
Event Tables and Handlers	26
Skipping Events	29
Pluggable Event Handlers	29
Dynamic Event Handlers	30
Window Identifiers	31
Defining Custom Events	34
Summary	37
4 Window Basics	39
Anatomy of a Window	39
The Concept of a Window	40
Client and Non-Client Areas	40
Scrollbars	41
Caret and Cursor	41
Top-Level Windows	41
Coordinate System	41
Painting	41
Color and Font	42
Window Variant	42
Sizing	42
Input	42
Idle Time Processing and UI Updates	42
Window Creation and Deletion	43
Window Styles	44
A Quick Guide to the Window Classes	44
Base Window Classes	45
Top-Level Windows	45
Container Windows	45
Non-Static Controls	45
Static Controls	46
Menus	46
Control Bars	46
Base Window Classes	46
<i>wxWindow</i>	47
<i>wxWindow</i> Styles	47
<i>wxWindow</i> Events	49

<i>wxWindow</i> Member Functions	50
<i>wxControl</i>	54
<i>wxControlWithItems</i>	54
<i>wxControlWithItems</i> Member Functions	54
Top-Level Windows	55
<i>wxFrame</i>	55
<i>wxFrame</i> Styles	58
<i>wxFrame</i> Events	59
<i>wxFrame</i> Member Functions	60
Non-Rectangular Frames	61
Mini-Frames	63
<i>wxMDIParentFrame</i>	63
<i>wxMDIParentFrame</i> Styles	64
<i>wxMDIParentFrame</i> Member Functions	65
<i>wxMDIChildFrame</i>	65
<i>wxMDIChildFrame</i> Styles	65
<i>wxMDIChildFrame</i> Member Functions	66
<i>wxDialog</i>	66
<i>wxDialog</i> Styles	69
<i>wxDialog</i> Events	70
<i>wxDialog</i> Member Functions	71
<i>wxPopupWindow</i>	71
Container Windows	72
<i>wxPanel</i>	72
<i>wxPanel</i> Styles	72
<i>wxPanel</i> Member Functions	72
<i>wxNotebook</i>	72
Notebook Theme Management	74
<i>wxNotebook</i> Styles	74
<i>wxNotebook</i> Events	75
<i>wxNotebook</i> Member Functions	75
Alternatives to <i>wxNotebook</i>	76
<i>wxScrolledWindow</i>	76
<i>wxScrolledWindow</i> Styles	78
<i>wxScrolledWindow</i> Events	78
<i>wxScrolledWindow</i> Member Functions	79
Scrolling Without Using <i>wxScrolledWindow</i>	80
<i>wxSplitterWindow</i>	81
<i>wxSplitterWindow</i> Styles	83
<i>wxSplitterWindow</i> Events	83
<i>wxSplitterWindow</i> Member Functions	84
Sizing Issues with <i>wxSplitterWindow</i>	84
Alternatives to <i>wxSplitterWindow</i>	85
Non-Static Controls	86
<i>wxButton</i>	86

<i>wxBUTTON</i> Styles	86
<i>wxBUTTON</i> Events	87
<i>wxBUTTON</i> Member Functions	87
<i>wxBUTTON</i> Labels	87
<i>wxBitmapButton</i>	89
<i>wxBitmapButton</i> Styles	90
<i>wxBitmapButton</i> Events	90
<i>wxBitmapButton</i> Member Functions	90
<i>wxChoice</i>	90
<i>wxChoice</i> Styles	91
<i>wxChoice</i> Events	91
<i>wxChoice</i> Member Functions	91
<i>wxComboBox</i>	91
<i>wxComboBox</i> Styles	92
<i>wxComboBox</i> Events	93
<i>wxComboBox</i> Member Functions	93
<i>wxCheckBox</i>	93
<i>wxCheckBox</i> Styles	94
<i>wxCheckBox</i> Events	95
<i>wxCheckBox</i> Member Functions	95
<i>wxListBox</i> and <i>wxCheckListBox</i>	95
<i>wxListBox</i> and <i>wxCheckListBox</i> Styles	97
<i>wxListBox</i> and <i>wxCheckListBox</i> Events	97
<i>wxListBox</i> Member Functions	98
<i>wxCheckListBox</i> Member Functions	98
<i>wxRadioBox</i>	98
<i>wxRadioBox</i> Styles	99
<i>wxRadioBox</i> Events	100
<i>wxRadioBox</i> Member Functions	100
<i>wxRadioButton</i>	100
<i>wxRadioButton</i> Styles	101
<i>wxRadioButton</i> Events	101
<i>wxRadioButton</i> Member Functions	101
<i>wxScrollBar</i>	101
<i>wxScrollBar</i> Styles	102
<i>wxScrollBar</i> Events	102
<i>wxScrollBar</i> Member Functions	103
<i>wxSpinButton</i>	103
<i>wxSpinButton</i> Styles	104
<i>wxSpinButton</i> Events	104
<i>wxSpinButton</i> Member Functions	104
<i>wxSpinCtrl</i>	104
<i>wxSpinCtrl</i> Styles	105
<i>wxSpinCtrl</i> Events	105
<i>wxSpinCtrl</i> Member Functions	106
<i>wxSlider</i>	106

<i>wxSlider</i> Styles	106
<i>wxSlider</i> Events	107
<i>wxSlider</i> Member Functions	107
<i>wxTextCtrl</i>	107
<i>wxTextCtrl</i> Styles	109
<i>wxTextCtrl</i> Events	110
<i>wxTextCtrl</i> Member Functions	110
<i>wxToggleButton</i>	111
<i>wxToggleButton</i> Styles	112
<i>wxToggleButton</i> Events	112
<i>wxToggleButton</i> Member Functions	112
Static Controls	112
<i>wxGauge</i>	112
<i>wxGauge</i> Styles	113
<i>wxGauge</i> Events	113
<i>wxGauge</i> Member Functions	113
<i>wxStaticText</i>	113
<i>wxStaticText</i> Styles	114
<i>wxStaticText</i> Member Functions	114
<i>wxStaticBitmap</i>	114
<i>wxStaticBitmap</i> Styles	115
<i>wxStaticBitmap</i> Member Functions	115
<i>wxStaticLine</i>	115
<i>wxStaticLine</i> Styles	115
<i>wxStaticLine</i> Member Functions	116
<i>wxStaticBox</i>	116
<i>wxStaticBox</i> Styles	116
<i>wxStaticBox</i> Member Functions	116
Menus	116
<i>wxMenu</i>	117
<i>wxMenu</i> Events	118
<i>wxMenu</i> Member Functions	120
Control Bars	122
<i>wxMenuBar</i>	122
<i>wxMenuBar</i> Styles	123
<i>wxMenuBar</i> Events	123
<i>wxMenuBar</i> Member Functions	123
<i>wxToolBar</i>	124
Tool Bitmap Colors Under Windows	125
<i>wxToolBar</i> Styles	126
<i>wxToolBar</i> Events	126
<i>wxToolBar</i> Member Functions	127
<i>wxStatusBar</i>	128
<i>wxStatusBar</i> Styles	129
<i>wxStatusBar</i> Events	129
<i>wxStatusBar</i> Member Functions	129
Summary	130

5 Drawing and Printing	31
Understanding Device Contexts	131
Available Device Contexts	132
Drawing on Windows with <i>wxClientDC</i>	133
Erasing Window Backgrounds	134
Drawing on Windows with <i>wxPaintDC</i>	135
Drawing on Bitmaps with <i>wxMemoryDC</i>	138
Creating Metafiles with <i>wxMetafileDC</i>	138
Accessing the Screen with <i>wxScreenDC</i>	139
Printing with <i>wxPrinterDC</i> and <i>wxPostScriptDC</i>	139
Drawing Tools	140
<i>wxColour</i>	141
<i>wxPen</i>	142
<i>wxBrush</i>	143
<i>wxFont</i>	145
<i>wxPalette</i>	147
Device Context Drawing Functions	148
Drawing Text	150
Drawing Lines and Shapes	152
Drawing Splines	156
Drawing Bitmaps	157
Filling Arbitrary Areas	159
Logical Functions	159
Using the Printing Framework	161
More on <i>wxPrintout</i>	163
Scaling for Printing and Previewing	165
Printing under Unix with GTK+	167
3D Graphics with <i>wxGLCanvas</i>	168
Summary	170
6 Handling Input	171
Mouse Input	171
Handling Button and Motion Events	173
Handling Mouse Wheel Events	174
Handling Keyboard Events	175
An Example Character Event Handler	178
Key Code Translation	179
Modifier Key Variations	179
Accelerators	180
Handling Joystick Events	181
<i>wxJoystick</i> Events	183
<i>wxJoystickEvent</i> Member Functions	184
<i>wxJoystick</i> Member Functions	184
Summary	185

7 Window Layout Using Sizers	187
Layout Basics	187
Sizers	188
Common Features of Sizers	190
Programming with Sizers	192
Programming with <i>wxBoxSizer</i>	193
Programming with <i>wxStaticBoxSizer</i>	196
Programming with <i>wxGridSizer</i>	197
Programming with <i>wxFlexGridSizer</i>	198
Programming with <i>wxGridBagSizer</i>	200
Further Layout Issues	201
Dialog Units	202
Platform-Adaptive Layouts	202
Dynamic Layouts	204
Summary	204
8 Using Standard Dialogs	205
Informative Dialogs	205
<i>wxMessageDialog</i>	205
<i>wxMessageDialog</i> Example	207
<i>wxMessageBox</i>	207
<i>wxProgressDialog</i>	208
<i>wxProgressDialog</i> Example	209
<i>wxBusyInfo</i>	210
<i>wxBusyInfo</i> Example	210
<i>wxShowTip</i>	210
<i>wxShowTip</i> Example	211
File and Directory Dialogs	212
<i>wxFileDialog</i>	212
<i>wxFileDialog</i> Styles	214
<i>wxFileDialog</i> Functions	215
<i>wxFileDialog</i> Example	215
<i>wxDirDialog</i>	216
<i>wxDirDialog</i> Functions	218
<i>wxDirDialog</i> Example	218
Choice and Selection Dialogs	218
<i>wxColourDialog</i>	218
<i>wxColourData</i> Functions	221
<i>wxColourDialog</i> Example	221
<i>wxFontDialog</i>	221
<i>wxFontData</i> Functions	223
Font Selector Example	224
<i>wxSingleChoiceDialog</i>	224
<i>wxSingleChoiceDialog</i> Example	225

<i>wxMultiChoiceDialog</i>	225
<i>wxMultiChoiceDialog</i> Example	226
Entry Dialogs	227
<i>wxNumberEntryDialog</i>	227
<i>wxNumberEntryDialog</i> Example	227
<i>wxTextEntryDialog</i> and <i>wxPasswordEntryDialog</i>	227
<i>wxTextEntryDialog</i> Example	228
<i>wxFindReplaceDialog</i>	228
Handling Events from the Dialog	229
<i>wxFindDialogEvent</i> Functions	230
Passing Data to the Dialog	230
<i>wxFindReplaceData</i> Functions	230
Find and Replace Example	231
Printing Dialogs	232
<i>wxPageSetupDialog</i>	232
<i>wxPageSetupData</i> Functions	235
<i>wxPageSetupDialog</i> Example	235
<i>wxPrintDialog</i>	235
<i>wxPrintDialogData</i> Functions	238
<i>wxPrintDialog</i> Example	238
Summary	239
9 Writing Custom Dialogs	241
Steps in Creating a Custom Dialog	241
An Example: <i>PersonalRecordDialog</i>	242
Deriving a New Class	243
Designing Data Storage	243
Coding the Controls and Layout	244
Data Transfer and Validation	247
Handling Events	249
Handling UI Updates	250
Adding Help	251
Tooltips	251
Context-Sensitive Help	251
Online Help	252
The Complete Class	253
Invoking the Dialog	253
Adapting Dialogs for Small Devices	254
Further Considerations in Dialog Design	255
Keyboard Navigation	255
Data and UI Separation	256
Layout	256
Aesthetics	257
Alternatives to Dialogs	257

Using wxWidgets Resource Files	257
Loading Resources	258
Using Binary and Embedded Resource Files	259
Translating Resources	261
The XRC Format	261
Writing Resource Handlers	262
Foreign Controls	263
Summary	264
10 Programming with Images	265
Image Classes in wxWidgets	265
Programming with <i>wxBitmap</i>	267
Creating a <i>wxBitmap</i>	268
Setting a <i>wxMask</i>	269
The XPM Format	270
Drawing with Bitmaps	271
Packaging Bitmap Resources	272
Programming with <i>wxIcon</i>	272
Creating a <i>wxIcon</i>	273
Using <i>wxIcon</i>	274
Associating an Icon with an Application	274
Programming with <i>wxCursor</i>	275
Creating a <i>wxCursor</i>	276
Using <i>wxCursor</i>	278
Using <i>wxSetCursorEvent</i>	278
Programming with <i>wxImage</i>	279
Loading and Saving Images	280
Transparency	282
Transformations	284
Color Reduction	284
Manipulating <i>wxImage</i> Data Directly	285
Image Lists and Icon Bundles	285
Customizing Art in wxWidgets	288
Summary	290
11 Clipboard and Drag and Drop	291
Data Objects	291
Data Source Duties	292
Data Target Duties	292
Using the Clipboard	293
Implementing Drag and Drop	294
Implementing a Drag Source	294
1. Preparation	295
2. Drag Start	295

3. Dragging	295
4. Processing the Result	296
Implementing a Drop Target	297
1. Initialization	297
2. Dragging	298
3. Drop	298
Using Standard Drop Targets	298
Creating a Custom Drop Target	299
More on <i>wxDropTarget</i>	300
Deriving from <i>wxDropTarget</i>	301
The <i>wxDropTarget</i> Drag and Drop Sample	302
Drag and Drop Helpers in <i>wxDropTarget</i>	311
<i>wxDropTarget</i>	311
<i>wxDropTargetList</i>	312
<i>wxDropTargetImage</i>	313
Summary	316
12 Advanced Window Classes	317
<i>wxDropTarget</i>	317
<i>wxDropTarget</i> Styles	319
<i>wxDropTarget</i> Events	320
<i>wxDropTarget</i> Member Functions	321
<i>wxDropTargetList</i>	322
<i>wxDropTargetList</i> Styles	323
<i>wxDropTargetList</i> Events	324
<i>wxDropTargetItem</i>	325
<i>wxDropTargetList</i> Member Functions	327
Using <i>wxDropTargetList</i>	329
Virtual List Controls	330
<i>wxDropTargetWizard</i>	331
<i>wxDropTargetWizard</i> Events	331
<i>wxDropTargetWizard</i> Member Functions	332
<i>wxDropTargetWizard</i> Example	332
<i>wxDropTargetHtmlWindow</i>	338
<i>wxDropTargetHtmlWindow</i> Styles	341
<i>wxDropTargetHtmlWindow</i> Member Functions	342
Embedding Windows in HTML Pages	343
HTML Printing	344
<i>wxDropTargetGrid</i>	346
The <i>wxDropTargetGrid</i> System of Classes	348
<i>wxDropTargetGrid</i> Events	349
<i>wxDropTargetGrid</i> Member Functions	350
Functions for Creation, Deletion, and Data Retrieval	350
Presentation Functions	351
Functions for Setting and Getting <i>wxDropTargetGrid</i> Metrics	352

Selection and Cursor Functions	352
Miscellaneous <i>wxGrid</i> Functions	353
<i>wxTaskBarIcon</i>	354
<i>wxTaskBarIcon</i> Events	358
<i>wxTaskBarIcon</i> Member Functions	358
Writing Your Own Controls	359
The Custom Control Declaration	360
Adding <i>DoGetBestSize</i>	362
Defining a New Event Class	363
Displaying Information on the Control	363
Handling Input	364
Defining Default Event Handlers	364
Implementing Validators	365
Implementing Resource Handlers	367
Determining Control Appearance	367
A More Complex Example: <i>wxThumbnailCtrl</i>	369
Summary	371
13 Data Structure Classes	373
Why Not STL?	373
Strings	374
Using <i>wxString</i>	374
<i>wxString</i> , Characters, and String Literals	375
Basic <i>wxString</i> to C Pointer Conversions	375
Standard C String Functions	376
Converting to and from Numbers	377
<i>wxStringTokenizer</i>	378
<i>wxRegEx</i>	379
<i>wxArray</i>	379
Array Types	379
<i>wxArrayString</i>	380
Array Construction, Destruction, and Memory Management	381
Array Sample Code	381
<i>wxList</i> and <i>wxNode</i>	383
<i>wxHashMap</i>	385
Storing and Processing Dates and Times	387
<i>wxDateTime</i>	388
<i>wxDateTime</i> Constructors and Modifiers	388
<i>wxDateTime</i> Accessors	388
Getting the Current Time	389
Parsing and Formatting Dates	389
Date Comparisons	389
Date Arithmetic	389
Helper Data Structures	391
<i>wxObject</i>	391

<i>wxLongLong</i>	392
<i>wxPoint</i> and <i>wxRealPoint</i>	392
<i>wxRect</i>	392
<i>wxRegion</i>	393
<i>wxSize</i>	394
<i>wxVariant</i>	394
Summary	395
14 Files and Streams	397
File Classes and Functions	397
<i>wxFile</i> and <i>wxFFile</i>	397
<i>wxTextFile</i>	399
<i>wxTempFile</i>	400
<i>wxDir</i>	400
<i>wxFileName</i>	401
File Functions	402
Stream Classes	403
File Streams	403
Memory and String Streams	405
Reading and Writing Data Types	405
Socket Streams	406
Filter Streams	407
Zip Streams	407
Virtual File Systems	408
Summary	411
15 Memory Management, Debugging, and Error Checking	413
Memory Management Basics	413
Creating and Deleting Window Objects	413
Creating and Copying Drawing Objects	415
Initializing Your Application Object	415
Cleaning Up Your Application	416
Detecting Memory Leaks and Other Errors	416
Facilities for Defensive Programming	418
Error Reporting	419
<i>wxMessageOutput</i> Versus <i>wxLog</i>	423
Providing Run-Time Type Information	424
Using <i>wxModule</i>	426
Loading Dynamic Libraries	427
Exception Handling	428
Debugging Tips	429
Debugging X11 Errors	429
Simplify the Problem	430
Debugging a Release Build	430
Summary	431

16 Writing International Applications	433
Introduction to Internationalization	433
Providing Translations	434
poEdit	434
Step-by-Step Guide to Using Message Catalogs	435
Using <i>wxLocale</i>	437
Character Encodings and Unicode	438
Converting Data	439
<i>wxEncodingConverter</i>	440
<i>wxCSSConv (wxMBConv)</i>	440
Converting Outside of a Temporary Buffer	441
Help Files	442
Numbers and Dates	443
Other Media	443
A Simple Sample	444
Summary	446
17 Writing Multithreaded Applications	447
When to Use Threads, and When Not To	447
Using <i>wxThread</i>	448
Creation	449
Specifying Stack Size	450
Specifying Priority	450
Starting the Thread	450
How to Pause a Thread or Wait for an External Condition	450
Termination	451
Synchronization Objects	451
<i>wxMutex</i>	451
Deadlocks	452
<i>wxCriticalSection</i>	453
<i>wxCondition</i>	453
<i>wxCondition</i> Example	453
<i>wxSemaphore</i>	456
The wxWidgets Thread Sample	456
Alternatives to Multithreading	457
Using <i>wxTimer</i>	457
Idle Time Processing	459
Yielding	460
Summary	460
18 Programming with <i>wxSocket</i>	463
Socket Classes and Functionality Overview	464
Introduction to Sockets and Basic Socket Processing	464
The Client	465
The Server	466

Connecting to a Server	467
Socket Addresses	468
Socket Clients	468
Socket Events	469
Socket Event Types	469
<i>wxSocketEvent</i> Major Member Functions	469
Using Socket Events	470
Socket Status and Error Notifications	470
Sending and Receiving Socket Data	471
Reading	471
Writing	472
Creating a Server	472
<i>wxSocketServer</i> Major Member Functions	472
Handling a New Connection Event	472
Socket Event Recap	473
Socket Flags	473
Blocking and Non-Blocking Sockets in <i>wxWidgets</i>	474
The Impossible Socket Combination	475
How Flags Affect Socket Behavior	475
Using <i>wxSocket</i> as a Standard Socket	476
Using Socket Streams	476
File Sending Thread	477
File Receiving Thread	478
Alternatives to <i>wxSocket</i>	479
Summary	480
19 Working with Documents and Views	481
Document/View Basics	481
Step 1: Choose an Interface Style	483
Step 2: Create and Use Frame Classes	483
Step 3: Define Your Document and View Classes	486
Step 4: Define Your Window Classes	495
Step 5: Use <i>wxDocManager</i> and <i>wxDocTemplate</i>	497
Other Document/View Capabilities	500
Standard Identifiers	500
Printing and Previewing	500
File History	500
Explicit Document Creation	501
Strategies for Implementing Undo/Redo	501
Summary	503
20 Perfecting Your Application	505
Single Instance or Multiple Instances?	506
Modifying Event Handling	510
Reducing Flicker	512