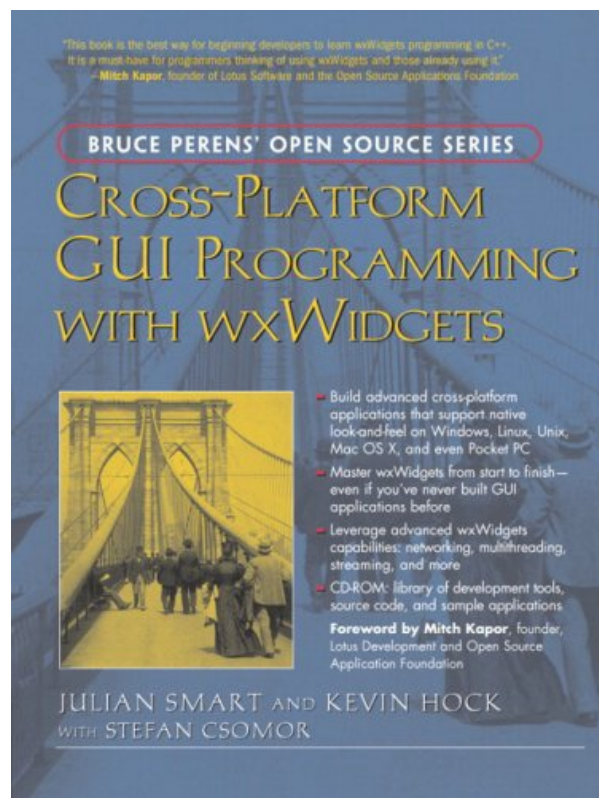


CROSS-PLATFORM GUI PROGRAMMING WITH WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR



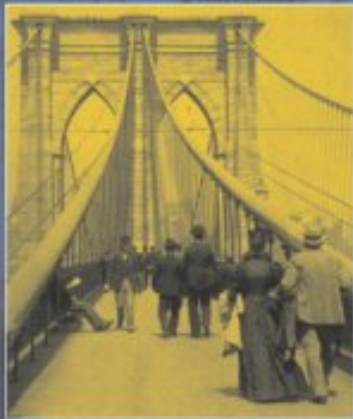
**DOWNLOAD EBOOK : CROSS-PLATFORM GUI PROGRAMMING WITH
WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR
PDF**



"This book is the best way for beginning developers to learn wxWidgets programming in C++.
It is a must-have for programmers thinking of using wxWidgets and those already using it."
— **Mitch Kapor**, founder of Lotus Software and the Open Source Applications Foundation

BRUCE PERENS' OPEN SOURCE SERIES

CROSS-PLATFORM GUI PROGRAMMING WITH wxWIDGETS



- Build advanced cross-platform applications that support native look-and-feel on Windows, Linux, Unix, Mac OS X, and even Pocket PC
 - Master wxWidgets from start to finish— even if you've never built GUI applications before
 - Leverage advanced wxWidgets capabilities: networking, multithreading, streaming, and more
 - CD-ROM: library of development tools, source code, and sample applications
- Foreword by Mitch Kapor**, founder, Lotus Development and Open Source Application Foundation

JULIAN SMART AND KEVIN HOCK
WITH STEFAN CSOMOR

Click link bellow and free register to download ebook:

CROSS-PLATFORM GUI PROGRAMMING WITH WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR

[DOWNLOAD FROM OUR ONLINE LIBRARY](#)

CROSS-PLATFORM GUI PROGRAMMING WITH WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR PDF

When you are hurried of task target date and also have no idea to obtain inspiration, **Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor** publication is one of your solutions to take. Schedule Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor will certainly give you the appropriate source as well as point to obtain inspirations. It is not only concerning the jobs for politic business, management, economics, and various other. Some bought works making some fiction jobs also need motivations to conquer the task. As just what you require, this Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor will possibly be your selection.

From the Back Cover

"This book is the best way for beginning developers to learn wxWidgets programming in C++. It is a must-have for programmers thinking of using wxWidgets and those already using it."

—Mitch Kapor, founder of Lotus Software and the Open Source Applications Foundation

- Build advanced cross-platform applications that support native look-and-feel on Windows, Linux, Unix, Mac OS X, and even Pocket PC
- Master wxWidgets from start to finish—even if you've never built GUI applications before
- Leverage advanced wxWidgets capabilities: networking, multithreading, streaming, and more
- CD-ROM: library of development tools, source code, and sample applications
- Foreword by Mitch Kapor, founder, Lotus Development and Open Source Application Foundation

wxWidgets is an easy-to-use, open source C++ API for writing GUI applications that run on Windows, Linux, Unix, Mac OS X, and even Pocket PC—supporting each platform's native look and feel with virtually no additional coding. Now, its creator and two leading developers teach you all you need to know to write robust cross-platform software with wxWidgets. This book covers everything from dialog boxes to drag-and-drop, from networking to multithreading. It includes all the tools and code you need to get great results, fast. From AMD to AOL, Lockheed Martin to Xerox, world-class developers are using wxWidgets to save money, increase efficiency, and reach new markets. With this book, you can, too.

- wxWidgets quickstart: event/input handling, window layouts, drawing, printing, dialogs, and more
- Working with window classes, from simple to advanced
- Memory management, debugging, error checking, internationalization, and other advanced topics
- Includes extensive code samples for Windows, Linux (GTK+), and Mac OS X

About the CD-ROM

The CD-ROM contains all of the source code from the book; wxWidgets distributions for Windows, Linux, Unix, Mac OS X, and other platforms; the wxWidgets reference guide; and development tools including the OpenWatcom C++ compiler, the poEdit translation helper, and the DialogBlocks user interface builder.

© Copyright Pearson Education. All rights reserved.

About the Author

Julian Smart has degrees from the University of St. Andrews and the University of Dundee. After working on model-based reasoning at the Scottish Crop Research Institute, he moved to the Artificial Intelligence Applications Institute at the University of Edinburgh, where he founded the wxWidgets project in 1992. Since starting Anthemion Software in 1996, Julian has been helping other companies deploy wxWidgets, and he sells tools for programmers, including DialogBlocks and HelpBlocks. He has worked as a consultant for various companies including Borland and was a member of Red Hat's eCos team, writing GUI tools to support the embedded operating system. In 2004, Julian and his wife Harriet launched a consumer product for fiction writers called Writer's Café, written with wxWidgets. Julian and Harriet live in Edinburgh with their daughter Toni.

Kevin Hock has degrees from Miami University (Oxford, Ohio) in Computer Science and Accounting and has taught courses at Miami in both Java and client-server systems. In 2002, he started work on an instant messaging system and founded BitWise Communications, LLC, in 2003, offering both professional and personal instant messaging. During the course of developing BitWise using wxWidgets, Kevin became a wxWidgets developer and has provided enhancements to all platforms. Kevin lives in Oxford, Ohio.

Stefan Csomor is director and owner of Advanced Concepts AG, a company that specializes in cross-platform development and consulting. In addition to being a qualified medical doctor, he has more than 15 years of experience in object-oriented programming and has been writing software for 25 years. Stefan is the main author of the Mac OS port of wxWidgets.

© Copyright Pearson Education. All rights reserved.

Excerpt. © Reprinted by permission. All rights reserved.

Preface Preface Who This Book Is For

This book is a guide to using wxWidgets, an open-source construction kit for writing sophisticated C++ applications targeting a variety of platforms, including Windows, Linux, Mac OS X, and Pocket PC. With help from this book, a competent programmer can create multi-platform applications with confidence. Developers already familiar with wxWidgets should also find it useful for brushing up their knowledge.

This book is accessible to developers with a variety of experience and backgrounds. You may come from a Windows or Unix perspective; you may previously have experience in MFC, OWL, Win32, Mac OS, Motif, or console-mode Unix programming. Or perhaps you have come from a different career entirely and are looking for a way to get up to speed on multiple platforms. The book can't specifically cover the details of the C++ language, but it's common for people to successfully learn C++ and wxWidgets at the same time, and the straightforward nature of the wxWidgets API makes this process easier. The reader does not need to

know more advanced C++ techniques like templates, streams, and exceptions. However, wxWidgets does not prevent you from using these techniques.

Managers will find the book useful in discovering what wxWidgets can do for them, particularly in Chapter 1, "Introduction." The combination of the book and the resources on the accompanying CD-ROM will give your staff all they need for getting started on cross-platform programming projects. You'll see how wxWidgets puts tools of tremendous power into your hands, with benefits that include:

- Cost savings from writing code once that will compile on Windows, Unix, Mac OS X, and other platforms.
- Customer satisfaction from delivering stable, fast, attractive applications with a native look and feel.
- Increased productivity from the wide variety of classes that wxWidgets provides, both for creating great GUIs and for general application development.
- Increased market share due to support for platforms you may not have previously considered, and the ability to internationalize your applications.
- Support from a large, active wxWidgets community that answers questions helpfully and provides prompt bug fixing. The sample of third-party add-ons listed in Appendix E, "Third-Party Tools for wxWidgets," is evidence of a thriving ecosystem.
- Access to the source for enhancement and trouble-shooting.

This is a guide to writing wxWidgets application with C++, but you can use a variety of other languages such as Python, Perl, a BASIC variant, Lua, Eiffel, JavaScript, Java, Ruby, Haskell, and C#. Some of these bindings are more advanced than others. For more information, please see Appendix E and the wxWidgets web site at <http://www.wxwidgets.org>.

We focus on three popular desktop platforms: Microsoft Windows, Linux using GTK+, and Mac OS X. However, most of the book also applies to other platforms supported by wxWidgets. In particular, wxWidgets can be used with most Unix variants.

The CD-ROM

The CD-ROM contains example code from the book, the wxWidgets 2.6 distribution for Windows, Linux, Mac OS X, and other platforms, and several tools to help you use wxWidgets, including the translation tool poEdit. For Windows users, we supply three free compilers you can use with wxWidgets: MinGW, Digital Mars C++, and OpenWatcom C++.

In addition, we provide you with DialogBlocks Personal Edition, a sophisticated rapid application development (RAD) tool for you to create complex windows with very little manual coding. You can use it to compile and run samples that accompany the book as well as to create your own applications for personal use, and it also provides convenient access to the wxWidgets reference manual.

Updates to the book and CD-ROM can be obtained from this site:

<http://www.wxwidgets.org/book>

How to Use This Book

It's advisable to read at least Chapters 1 through 10 in order, but you can skip to other chapters if you need to complete a particular task. If you haven't installed wxWidgets before, you may want to look at Appendix A, "Installing wxWidgets," early on. MFC programmers will find it useful to read Appendix K, "Porting from MFC," as a point of reference.

Because this book is not a complete API reference, you'll find it useful to keep the wxWidgets reference manual open. The reference manual is available in a number of formats, including Windows HTML Help and PDF, and it should be in your wxWidgets distribution; if not, it can be downloaded from the wxWidgets web site. You can also refer to the many samples in the wxWidgets distribution to supplement the examples given in this book.

Note that the book is intended to be used in conjunction with wxWidgets 2.6 or later. The majority of the book will apply to earlier versions, but be aware that some functionality will be missing, and in a small number of cases, the behavior may be different. In particular, sizer behavior changed somewhat between 2.4 and 2.5. For details, please see the topic "Changes Since 2.4.x" in the wxWidgets reference manual.

Conventions

For code examples, we mostly follow the wxWidgets style guidelines, for example:

- Words within class names and functions have an initial capital, for example `MyFunkyClass`.
- The `m_` prefix denotes a member variable, `s_` denotes a static variable, `g_` denotes a global variable; local variables generally start with a lowercase letter, for example `textCtrl`.

You can find more about the wxWidgets style guidelines at <http://www.wxwidgets.org/standard.htm>.

Sometimes we'll also use comments that can be parsed by the documentation tool Doxygen, such as:

```
/// A function description
```

Classes, functions, identifiers, variables, and standard wxWidgets objects are marked with a teletype font in the text. User interface commands, such as menu and button labels, are marked in italics.

Chapter Summary Chapter 1: Introduction

What is wxWidgets, and why use it? A brief history; the wxWidgets community; the license; wxWidgets ports and architecture explained.

Chapter 2: Getting Started

A small wxWidgets sample: the application class; the main window; the event table; an outline of program flow.

Chapter 3: Event Handling

Event tables and handlers; how a button click is processed; skipping events; pluggable and dynamic event handlers; defining custom events; window identifiers.

Chapter 4: Window Basics

The main features of a window explained; a quick guide to the commonest window classes; base window classes such as `wxWindow`; top-level windows; container windows; non-static controls; static controls; menus; control bars.

Chapter 5: Drawing and Printing

Device context principles; the main device context classes described; buffered drawing; drawing tools;

device context drawing functions; using the printing framework; 3D graphics with wxGLCanvas.

Chapter 6: Handling Input

Handling mouse and mouse wheel events; handling keyboard events; keycodes; modifier key variations; accelerators; handling joystick events.

Chapter 7: Window Layout Using Sizers

Layout basics; sizers introduced; common features of sizers; programming with sizers. Further layout issues: dialog units; platform-adaptive layouts; dynamic layouts.

Chapter 8: Using Standard Dialogs

Informative dialogs such as wxMessageBox and wxProgressDialog; file and directory dialogs such as wxFileDialog; choice and selection dialogs such as wxColourDialog and wxFontDialog; entry dialogs such as wxTextEntryDialog and wxFindReplaceDialog; printing dialogs: wxPageSetupDialog and wxPrintDialog.

Chapter 9: Writing Custom Dialogs

Steps in creating a custom dialog; an example: PersonalRecordDialog; deriving a new class; designing data storage; coding the controls and layout; data transfer and validation; handling events; handling UI updates; adding help; adapting dialogs for small devices; further considerations in dialog design; using wxWidgets resource files; loading resources; using binary and embedded resource files; translating resources; the XRC format; writing resource handlers; foreign controls.

Chapter 10: Programming with Images

Image classes in wxWidgets; programming with wxBitmap; programming with wxIcon; programming with wxCursor; programming with wxImage; image lists and icon bundles; customizing wxWidgets graphics with wxArtProvider.

Chapter 11: Clipboard and Drag and Drop

Data objects; data source duties; data target duties; using the clipboard; implementing drag and drop; implementing a drag source; implementing a drop target; using standard drop targets; creating a custom drop target; more on wxDataObject; drag and drop helpers in wxWidgets.

Chapter 12: Advanced Window Classes

wxTreeCtrl; wxListCtrl; wxWizard; wxHtmlWindow; wxGrid; wxTaskBarIcon; writing your own controls; the control declaration; defining a new event class; displaying information; handling input; defining default event handlers; implementing validators; implementing resource handlers; determining control appearance.

Chapter 13: Data Structure Classes

Why not STL? wxString; wxStringTokenizer; wxRegex; wxArray; wxList; wxHashMap; wxDateTime; wxObject; wxLongLong; wxPoint and wxRealPoint; wxRect; wxRegion; wxSize; wxVariant.

Chapter 14: Files and Streams

wxFile and wxFFile; wxTextFile; wxTempFile; wxDir; wxFileName; file functions; file streams; memory

and string streams; data streams; socket streams; filter streams; zip streams; virtual file systems.

Chapter 15: Memory Management, Debugging, and Error Checking

Creating and deleting window objects; creating and copying drawing objects; initializing your application object; cleaning up your application; detecting memory leaks and other errors; facilities for defensive programming; error reporting; providing run-time type information; using wxModule; loading dynamic libraries; exception handling; debugging tips.

Chapter 16: Writing International Applications

Introduction to internationalization; providing translations; using message catalogs; using wxLocale; character encodings and Unicode; converting data; help files; numbers and dates; other media; an example.

Chapter 17: Writing Multithreaded Applications

When to use threads, and when not to; using wxThread; thread creation; starting the thread; how to pause a thread or wait for an external condition; termination; synchronization objects; wxMutex; deadlocks; wxCriticalSection; wxCondition; wxSemaphore; the wxWidgets thread sample; alternatives to multithreading: wxTimer, idle time processing, and yielding.

Chapter 18: Programming with wxSocket

Socket classes and functionality overview; introduction to sockets and basic socket processing; connecting to a server; socket events; socket status and error notifications; sending and receiving socket data; creating a server; socket event recap; socket flags; blocking and non-blocking sockets in wxWidgets; how flags affect socket behavior; using wxSocket as a standard socket; using socket streams; alternatives to wxSocket.

Chapter 19: Working with Documents and Views

Document/view basics; choosing an interface style; creating and using frame classes; defining your document and view classes; defining your window classes; using wxDocManager and wxDocTemplate; other document/view capabilities; standard identifiers; printing and previewing; file history; explicit document creation; strategies for implementing undo/redo.

Chapter 20: Perfecting Your Application

Single instance versus multiple instances; modifying event handling; reducing flicker; using a help controller; extended wxWidgets HTML help; authoring help; other ways to provide help; parsing the command line; storing application resources; invoking other applications; launching documents; redirecting process input and output; managing application settings; application installation on Windows, Linux and Mac OS X; following UI design guidelines.

Appendix A: Installing wxWidgets

Downloading and unpacking wxWidgets; configuration/build options; Windows—Microsoft Visual Studio and VC++ command-line; Windows—Borland C++; Windows—MinGW with and without MSYS; Unix/Linux and Mac OS X—GCC; customizing setup.h; rebuilding after updating wxWidgets files; using contrib libraries.

Appendix B: Building Your Own wxWidgets Applications

Windows—Microsoft Visual Studio; Linux—KDevelop; Mac OS X—Xcode; makefiles; cross-platform builds using Bakefile; wxWidgets symbols and headers; using wx-config.

Appendix C: Creating Applications with DialogBlocks

What is DialogBlocks? Installing and upgrading DialogBlocks; the DialogBlocks interface; the sample project; compiling the sample; creating a new project; creating a dialog; creating a frame; creating an application object; debugging your application.

Appendix D: Other Features in wxWidgets

Further window classes; ODBC classes; MIME types manager; network functionality; multimedia classes; embedded web browsers; accessibility; OLE automation; renderer classes; event loops.

Appendix E: Third-Party Tools for wxWidgets

Language bindings such as wxPython and wxPerl; tools such as wxDesigner, DialogBlocks and poEdit; add-on libraries such as wxMozilla, wxCURL, wxPropertyGrid.

Appendix F: wxWidgets Application Showcase

Descriptions of notable wxWidgets applications, such as AOL Communicator and Audacity.

Appendix G: Using the CD-ROM

Browsing the CD-ROM; the CD-ROM contents.

Appendix H: How wxWidgets Processes Events

An illustrated description of how event processing works.

Appendix I: Event Classes and Macros

A summary of the important event classes and macros.

Appendix J: Code Listings

Code listings for the PersonalRecordDialog and the wxWizard examples.

Appendix K: Porting from MFC

General observations; application initialization; message maps; converting dialogs and other resources; documents and views; printing; string handling and translation; database access; configurable control bars; equivalent functionality by macros and classes.

Acknowledgments

wxWidgets owes its success to the hard work of many talented people. We would like to thank them all, with special consideration for that essential support network: our long-suffering families and partners. wxWidgets supporters and contributors include the following (apologies for any unintentional omissions):

Yiorgos Adamopoulos, Jamshid Afshar, Alejandro Aguilar-Sierra, Patrick Albert, Bruneau Babet, Mitchell Baker, Mattia Barbon, Nerijus Baliunas, Karsten Ballueder, Jonathan Bayer, Michael Bedward, Kai Bendorf,

Yura Bidus, Jorgen Bodde, Borland, Keith Gary Boyce, Chris Breeze, Sylvain Bougnoux, Wade Brainerd, Pete Britton, Ian Brown, C. Buckley, Doug Card, Marco Cavallini, Dmitri Chubraev, Robin Corbet, Cecil Coupe, Stefan Csomor, Andrew Davison, Gilles Depeyrot, Duane Doran, Neil Dudman, Robin Dunn, Hermann Dunkel, Jos van Eijndhoven, Chris Elliott, David Elliott, David Falkinder, Rob Farnum, Joel Farley, Tom Felici, Thomas Fettig, Matthew Flatt, Pasquale Foggia, Josep Fortiana, Todd Fries, Dominic Gallagher, Roger Gammans, Guillermo Rodriguez Garcia, Brian Gavin, Wolfram Gloger, Aleksandras Gluchovas, Markus Greither, Norbert Grotz, Stephane Gully, Stefan Gunter, Bill Hale, Patrick Halke, Stefan Hammes, Guillaume Helle, Harco de Hilster, Kevin Hock, Cord Hockemeyer, Klaas Holwerda, Markus Holzem, Ove Kaaven, Mitch Kapor, Matt Kimball, Hajo Kirchoff, Olaf Klein, Jacob Jansen, Leif Jensen, Mark Johnson, Bart Jourquin, John Labenski, Guilhem Lavaux, Ron Lee, Hans Van Leemputten, Peter Lenhard, Jan Lessner, Nicholas Liebmann, Torsten Liermann, Per Lindqvist, Jesse Lovelace, Tatu Männistö, Lindsay Mathieson, Scott Maxwell, Bob Mitchell, Thomas Myers, Oliver Niedung, Stefan Neis, Ryan Norton, Robert O'Connor, Jeffrey Ollie, Kevin Ollivier, William Osborne, Hernan Otero, Ian Perrigo, Timothy Peters, Giordano Pezzoli, Harri Pasanen, Thomaso Paoletti, Garrett Potts, Robert Rae, Marcel Rasche, Mart Raudsepp, Andy Robinson, Robert Roebing, Alec Ross, Gunnar Roth, Thomas Runge, Tom Ryan, Dino Scaringella, Jobst Schmalenbach, Dimitri Schoolwerth, Arthur Seaton, Paul Shirley, Wlodzimierz Skiba, John Skiff, Vaclav Slavik, Brian Smith, Neil Smith, Stein Somers, Petr Smilauer, Kari Systä, George Tasker, Austin Tate, Arthur Tetzlaff-Deas, Paul Thiessen, Jonathan Tonberg, Jyrki Tuomi, Janos Vegh, Andrea Venturoli, David Webster, Michael Wetherell, Otto Wyss, Vadim Zeitlin, Xiaokun Zhu, Zbigniew Zagórski, Edward Zimmermann. Thanks also to Dotsrc.org and SourceForge for hosting project services.

Thanks are due in particular to Vadim Zeitlin, Vaclav Slavik, Robert Roebing, Stefan Csomor, and Robin Dunn for permission to adapt some of their contributions to the wxWidgets reference manual.

Special thanks go to Stefan Csomor who contributed Chapter 16 and Chapter 17, and to Kevin Ollivier who wrote the Bakefile tutorial in Appendix B. We would also like to thank Mitch Kapor for writing the foreword.

We are very grateful to Mark Taub for his patience and advice throughout. A big thank you goes to Marita Allwood, Harriet Smart, Antonia Smart, Clayton Hock, and Ethel Hock for all their love, support, and encouragement. A debt is also owed to all those who have reviewed and suggested improvements to the book, including: Stefan Csomor, Dimitri Schoolwerth, Robin Dunn, Carl Godkin, Bob Paddock, Chris Elliott, Michalis Kabrianis, Marc-Andre Lureau, Jonas Karlsson, Arnout Engelen, Erik van der Wal, Greg Smith, and Alexander Stigsen.

Finally, we hope that you enjoy reading this book and, most importantly, have fun using wxWidgets to build great-looking, multi-platform applications!

Julian Smart and Kevin Hock
June 2005

© Copyright Pearson Education. All rights reserved.

CROSS-PLATFORM GUI PROGRAMMING WITH WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR PDF

[Download: CROSS-PLATFORM GUI PROGRAMMING WITH WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR PDF](#)

This is it guide **Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor** to be best seller just recently. We give you the most effective offer by getting the amazing book *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor* in this web site. This *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor* will certainly not only be the type of book that is challenging to discover. In this internet site, all sorts of books are given. You can search title by title, writer by author, and publisher by author to discover the best book *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor* that you can read now.

By reading *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor*, you can understand the knowledge as well as points even more, not only regarding what you get from individuals to individuals. Reserve *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor* will be more trusted. As this *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor*, it will really give you the smart idea to be successful. It is not just for you to be success in specific life; you can be effective in everything. The success can be begun by recognizing the standard knowledge as well as do activities.

From the combination of understanding as well as actions, somebody could improve their ability and also ability. It will lead them to live and also function far better. This is why, the students, workers, or even employers ought to have reading behavior for publications. Any sort of book *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor* will offer particular expertise to take all perks. This is what this *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor* informs you. It will add more understanding of you to life as well as work far better. *Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor*, Try it as well as prove it.

CROSS-PLATFORM GUI PROGRAMMING WITH WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR PDF

"This book is the best way for beginning developers to learn wxWidgets programming in C++. It is a must-have for programmers thinking of using wxWidgets and those already using it."

–Mitch Kapor, founder of Lotus Software and the Open Source Applications Foundation

- Build advanced cross-platform applications that support native look-and-feel on Windows, Linux, Unix, Mac OS X, and even Pocket PC
- Master wxWidgets from start to finish—even if you've never built GUI applications before
- Leverage advanced wxWidgets capabilities: networking, multithreading, streaming, and more
- CD-ROM: library of development tools, source code, and sample applications
- Foreword by Mitch Kapor, founder, Lotus Development and Open Source Application Foundation

wxWidgets is an easy-to-use, open source C++ API for writing GUI applications that run on Windows, Linux, Unix, Mac OS X, and even Pocket PC—supporting each platform's native look and feel with virtually no additional coding. Now, its creator and two leading developers teach you all you need to know to write robust cross-platform software with wxWidgets. This book covers everything from dialog boxes to drag-and-drop, from networking to multithreading. It includes all the tools and code you need to get great results, fast. From AMD to AOL, Lockheed Martin to Xerox, world-class developers are using wxWidgets to save money, increase efficiency, and reach new markets. With this book, you can, too.

- wxWidgets quickstart: event/input handling, window layouts, drawing, printing, dialogs, and more
- Working with window classes, from simple to advanced
- Memory management, debugging, error checking, internationalization, and other advanced topics
- Includes extensive code samples for Windows, Linux (GTK+), and Mac OS X

About the CD-ROM

The CD-ROM contains all of the source code from the book; wxWidgets distributions for Windows, Linux, Unix, Mac OS X, and other platforms; the wxWidgets reference guide; and development tools including the OpenWatcom C++ compiler, the poEdit translation helper, and the DialogBlocks user interface builder.

© Copyright Pearson Education. All rights reserved.

- Sales Rank: #457876 in Books
- Published on: 2005-08-05
- Original language: English
- Number of items: 1
- Dimensions: 9.06" h x 1.50" w x 6.77" l, 2.36 pounds
- Binding: Paperback
- 744 pages

From the Back Cover

"This book is the best way for beginning developers to learn wxWidgets programming in C++. It is a must-have for programmers thinking of using wxWidgets and those already using it."

—Mitch Kapor, founder of Lotus Software and the Open Source Applications Foundation

- Build advanced cross-platform applications that support native look-and-feel on Windows, Linux, Unix, Mac OS X, and even Pocket PC
- Master wxWidgets from start to finish—even if you've never built GUI applications before
- Leverage advanced wxWidgets capabilities: networking, multithreading, streaming, and more
- CD-ROM: library of development tools, source code, and sample applications
- Foreword by Mitch Kapor, founder, Lotus Development and Open Source Application Foundation

wxWidgets is an easy-to-use, open source C++ API for writing GUI applications that run on Windows, Linux, Unix, Mac OS X, and even Pocket PC—supporting each platform's native look and feel with virtually no additional coding. Now, its creator and two leading developers teach you all you need to know to write robust cross-platform software with wxWidgets. This book covers everything from dialog boxes to drag-and-drop, from networking to multithreading. It includes all the tools and code you need to get great results, fast. From AMD to AOL, Lockheed Martin to Xerox, world-class developers are using wxWidgets to save money, increase efficiency, and reach new markets. With this book, you can, too.

- wxWidgets quickstart: event/input handling, window layouts, drawing, printing, dialogs, and more
- Working with window classes, from simple to advanced
- Memory management, debugging, error checking, internationalization, and other advanced topics
- Includes extensive code samples for Windows, Linux (GTK+), and Mac OS X

About the CD-ROM

The CD-ROM contains all of the source code from the book; wxWidgets distributions for Windows, Linux, Unix, Mac OS X, and other platforms; the wxWidgets reference guide; and development tools including the OpenWatcom C++ compiler, the poEdit translation helper, and the DialogBlocks user interface builder.

© Copyright Pearson Education. All rights reserved.

About the Author

Julian Smart has degrees from the University of St. Andrews and the University of Dundee. After working on model-based reasoning at the Scottish Crop Research Institute, he moved to the Artificial Intelligence Applications Institute at the University of Edinburgh, where he founded the wxWidgets project in 1992. Since starting Anthemion Software in 1996, Julian has been helping other companies deploy wxWidgets, and he sells tools for programmers, including DialogBlocks and HelpBlocks. He has worked as a consultant for various companies including Borland and was a member of Red Hat's eCos team, writing GUI tools to support the embedded operating system. In 2004, Julian and his wife Harriet launched a consumer product for fiction writers called Writer's Café, written with wxWidgets. Julian and Harriet live in Edinburgh with their daughter Toni.

Kevin Hock has degrees from Miami University (Oxford, Ohio) in Computer Science and Accounting and has taught courses at Miami in both Java and client-server systems. In 2002, he started work on an instant

messaging system and founded BitWise Communications, LLC, in 2003, offering both professional and personal instant messaging. During the course of developing BitWise using wxWidgets, Kevin became a wxWidgets developer and has provided enhancements to all platforms. Kevin lives in Oxford, Ohio.

Stefan Csomor is director and owner of Advanced Concepts AG, a company that specializes in cross-platform development and consulting. In addition to being a qualified medical doctor, he has more than 15 years of experience in object-oriented programming and has been writing software for 25 years. Stefan is the main author of the Mac OS port of wxWidgets.

© Copyright Pearson Education. All rights reserved.

Excerpt. © Reprinted by permission. All rights reserved.

Preface Preface Who This Book Is For

This book is a guide to using wxWidgets, an open-source construction kit for writing sophisticated C++ applications targeting a variety of platforms, including Windows, Linux, Mac OS X, and Pocket PC. With help from this book, a competent programmer can create multi-platform applications with confidence. Developers already familiar with wxWidgets should also find it useful for brushing up their knowledge.

This book is accessible to developers with a variety of experience and backgrounds. You may come from a Windows or Unix perspective; you may previously have experience in MFC, OWL, Win32, Mac OS, Motif, or console-mode Unix programming. Or perhaps you have come from a different career entirely and are looking for a way to get up to speed on multiple platforms. The book can't specifically cover the details of the C++ language, but it's common for people to successfully learn C++ and wxWidgets at the same time, and the straightforward nature of the wxWidgets API makes this process easier. The reader does not need to know more advanced C++ techniques like templates, streams, and exceptions. However, wxWidgets does not prevent you from using these techniques.

Managers will find the book useful in discovering what wxWidgets can do for them, particularly in Chapter 1, "Introduction." The combination of the book and the resources on the accompanying CD-ROM will give your staff all they need for getting started on cross-platform programming projects. You'll see how wxWidgets puts tools of tremendous power into your hands, with benefits that include:

- Cost savings from writing code once that will compile on Windows, Unix, Mac OS X, and other platforms.
- Customer satisfaction from delivering stable, fast, attractive applications with a native look and feel.
- Increased productivity from the wide variety of classes that wxWidgets provides, both for creating great GUIs and for general application development.
- Increased market share due to support for platforms you may not have previously considered, and the ability to internationalize your applications.
- Support from a large, active wxWidgets community that answers questions helpfully and provides prompt bug fixing. The sample of third-party add-ons listed in Appendix E, "Third-Party Tools for wxWidgets," is evidence of a thriving ecosystem.
- Access to the source for enhancement and trouble-shooting.

This is a guide to writing wxWidgets application with C++, but you can use a variety of other languages such as Python, Perl, a BASIC variant, Lua, Eiffel, JavaScript, Java, Ruby, Haskell, and C#. Some of these bindings are more advanced than others. For more information, please see Appendix E and the wxWidgets web site at <http://www.wxwidgets.org>.

We focus on three popular desktop platforms: Microsoft Windows, Linux using GTK+, and Mac OS X. However, most of the book also applies to other platforms supported by wxWidgets. In particular, wxWidgets can be used with most Unix variants.

The CD-ROM

The CD-ROM contains example code from the book, the wxWidgets 2.6 distribution for Windows, Linux, Mac OS X, and other platforms, and several tools to help you use wxWidgets, including the translation tool poEdit. For Windows users, we supply three free compilers you can use with wxWidgets: MinGW, Digital Mars C++, and OpenWatcom C++.

In addition, we provide you with DialogBlocks Personal Edition, a sophisticated rapid application development (RAD) tool for you to create complex windows with very little manual coding. You can use it to compile and run samples that accompany the book as well as to create your own applications for personal use, and it also provides convenient access to the wxWidgets reference manual.

Updates to the book and CD-ROM can be obtained from this site:

<http://www.wxwidgets.org/book>

How to Use This Book

It's advisable to read at least Chapters 1 through 10 in order, but you can skip to other chapters if you need to complete a particular task. If you haven't installed wxWidgets before, you may want to look at Appendix A, "Installing wxWidgets," early on. MFC programmers will find it useful to read Appendix K, "Porting from MFC," as a point of reference.

Because this book is not a complete API reference, you'll find it useful to keep the wxWidgets reference manual open. The reference manual is available in a number of formats, including Windows HTML Help and PDF, and it should be in your wxWidgets distribution; if not, it can be downloaded from the wxWidgets web site. You can also refer to the many samples in the wxWidgets distribution to supplement the examples given in this book.

Note that the book is intended to be used in conjunction with wxWidgets 2.6 or later. The majority of the book will apply to earlier versions, but be aware that some functionality will be missing, and in a small number of cases, the behavior may be different. In particular, sizer behavior changed somewhat between 2.4 and 2.5. For details, please see the topic "Changes Since 2.4.x" in the wxWidgets reference manual.

Conventions

For code examples, we mostly follow the wxWidgets style guidelines, for example:

- Words within class names and functions have an initial capital, for example MyFunkyClass.
- The `m_` prefix denotes a member variable, `s_` denotes a static variable, `g_` denotes a global variable; local variables generally start with a lowercase letter, for example `textCtrl`.

You can find more about the wxWidgets style guidelines at <http://www.wxwidgets.org/standard.htm>.

Sometimes we'll also use comments that can be parsed by the documentation tool Doxygen, such as:

```
/// A function description
```

Classes, functions, identifiers, variables, and standard wxWidgets objects are marked with a teletype font in the text. User interface commands, such as menu and button labels, are marked in italics.

Chapter Summary Chapter 1: Introduction

What is wxWidgets, and why use it? A brief history; the wxWidgets community; the license; wxWidgets ports and architecture explained.

Chapter 2: Getting Started

A small wxWidgets sample: the application class; the main window; the event table; an outline of program flow.

Chapter 3: Event Handling

Event tables and handlers; how a button click is processed; skipping events; pluggable and dynamic event handlers; defining custom events; window identifiers.

Chapter 4: Window Basics

The main features of a window explained; a quick guide to the commonest window classes; base window classes such as wxWindow; top-level windows; container windows; non-static controls; static controls; menus; control bars.

Chapter 5: Drawing and Printing

Device context principles; the main device context classes described; buffered drawing; drawing tools; device context drawing functions; using the printing framework; 3D graphics with wxGLCanvas.

Chapter 6: Handling Input

Handling mouse and mouse wheel events; handling keyboard events; keycodes; modifier key variations; accelerators; handling joystick events.

Chapter 7: Window Layout Using Sizers

Layout basics; sizers introduced; common features of sizers; programming with sizers. Further layout issues: dialog units; platform-adaptive layouts; dynamic layouts.

Chapter 8: Using Standard Dialogs

Informative dialogs such as wxMessageBox and wxProgressDialog; file and directory dialogs such as wxFileDialog; choice and selection dialogs such as wxColourDialog and wxFontDialog; entry dialogs such as wxTextEntryDialog and wxFindReplaceDialog; printing dialogs: wxPageSetupDialog and wxPrintDialog.

Chapter 9: Writing Custom Dialogs

Steps in creating a custom dialog; an example: PersonalRecordDialog; deriving a new class; designing data storage; coding the controls and layout; data transfer and validation; handling events; handling UI updates; adding help; adapting dialogs for small devices; further considerations in dialog design; using wxWidgets resource files; loading resources; using binary and embedded resource files; translating resources; the XRC format; writing resource handlers; foreign controls.

Chapter 10: Programming with Images

Image classes in wxWidgets; programming with wxBitmap; programming with wxIcon; programming with wxCursor; programming with wxImage; image lists and icon bundles; customizing wxWidgets graphics with wxArtProvider.

Chapter 11: Clipboard and Drag and Drop

Data objects; data source duties; data target duties; using the clipboard; implementing drag and drop; implementing a drag source; implementing a drop target; using standard drop targets; creating a custom drop target; more on wxDataObject; drag and drop helpers in wxWidgets.

Chapter 12: Advanced Window Classes

wxTreeCtrl; wxListCtrl; wxWizard; wxHtmlWindow; wxGrid; wxTaskBarIcon; writing your own controls; the control declaration; defining a new event class; displaying information; handling input; defining default event handlers; implementing validators; implementing resource handlers; determining control appearance.

Chapter 13: Data Structure Classes

Why not STL? wxString; wxStringTokenizer; wxRegEx; wxArray; wxList; wxHashMap; wxDateTime; wxObject; wxLongLong; wxPoint and wxRealPoint; wxRect; wxRegion; wxSize; wxVariant.

Chapter 14: Files and Streams

wxFile and wxFFile; wxTextFile; wxTempFile; wxDir; wxFileName; file functions; file streams; memory and string streams; data streams; socket streams; filter streams; zip streams; virtual file systems.

Chapter 15: Memory Management, Debugging, and Error Checking

Creating and deleting window objects; creating and copying drawing objects; initializing your application object; cleaning up your application; detecting memory leaks and other errors; facilities for defensive programming; error reporting; providing run-time type information; using wxModule; loading dynamic libraries; exception handling; debugging tips.

Chapter 16: Writing International Applications

Introduction to internationalization; providing translations; using message catalogs; using wxLocale; character encodings and Unicode; converting data; help files; numbers and dates; other media; an example.

Chapter 17: Writing Multithreaded Applications

When to use threads, and when not to; using wxThread; thread creation; starting the thread; how to pause a thread or wait for an external condition; termination; synchronization objects; wxMutex; deadlocks; wxCriticalSection; wxCondition; wxSemaphore; the wxWidgets thread sample; alternatives to multithreading: wxTimer, idle time processing, and yielding.

Chapter 18: Programming with wxSocket

Socket classes and functionality overview; introduction to sockets and basic socket processing; connecting to a server; socket events; socket status and error notifications; sending and receiving socket data; creating a server; socket event recap; socket flags; blocking and non-blocking sockets in wxWidgets; how flags affect

socket behavior; using wxSocket as a standard socket; using socket streams; alternatives to wxSocket.

Chapter 19: Working with Documents and Views

Document/view basics; choosing an interface style; creating and using frame classes; defining your document and view classes; defining your window classes; using wxDocManager and wxDocTemplate; other document/view capabilities; standard identifiers; printing and previewing; file history; explicit document creation; strategies for implementing undo/redo.

Chapter 20: Perfecting Your Application

Single instance versus multiple instances; modifying event handling; reducing flicker; using a help controller; extended wxWidgets HTML help; authoring help; other ways to provide help; parsing the command line; storing application resources; invoking other applications; launching documents; redirecting process input and output; managing application settings; application installation on Windows, Linux and Mac OS X; following UI design guidelines.

Appendix A: Installing wxWidgets

Downloading and unpacking wxWidgets; configuration/build options; Windows—Microsoft Visual Studio and VC++ command-line; Windows—Borland C++; Windows—MinGW with and without MSYS; Unix/Linux and Mac OS X—GCC; customizing setup.h; rebuilding after updating wxWidgets files; using contrib libraries.

Appendix B: Building Your Own wxWidgets Applications

Windows—Microsoft Visual Studio; Linux—KDevelop; Mac OS X—Xcode; makefiles; cross-platform builds using Bakefile; wxWidgets symbols and headers; using wx-config.

Appendix C: Creating Applications with DialogBlocks

What is DialogBlocks? Installing and upgrading DialogBlocks; the DialogBlocks interface; the sample project; compiling the sample; creating a new project; creating a dialog; creating a frame; creating an application object; debugging your application.

Appendix D: Other Features in wxWidgets

Further window classes; ODBC classes; MIME types manager; network functionality; multimedia classes; embedded web browsers; accessibility; OLE automation; renderer classes; event loops.

Appendix E: Third-Party Tools for wxWidgets

Language bindings such as wxPython and wxPerl; tools such as wxDesigner, DialogBlocks and poEdit; add-on libraries such as wxMozilla, wxCURL, wxPropertyGrid.

Appendix F: wxWidgets Application Showcase

Descriptions of notable wxWidgets applications, such as AOL Communicator and Audacity.

Appendix G: Using the CD-ROM

Browsing the CD-ROM; the CD-ROM contents.

Appendix H: How wxWidgets Processes Events

An illustrated description of how event processing works.

Appendix I: Event Classes and Macros

A summary of the important event classes and macros.

Appendix J: Code Listings

Code listings for the PersonalRecordDialog and the wxWizard examples.

Appendix K: Porting from MFC

General observations; application initialization; message maps; converting dialogs and other resources; documents and views; printing; string handling and translation; database access; configurable control bars; equivalent functionality by macros and classes.

Acknowledgments

wxWidgets owes its success to the hard work of many talented people. We would like to thank them all, with special consideration for that essential support network: our long-suffering families and partners. wxWidgets supporters and contributors include the following (apologies for any unintentional omissions):

Yiorgos Adamopoulos, Jamshid Afshar, Alejandro Aguilar-Sierra, Patrick Albert, Bruneau Babet, Mitchell Baker, Mattia Barbon, Nerijus Baliunas, Karsten Ballueder, Jonathan Bayer, Michael Bedward, Kai Bendorf, Yura Bidus, Jorgen Bodde, Borland, Keith Gary Boyce, Chris Breeze, Sylvain Bougnoux, Wade Brainerd, Pete Britton, Ian Brown, C. Buckley, Doug Card, Marco Cavallini, Dmitri Chubraev, Robin Corbet, Cecil Coupe, Stefan Csomor, Andrew Davison, Gilles Depeyrot, Duane Doran, Neil Dudman, Robin Dunn, Hermann Dunkel, Jos van Eijndhoven, Chris Elliott, David Elliott, David Falkinder, Rob Farnum, Joel Farley, Tom Felici, Thomas Fettig, Matthew Flatt, Pasquale Foggia, Josep Fortiana, Todd Fries, Dominic Gallagher, Roger Gammans, Guillermo Rodriguez Garcia, Brian Gavin, Wolfram Gloger, Aleksandras Gluchovas, Markus Greither, Norbert Grotz, Stephane Gully, Stefan Gunter, Bill Hale, Patrick Halke, Stefan Hammes, Guillaume Helle, Harco de Hilster, Kevin Hock, Cord Hockemeyer, Klaas Holwerda, Markus Holzem, Ove Kaaven, Mitch Kapur, Matt Kimball, Hajo Kirchoff, Olaf Klein, Jacob Jansen, Leif Jensen, Mark Johnson, Bart Jourquin, John Labenski, Guilhem Lavaux, Ron Lee, Hans Van Leemputten, Peter Lenhard, Jan Lessner, Nicholas Liebmann, Torsten Liermann, Per Lindqvist, Jesse Lovelace, Tatu Männistö, Lindsay Mathieson, Scott Maxwell, Bob Mitchell, Thomas Myers, Oliver Niedung, Stefan Neis, Ryan Norton, Robert O'Connor, Jeffrey Ollie, Kevin Ollivier, William Osborne, Hernan Otero, Ian Perrigo, Timothy Peters, Giordano Pezzoli, Harri Pasanen, Thomaso Paoletti, Garrett Potts, Robert Rae, Marcel Rasche, Mart Raudsepp, Andy Robinson, Robert Roebing, Alec Ross, Gunnar Roth, Thomas Runge, Tom Ryan, Dino Scaringella, Jobst Schmalenbach, Dimitri Schoolwerth, Arthur Seaton, Paul Shirley, Wlodzimierz Skiba, John Skiff, Vaclav Slavik, Brian Smith, Neil Smith, Stein Somers, Petr Smilauer, Kari Systä, George Tasker, Austin Tate, Arthur Tetzlaff-Deas, Paul Thiessen, Jonathan Tonberg, Jyrki Tuomi, Janos Vegh, Andrea Venturoli, David Webster, Michael Wetherell, Otto Wyss, Vadim Zeitlin, Xiaokun Zhu, Zbigniew Zagórski, Edward Zimmermann. Thanks also to Dotsrc.org and SourceForge for hosting project services.

Thanks are due in particular to Vadim Zeitlin, Vaclav Slavik, Robert Roebing, Stefan Csomor, and Robin Dunn for permission to adapt some of their contributions to the wxWidgets reference manual.

Special thanks go to Stefan Csomor who contributed Chapter 16 and Chapter 17, and to Kevin Ollivier who wrote the Bakefile tutorial in Appendix B. We would also like to thank Mitch Kapor for writing the foreword.

We are very grateful to Mark Taub for his patience and advice throughout. A big thank you goes to Marita Allwood, Harriet Smart, Antonia Smart, Clayton Hock, and Ethel Hock for all their love, support, and encouragement. A debt is also owed to all those who have reviewed and suggested improvements to the book, including: Stefan Csomor, Dimitri Schoolwerth, Robin Dunn, Carl Godkin, Bob Paddock, Chris Elliott, Michalis Kabrianis, Marc-Andre Lureau, Jonas Karlsson, Arnout Engelen, Erik van der Wal, Greg Smith, and Alexander Stigsen.

Finally, we hope that you enjoy reading this book and, most importantly, have fun using wxWidgets to build great-looking, multi-platform applications!

Julian Smart and Kevin Hock
June 2005

© Copyright Pearson Education. All rights reserved.

Most helpful customer reviews

2 of 4 people found the following review helpful.

Developing elegant applications for Linux

By Amazon Customer

wxWidgets is an application framework for developers who want to develop very good-looking applications for Linux, Windows or Mac. The fact that your application can be re-compiled for any of the three operating systems is definitely an attractive plus. My development is mainly for Linux, and when I first ran across wxWidgets, I was amazed at how quickly I could get a simple application going in under an hour. Compiling the library is as easy as issuing a few commands from a terminal window, changing to the samples directory, and running make under the minimal sample directory.

Bear in mind that developing applications is not for complete beginners. You should know C/C++ already, and be reasonably familiar with your OS and compiler. That way, following the examples given in the book, and reading the source code from wxWidgets samples directory, can get you up and running in a minimum of time. I came from a C background, but I had a reasonably complicated inter-airport messaging system for a local government authority working in two weeks, with the help of a serial communications library contributed by another wxWidgets user.

The book comes with a CD full of useful tools and community contributions. I have yet to see, in my 30+ years of programming and project management, a framework as complete as wxWidgets. The support from other users is extraordinary, and you can often be talking to the original developers who are more than willing to help you get going. The fact that it is open source ensures this fantastic library will always be improving.

This book is something I was awaiting for a long time. It is definitely worth the price (and then some).

0 of 1 people found the following review helpful.

Still troubles on OSX

By Amazon Customer

I was never able to get DialogBlocks to build the wx library. Kept freezing halfway through. Eventually I gave up and spent a lot of time setting up an Eclipse environment. (This is on an old Powerbook...) Haven't worked with wx in windows or linux yet, but so far I can say that this book is NOT for the beginner. You should have a pretty solid foundation in windowed application design, makefiles and the gcc linker.

This isn't the book's fault, but cross-platform GUI development is still quite complicated. wxWidgets is a great tool that has really grown into maturity, but it is certainly not panacea :) Fortunately, there are some very patient and experienced developers in the wx-user mailing list.

0 of 0 people found the following review helpful.

Cross-Platform GUI book

By Oleksandr Dzyubak

The book itself is the only one on this matter. It is well written with tons of examples. I would consider this book as a must for those who are writing the

Cross-Platform GUI using open source libs. The book came very fast and in a perfect condition. I would highly recommend buying books from this seller.

See all 37 customer reviews...

CROSS-PLATFORM GUI PROGRAMMING WITH WXWIDGETS BY JULIAN SMART, KEVIN HOCK WITH, STEFAN CSOMOR PDF

Based on some encounters of many people, it remains in truth that reading this **Cross-Platform GUI Programming With wxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor** could help them making better choice as well as offer even more encounter. If you intend to be one of them, let's acquisition this book Cross-Platform GUI Programming With wxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor by downloading guide on link download in this website. You can get the soft file of this publication Cross-Platform GUI Programming With wxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor to download and install and deposit in your readily available digital gadgets. Exactly what are you waiting for? Allow get this book Cross-Platform GUI Programming With wxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor online and read them in any time and also any kind of area you will read. It will not encumber you to bring hefty book Cross-Platform GUI Programming With wxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor inside of your bag.

From the Back Cover

"This book is the best way for beginning developers to learn wxWidgets programming in C++. It is a must-have for programmers thinking of using wxWidgets and those already using it."

—Mitch Kapor, founder of Lotus Software and the Open Source Applications Foundation

- Build advanced cross-platform applications that support native look-and-feel on Windows, Linux, Unix, Mac OS X, and even Pocket PC
- Master wxWidgets from start to finish—even if you've never built GUI applications before
- Leverage advanced wxWidgets capabilities: networking, multithreading, streaming, and more
- CD-ROM: library of development tools, source code, and sample applications
- Foreword by Mitch Kapor, founder, Lotus Development and Open Source Application Foundation

wxWidgets is an easy-to-use, open source C++ API for writing GUI applications that run on Windows, Linux, Unix, Mac OS X, and even Pocket PC—supporting each platform's native look and feel with virtually no additional coding. Now, its creator and two leading developers teach you all you need to know to write robust cross-platform software with wxWidgets. This book covers everything from dialog boxes to drag-and-drop, from networking to multithreading. It includes all the tools and code you need to get great results, fast. From AMD to AOL, Lockheed Martin to Xerox, world-class developers are using wxWidgets to save money, increase efficiency, and reach new markets. With this book, you can, too.

- wxWidgets quickstart: event/input handling, window layouts, drawing, printing, dialogs, and more
- Working with window classes, from simple to advanced
- Memory management, debugging, error checking, internationalization, and other advanced topics
- Includes extensive code samples for Windows, Linux (GTK+), and Mac OS X

About the CD-ROM

The CD-ROM contains all of the source code from the book; wxWidgets distributions for Windows, Linux,

Unix, Mac OS X, and other platforms; the wxWidgets reference guide; and development tools including the OpenWatcom C++ compiler, the poEdit translation helper, and the DialogBlocks user interface builder.

© Copyright Pearson Education. All rights reserved.

About the Author

Julian Smart has degrees from the University of St. Andrews and the University of Dundee. After working on model-based reasoning at the Scottish Crop Research Institute, he moved to the Artificial Intelligence Applications Institute at the University of Edinburgh, where he founded the wxWidgets project in 1992. Since starting Anthemion Software in 1996, Julian has been helping other companies deploy wxWidgets, and he sells tools for programmers, including DialogBlocks and HelpBlocks. He has worked as a consultant for various companies including Borland and was a member of Red Hat's eCos team, writing GUI tools to support the embedded operating system. In 2004, Julian and his wife Harriet launched a consumer product for fiction writers called Writer's Café, written with wxWidgets. Julian and Harriet live in Edinburgh with their daughter Toni.

Kevin Hock has degrees from Miami University (Oxford, Ohio) in Computer Science and Accounting and has taught courses at Miami in both Java and client-server systems. In 2002, he started work on an instant messaging system and founded BitWise Communications, LLC, in 2003, offering both professional and personal instant messaging. During the course of developing BitWise using wxWidgets, Kevin became a wxWidgets developer and has provided enhancements to all platforms. Kevin lives in Oxford, Ohio.

Stefan Csomor is director and owner of Advanced Concepts AG, a company that specializes in cross-platform development and consulting. In addition to being a qualified medical doctor, he has more than 15 years of experience in object-oriented programming and has been writing software for 25 years. Stefan is the main author of the Mac OS port of wxWidgets.

© Copyright Pearson Education. All rights reserved.

Excerpt. © Reprinted by permission. All rights reserved.

Preface Preface Who This Book Is For

This book is a guide to using wxWidgets, an open-source construction kit for writing sophisticated C++ applications targeting a variety of platforms, including Windows, Linux, Mac OS X, and Pocket PC. With help from this book, a competent programmer can create multi-platform applications with confidence. Developers already familiar with wxWidgets should also find it useful for brushing up their knowledge.

This book is accessible to developers with a variety of experience and backgrounds. You may come from a Windows or Unix perspective; you may previously have experience in MFC, OWL, Win32, Mac OS, Motif, or console-mode Unix programming. Or perhaps you have come from a different career entirely and are looking for a way to get up to speed on multiple platforms. The book can't specifically cover the details of the C++ language, but it's common for people to successfully learn C++ and wxWidgets at the same time, and the straightforward nature of the wxWidgets API makes this process easier. The reader does not need to know more advanced C++ techniques like templates, streams, and exceptions. However, wxWidgets does

not prevent you from using these techniques.

Managers will find the book useful in discovering what wxWidgets can do for them, particularly in Chapter 1, "Introduction." The combination of the book and the resources on the accompanying CD-ROM will give your staff all they need for getting started on cross-platform programming projects. You'll see how wxWidgets puts tools of tremendous power into your hands, with benefits that include:

- Cost savings from writing code once that will compile on Windows, Unix, Mac OS X, and other platforms.
- Customer satisfaction from delivering stable, fast, attractive applications with a native look and feel.
- Increased productivity from the wide variety of classes that wxWidgets provides, both for creating great GUIs and for general application development.
- Increased market share due to support for platforms you may not have previously considered, and the ability to internationalize your applications.
- Support from a large, active wxWidgets community that answers questions helpfully and provides prompt bug fixing. The sample of third-party add-ons listed in Appendix E, "Third-Party Tools for wxWidgets," is evidence of a thriving ecosystem.
- Access to the source for enhancement and trouble-shooting.

This is a guide to writing wxWidgets application with C++, but you can use a variety of other languages such as Python, Perl, a BASIC variant, Lua, Eiffel, JavaScript, Java, Ruby, Haskell, and C#. Some of these bindings are more advanced than others. For more information, please see Appendix E and the wxWidgets web site at <http://www.wxwidgets.org>.

We focus on three popular desktop platforms: Microsoft Windows, Linux using GTK+, and Mac OS X. However, most of the book also applies to other platforms supported by wxWidgets. In particular, wxWidgets can be used with most Unix variants.

The CD-ROM

The CD-ROM contains example code from the book, the wxWidgets 2.6 distribution for Windows, Linux, Mac OS X, and other platforms, and several tools to help you use wxWidgets, including the translation tool poEdit. For Windows users, we supply three free compilers you can use with wxWidgets: MinGW, Digital Mars C++, and OpenWatcom C++.

In addition, we provide you with DialogBlocks Personal Edition, a sophisticated rapid application development (RAD) tool for you to create complex windows with very little manual coding. You can use it to compile and run samples that accompany the book as well as to create your own applications for personal use, and it also provides convenient access to the wxWidgets reference manual.

Updates to the book and CD-ROM can be obtained from this site:

<http://www.wxwidgets.org/book>

How to Use This Book

It's advisable to read at least Chapters 1 through 10 in order, but you can skip to other chapters if you need to complete a particular task. If you haven't installed wxWidgets before, you may want to look at Appendix A, "Installing wxWidgets," early on. MFC programmers will find it useful to read Appendix K, "Porting from MFC," as a point of reference.

Because this book is not a complete API reference, you'll find it useful to keep the wxWidgets reference

manual open. The reference manual is available in a number of formats, including Windows HTML Help and PDF, and it should be in your wxWidgets distribution; if not, it can be downloaded from the wxWidgets web site. You can also refer to the many samples in the wxWidgets distribution to supplement the examples given in this book.

Note that the book is intended to be used in conjunction with wxWidgets 2.6 or later. The majority of the book will apply to earlier versions, but be aware that some functionality will be missing, and in a small number of cases, the behavior may be different. In particular, sizer behavior changed somewhat between 2.4 and 2.5. For details, please see the topic "Changes Since 2.4.x" in the wxWidgets reference manual.

Conventions

For code examples, we mostly follow the wxWidgets style guidelines, for example:

- Words within class names and functions have an initial capital, for example `MyFunkyClass`.
- The `m_` prefix denotes a member variable, `s_` denotes a static variable, `g_` denotes a global variable; local variables generally start with a lowercase letter, for example `textCtrl`.

You can find more about the wxWidgets style guidelines at <http://www.wxwidgets.org/standard.htm>.

Sometimes we'll also use comments that can be parsed by the documentation tool Doxygen, such as:

```
/// A function description
```

Classes, functions, identifiers, variables, and standard wxWidgets objects are marked with a teletype font in the text. User interface commands, such as menu and button labels, are marked in italics.

Chapter Summary Chapter 1: Introduction

What is wxWidgets, and why use it? A brief history; the wxWidgets community; the license; wxWidgets ports and architecture explained.

Chapter 2: Getting Started

A small wxWidgets sample: the application class; the main window; the event table; an outline of program flow.

Chapter 3: Event Handling

Event tables and handlers; how a button click is processed; skipping events; pluggable and dynamic event handlers; defining custom events; window identifiers.

Chapter 4: Window Basics

The main features of a window explained; a quick guide to the commonest window classes; base window classes such as `wxWindow`; top-level windows; container windows; non-static controls; static controls; menus; control bars.

Chapter 5: Drawing and Printing

Device context principles; the main device context classes described; buffered drawing; drawing tools; device context drawing functions; using the printing framework; 3D graphics with `wxGLCanvas`.

Chapter 6: Handling Input

Handling mouse and mouse wheel events; handling keyboard events; keycodes; modifier key variations; accelerators; handling joystick events.

Chapter 7: Window Layout Using Sizers

Layout basics; sizers introduced; common features of sizers; programming with sizers. Further layout issues: dialog units; platform-adaptive layouts; dynamic layouts.

Chapter 8: Using Standard Dialogs

Informative dialogs such as `wxMessageBox` and `wxProgressDialog`; file and directory dialogs such as `wxFileDialog`; choice and selection dialogs such as `wxColourDialog` and `wxFontDialog`; entry dialogs such as `wxTextEntryDialog` and `wxFindReplaceDialog`; printing dialogs: `wxPageSetupDialog` and `wxPrintDialog`.

Chapter 9: Writing Custom Dialogs

Steps in creating a custom dialog; an example: `PersonalRecordDialog`; deriving a new class; designing data storage; coding the controls and layout; data transfer and validation; handling events; handling UI updates; adding help; adapting dialogs for small devices; further considerations in dialog design; using `wxWidgets` resource files; loading resources; using binary and embedded resource files; translating resources; the XRC format; writing resource handlers; foreign controls.

Chapter 10: Programming with Images

Image classes in `wxWidgets`; programming with `wxBitmap`; programming with `wxIcon`; programming with `wxCursor`; programming with `wxImage`; image lists and icon bundles; customizing `wxWidgets` graphics with `wxArtProvider`.

Chapter 11: Clipboard and Drag and Drop

Data objects; data source duties; data target duties; using the clipboard; implementing drag and drop; implementing a drag source; implementing a drop target; using standard drop targets; creating a custom drop target; more on `wxDataObject`; drag and drop helpers in `wxWidgets`.

Chapter 12: Advanced Window Classes

`wxTreeCtrl`; `wxListCtrl`; `wxWizard`; `wxHtmlWindow`; `wxGrid`; `wxTaskBarIcon`; writing your own controls; the control declaration; defining a new event class; displaying information; handling input; defining default event handlers; implementing validators; implementing resource handlers; determining control appearance.

Chapter 13: Data Structure Classes

Why not STL? `wxString`; `wxStringTokenizer`; `wxRegEx`; `wxArray`; `wxList`; `wxHashMap`; `wxDateTime`; `wxObject`; `wxLongLong`; `wxPoint` and `wxRealPoint`; `wxRect`; `wxRegion`; `wxSize`; `wxVariant`.

Chapter 14: Files and Streams

`wxFile` and `wxFFFile`; `wxTextFile`; `wxTempFile`; `wxDir`; `wxFileName`; file functions; file streams; memory and string streams; data streams; socket streams; filter streams; zip streams; virtual file systems.

Chapter 15: Memory Management, Debugging, and Error Checking

Creating and deleting window objects; creating and copying drawing objects; initializing your application object; cleaning up your application; detecting memory leaks and other errors; facilities for defensive programming; error reporting; providing run-time type information; using `wxModule`; loading dynamic libraries; exception handling; debugging tips.

Chapter 16: Writing International Applications

Introduction to internationalization; providing translations; using message catalogs; using `wxLocale`; character encodings and Unicode; converting data; help files; numbers and dates; other media; an example.

Chapter 17: Writing Multithreaded Applications

When to use threads, and when not to; using `wxThread`; thread creation; starting the thread; how to pause a thread or wait for an external condition; termination; synchronization objects; `wxMutex`; deadlocks; `wxCriticalSection`; `wxCondition`; `wxSemaphore`; the `wxWidgets` thread sample; alternatives to multithreading: `wxTimer`, idle time processing, and yielding.

Chapter 18: Programming with `wxSocket`

Socket classes and functionality overview; introduction to sockets and basic socket processing; connecting to a server; socket events; socket status and error notifications; sending and receiving socket data; creating a server; socket event recap; socket flags; blocking and non-blocking sockets in `wxWidgets`; how flags affect socket behavior; using `wxSocket` as a standard socket; using socket streams; alternatives to `wxSocket`.

Chapter 19: Working with Documents and Views

Document/view basics; choosing an interface style; creating and using frame classes; defining your document and view classes; defining your window classes; using `wxDocManager` and `wxDocTemplate`; other document/view capabilities; standard identifiers; printing and previewing; file history; explicit document creation; strategies for implementing undo/redo.

Chapter 20: Perfecting Your Application

Single instance versus multiple instances; modifying event handling; reducing flicker; using a help controller; extended `wxWidgets` HTML help; authoring help; other ways to provide help; parsing the command line; storing application resources; invoking other applications; launching documents; redirecting process input and output; managing application settings; application installation on Windows, Linux and Mac OS X; following UI design guidelines.

Appendix A: Installing `wxWidgets`

Downloading and unpacking `wxWidgets`; configuration/build options; Windows—Microsoft Visual Studio and VC++ command-line; Windows—Borland C++; Windows—MinGW with and without MSYS; Unix/Linux and Mac OS X—GCC; customizing `setup.h`; rebuilding after updating `wxWidgets` files; using contrib libraries.

Appendix B: Building Your Own `wxWidgets` Applications

Windows—Microsoft Visual Studio; Linux—KDevelop; Mac OS X—Xcode; makefiles; cross-platform builds using Bakefile; `wxWidgets` symbols and headers; using `wx-config`.

Appendix C: Creating Applications with DialogBlocks

What is DialogBlocks? Installing and upgrading DialogBlocks; the DialogBlocks interface; the sample project; compiling the sample; creating a new project; creating a dialog; creating a frame; creating an application object; debugging your application.

Appendix D: Other Features in wxWidgets

Further window classes; ODBC classes; MIME types manager; network functionality; multimedia classes; embedded web browsers; accessibility; OLE automation; renderer classes; event loops.

Appendix E: Third-Party Tools for wxWidgets

Language bindings such as wxPython and wxPerl; tools such as wxDesigner, DialogBlocks and poEdit; add-on libraries such as wxMozilla, wxCURL, wxPropertyGrid.

Appendix F: wxWidgets Application Showcase

Descriptions of notable wxWidgets applications, such as AOL Communicator and Audacity.

Appendix G: Using the CD-ROM

Browsing the CD-ROM; the CD-ROM contents.

Appendix H: How wxWidgets Processes Events

An illustrated description of how event processing works.

Appendix I: Event Classes and Macros

A summary of the important event classes and macros.

Appendix J: Code Listings

Code listings for the PersonalRecordDialog and the wxWizard examples.

Appendix K: Porting from MFC

General observations; application initialization; message maps; converting dialogs and other resources; documents and views; printing; string handling and translation; database access; configurable control bars; equivalent functionality by macros and classes.

Acknowledgments

wxWidgets owes its success to the hard work of many talented people. We would like to thank them all, with special consideration for that essential support network: our long-suffering families and partners. wxWidgets supporters and contributors include the following (apologies for any unintentional omissions):

Yiorgos Adamopoulos, Jamshid Afshar, Alejandro Aguilar-Sierra, Patrick Albert, Bruneau Babet, Mitchell Baker, Mattia Barbon, Nerijus Baliunas, Karsten Ballueder, Jonathan Bayer, Michael Bedward, Kai Bendorf, Yura Bidus, Jorgen Bodde, Borland, Keith Gary Boyce, Chris Breeze, Sylvain Bougnoux, Wade Brainerd, Pete Britton, Ian Brown, C. Buckley, Doug Card, Marco Cavallini, Dmitri Chubraev, Robin Corbet, Cecil

Coupe, Stefan Csomor, Andrew Davison, Gilles Depeyrot, Duane Doran, Neil Dudman, Robin Dunn, Hermann Dunkel, Jos van Eijndhoven, Chris Elliott, David Elliott, David Falkinder, Rob Farnum, Joel Farley, Tom Felici, Thomas Fettig, Matthew Flatt, Pasquale Foggia, Josep Fortiana, Todd Fries, Dominic Gallagher, Roger Gammans, Guillermo Rodriguez Garcia, Brian Gavin, Wolfram Gloger, Aleksandras Gluchovas, Markus Greither, Norbert Grotz, Stephane Gully, Stefan Gunter, Bill Hale, Patrick Halke, Stefan Hammes, Guillaume Helle, Harco de Hilster, Kevin Hock, Cord Hockemeyer, Klaas Holwerda, Markus Holzem, Ove Kaaven, Mitch Kapor, Matt Kimball, Hajo Kirchoff, Olaf Klein, Jacob Jansen, Leif Jensen, Mark Johnson, Bart Jourquin, John Labenski, Guilhem Lavaux, Ron Lee, Hans Van Leemputten, Peter Lenhard, Jan Lessner, Nicholas Liebmann, Torsten Liermann, Per Lindqvist, Jesse Lovelace, Tatu Männistö, Lindsay Mathieson, Scott Maxwell, Bob Mitchell, Thomas Myers, Oliver Niedung, Stefan Neis, Ryan Norton, Robert O'Connor, Jeffrey Ollie, Kevin Ollivier, William Osborne, Hernan Otero, Ian Perrigo, Timothy Peters, Giordano Pezzoli, Harri Pasanen, Thomaso Paoletti, Garrett Potts, Robert Rae, Marcel Rasche, Mart Raudsepp, Andy Robinson, Robert Roebing, Alec Ross, Gunnar Roth, Thomas Runge, Tom Ryan, Dino Scaringella, Jobst Schmalenbach, Dimitri Schoolwerth, Arthur Seaton, Paul Shirley, Wlodzimierz Skiba, John Skiff, Vaclav Slavik, Brian Smith, Neil Smith, Stein Somers, Petr Smilauer, Kari Systä, George Tasker, Austin Tate, Arthur Tetzlaff-Deas, Paul Thiessen, Jonathan Tonberg, Jyrki Tuomi, Janos Vegh, Andrea Venturoli, David Webster, Michael Wetherell, Otto Wyss, Vadim Zeitlin, Xiaokun Zhu, Zbigniew Zagórski, Edward Zimmermann. Thanks also to Dotsrc.org and SourceForge for hosting project services.

Thanks are due in particular to Vadim Zeitlin, Vaclav Slavik, Robert Roebing, Stefan Csomor, and Robin Dunn for permission to adapt some of their contributions to the wxWidgets reference manual.

Special thanks go to Stefan Csomor who contributed Chapter 16 and Chapter 17, and to Kevin Ollivier who wrote the Bakefile tutorial in Appendix B. We would also like to thank Mitch Kapor for writing the foreword.

We are very grateful to Mark Taub for his patience and advice throughout. A big thank you goes to Marita Allwood, Harriet Smart, Antonia Smart, Clayton Hock, and Ethel Hock for all their love, support, and encouragement. A debt is also owed to all those who have reviewed and suggested improvements to the book, including: Stefan Csomor, Dimitri Schoolwerth, Robin Dunn, Carl Godkin, Bob Paddock, Chris Elliott, Michalis Kabrianis, Marc-Andre Lureau, Jonas Karlsson, Arnout Engelen, Erik van der Wal, Greg Smith, and Alexander Stigsen.

Finally, we hope that you enjoy reading this book and, most importantly, have fun using wxWidgets to build great-looking, multi-platform applications!

Julian Smart and Kevin Hock
June 2005

© Copyright Pearson Education. All rights reserved.

When you are hurried of task target date and also have no idea to obtain inspiration, **Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor** publication is one of your solutions to take. Schedule Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor will certainly give you the appropriate source as well as point to obtain inspirations. It is not only concerning the jobs for politic business, management, economics, and various other. Some bought works making some fiction jobs also need motivations to conquer the task. As just what

you require, this Cross-Platform GUI Programming With WxWidgets By Julian Smart, Kevin Hock With, Stefan Csomor will possibly be your selection.