[jamesdewolf@apress.com](mailto:jamesdewolf@apress.com)

The Windows and Compiler books

Hi Jim,

Long time, no see! How are you? I hope everything is fine with you. I have been busy lately, during the spring I have moved to another city and begun a new job.

I have heard that Windows 10 will be released on July 29. I will upgrade my computer to Windows 10 as soon as possible, modify the code in accordance with Windows 10, and I hope we can publish the Windows book rather soon.

I have also written the first draft of the compiler book, which I hope you are interested in. It describes in detail how to develop an optimized C compiler in Java. The book starts with a small toy compiler demonstrating how to use the scanner and parser generating tools JLex and CUP. The rest of the book describes the phases of the C compiler: scanning, parsing, type checking, symbol table generation, middle code generation and optimization, target code generation, register allocation, and linking. The grammar, preprocessor, standard library and the C crash course from my original proposal about Windows programming in C are also included. I would say that the code is finished, but the text describing the code may need some more work (especially Chapter 12), I hope we can work on it together.

There are other books about compiler technology, below are the most popular ones. However, they do not target any specific source language or target architecture.

Alfred V. Aho, Lam, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers – Principles, Techniques, and Tools, 2nd edition. Prentice Hall, 2006.

Keith Cooper, Linda Torczon. Engineering a Compiler, 2nd edition. Morgan Kaufman, 2011.

Charles N. Fischer, Ron K. Cytron, Richard J. LeBlanc. Crafting a Compiler. Pearson Education, 2009.

Andrew W Appel. Modern Compiler Implementation in Java, 2nd edition. Cambridge University Press, 2002.

Steven S. Muchnick. Advanced Compiler Design & Implementation. Morgan Kaufmann, 2003.

The following book is the one that is most similar to mine, it describes a C compiler developed in C. However, it is outdated and it does not describe any particular target architecture.

Christopher W. Fraser, David R. Hanson. A Retargetable C Compiler : Design and Implementation. Benjamin Cummings, 1995.

If you are interested in publishing the books, which I certainly hope you are, please let me know when you want to start proofreading them, and I will send you the latest drafts.

I enclose the script in Word and PDF formats. I admit that the scripts are not perfect at the moment, but with some more work they can become real good books.

Best Regards,

Stefan Björnander

[jamesdewolf@apress.com](mailto:jamesdewolf@apress.com)

Hi Jim,

I just want to add that I would appreciate if you let me know when you assign a proof reader, so that I can send you the latest version of the script.

Best Regards,

Stefan

Hi Jim,

How are you? It has been a while, I hope everything is fine with you.

I have been working with the Windows book. I have mostly focused on the code, and I start to feel satisfied with its quality. The ultimate (but unrealistic) goal is that the code is so clear that it is self-explaining. I have also written the first draft of the script, which I enclose; it holds at the moment 362 pages. I will continue to work on the script because I feel that the difficult parts need some more explanation, and maybe I need to lighten up the explanations from the easier parts. I also need to look at the bigger picture, explain the overall purpose and function of the applications before I dive into the details.

I regret to say that my personal situation is unchanged. I have applied for work and have attended some interviews, but I have not yet been offered a job. My plan is to finish both the Windows and compiler books before I find and begin a new job. Therefore, I plan to finish the compiler and start to write its script. I will continue to work on the Windows script eventually, but in the near future, I will focus on the compiler. The idea is that if I take a break with the Windows script, I will later read it with fresh eyes and more clearly see what needs to be improved.

Best Regards,

Stefan Björnander

Hi Jim,

I hope you have had a good summer so far. I have recently visited my parents at the Swedish west coast, and we have had unusually warm weather.

I have also written a tutorial for Small Windows. I think it is a good idea to introduce the reader to the applications from the start, one chapter for each application, and then add the tutorial as an appendix together with a final appendix that explains the architecture of Small Windows.

Best Regards,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎måndag‎ den ‎30‎ ‎juni‎ ‎2014 ‎16‎:‎43  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan.  Thanks for the update on your book and for the new project.  I’m sorry to hear about your job.  I don’t have a lot of contacts who might be hiring, but I think the market is good for people with your skills so hopefully it won’t take too long for you to find a new position.

Let me take some time to review your compiler idea in more detail, and I’ll get back to you.  BTW, I had the privilege to work with Aho, Sethi and Ullman on the last revision of their classic Dragon compiler book.

Thanks,

Jim

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Friday, June 27, 2014 7:29 AM  
**To:** James T. DeWolf  
**Subject:** Three points I would like to communicate to you

Hi Jim,

I hope you have had a good vacation. I will also go on vacation for a few weeks. However, I would like to communicate three points to you:

1.      I have rewritten the book’s source code, and it seems to work fine. My plan is to divide the book into two parts, where the first part is made up by five chapters, one chapter for each application. The applications are written in C++ with the Small Windows library, with the Win32 API details completely hidden. In this way we can introduce the reader to Windows development and focus on the application-specific problems without the complex technical Win32 API details.

The second part is made up by two chapters where the first one is a Small Windows tutorial describing the library without explaining it. Finally, the second chapter explains the technical details of Small Windows.

In this way we can address a large set of readers, ranging from the readers who want a head-first approach into Windows development without complicated technical details to the readers who want to know every detail of the Windows architecture.

2.      I became unemployed a few week ago. Until recently I was employed as an industrial Ph.D. candidate at a Swedish company. However, due to the financial recession**,** the management was forced to reduce the work force. In Sweden, we have a rule that says that the last one hired becomes the first one to be given notice in case of shortage of work, and I was the last one hired.

The good news is that I can spend more time at the book, the bad news is that I will need a job in the not-to-distant future. Therefore, I wonder if you know any company in New York (or elsewhere in America) that needs a skilled programmer. If you do, I would be grateful if you will introduce me.

One possible problem is that it could be difficult for me as non-US citizen to be granted a work permit, but I feel it is worth a try. After all, there are many people who have immigrated to the United States, and built good lives for themselves.

3.      I have another book proposal for you, this time about compiler technology. The book will describe the design and implementation of a C++ compiler (including the preprocessor, linker, and standard library), with all code given.

So far, I have written a C compiler and I plan to extend it to a C++ compiler in the near future. Even though C++ is a much larger language than C, it not too much extra job. Most of the job has already been done with the C compiler.

The idea of the book is basically the same as the Windows book: I explain the details and provide all source code of the compiler. The source code written in Java with CUP and JLex and generates both assembler code and executable code.

The books below are the latest editions of three classic books. They describe in detail the basic features of compiler construction and to some extent advanced features. The structure of these books is similar to mine, one can say that I have followed these books. However, while these books give a large number of small examples, my book describes in each chapter the features necessary to be included in a C++ compiler.

Alfred V. Aho, Lam, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers – Principles, Techniques, and Tools, 2nd edition. Prentice Hall, 2006.

Keith Cooper, Linda Torczon. Engineering a Compiler, 2nd edition. Morgan Kaufman, 2011.

Charles N. Fischer, Ron K. Cytron, Richard J. LeBlanc. Crafting a Compiler. Pearson Education, 2009.

The following books describe compiler construction in Java, C, and ML. They describe a compiler for a smaller language with excerpt from the code given in the book (the complete code is downloadable). However, these books are briefer than the books above, and describe the compiler features in less detail.

Andrew W Appel. Modern Compiler Implementation in Java, 2nd edition. Cambridge University Press, 2002.

Andrew W Appel. Modern Compiler Implementation in C, 2nd edition. Cambridge University Press, 2004.

Andrew W Appel. Modern Compiler Implementation in ML, 2nd edition. Cambridge University Press, 2004.

The following book could be said to be closely related to mine, as it also described the code for a C compiler. However, this book is more or less unreadable. It presents the code, which (in my option) is unstructured, and it does not include much text describing the code. Hopefully, my book will describe the topic in a clearer way. I also plan to describe the theory of a compiler, not just the code to implement it. Moreover, this book does not include the preprocessor, linker, or standard library, and the code is written in C, not Java.

Christopher W. Fraser, David R. Hanson. A Retargetable C Compiler : Design and Implementation. Benjamin Cummings, 1995.

This books deals with the more advanced parts of compiler construction; that is, compiler optimization. Even though I will include some optimization at the end of my book, it does not compete with this book.

Steven S. Muchnick. Advanced Compiler Design & Implementation. Morgan Kaufmann, 2003.

Here is a suggestion for the chapters:

1. Introduction. Introduces the compiler phases described in the following chapters by demonstrating a compiler for a small toy language generating MIPS-code executable in the SPIM simulator.

2. The Scanner. The scanner is a relatively small part of the compiler. Its task is to put together characters into tokens (the smallest significant parts of the source code). Examples are as key words, operators, and numerical values. The scanner is written in JLex, a lexical generator for Java, based on Lex for C.

3. The Parser. The parser, on the other hand, is a large part of the compiler. Its task is to confirm that the given tokens (generated by the scanner) agree with the syntax of the programming language, which is represented by a set of grammatical rules. The parser is defined in CUP, a syntax generator for Java, based on Yacc for C. Each rule can also be equipped with code dealing with type checking and target code generation. However, I will have try to omit as much as possible of the code in this chapter. Most of the code is made up of calls to methods defined in later chapters.

4. Declarations and the Symbol Table. C++ has a rather complicated declaration system with aggregated types such as classes, structures and arrays with a corresponding complicated syntax. All defined variables and functions are stored in the symbol table, which is a hierarchical structure matching the program structure.

5. Type Checking. C++ has a rather large set of operators with complicated rules that need to be checked.

6. Intermediate Code Generation. When the types of expressions and statements are checked, three-address-code are generated, which is a simple intermediate language used to represent the code internally. Type conversation is also included in this chapter.

7. Static and Dynamic Initialization. In C++, variables can be initialized. If the variable is static, the data shall be generated and placed in the static area of the final target code. If it is dynamic, the initialization will result in a series of assignments. One thing that complicates the issue is that it is possible to initialize hierarchical structures made up by structures and arrays with one flat list.

8. Intermediate Code Optimization. The intermediate code can be optimized. For instance, code that is never reached and assignment of variables that are never used shall be removed.

9. Target Code Generation. The target code of the C++ compiler of this book is Intel x86, which is harder to deal with than the MIPS code the first chapter. It holds a few registers and registers of different sizes overlap. Therefore, the register allocation process needs to closely keep track on which variable values that are currently stored in the registers.

10. The Preprocessor. Before the actually compilation starts, the source code has been traversed by the preprocessor that replaces macros with text, includes header files, and provides conditional programming.

11. The Linker. When the target code has finally been generated, it becomes stored into an object file. As the source code can be distributed over several files, the target code need to be merged into one executable file. This is the task of the linker, it merges the code and data area, resolve the static and extern references and generate the executable file.

12. The C Standard Library, made up by functions and macros, which will be included by the linker in the final executable file.

13. The C++ Standard Library, made up by classes, to a certain extent is based on C standard library.

Best Regards,

Stefan

Hi Jim. Yes, I am free.

Skickat från Windows E-post

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎fredag‎ den ‎23‎ ‎maj‎ ‎2014 ‎16‎:‎42  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan. I just skype message you but you may not have seen it.  Are you free for a brief chat?

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Thursday, May 22, 2014 8:10 PM  
**To:** James T. DeWolf  
**Subject:** Re: C Windows Programming proposal--Question on Windows 8.1

Hi Jim,

That will be fine, I will be at the computer tomorrow.

I have just seen “X-men – days of future past”, which I can recommend.

Best,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎torsdag‎ den ‎22‎ ‎maj‎ ‎2014 ‎22‎:‎13  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan.  Yes, let’s do that. I’ll review all the revised material and be ready in the morning my time.  I am going away for the two weeks following so it would be good to get this before the ed board one more time before then.

Have a good evening

Jim

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Thursday, May 22, 2014 2:59 PM  
**To:** James T. DeWolf  
**Subject:** Re: C Windows Programming proposal--Question on Windows 8.1

Hi Jim,

I am on my way to a movie tonight, but maybe we can talk by Skype tomorrow?

Best Regards,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎onsdag‎ den ‎21‎ ‎maj‎ ‎2014 ‎22‎:‎20  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Thanks Stefan. I’m looking forward to the next installment.

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Wednesday, May 21, 2014 12:29 PM  
**To:** James T. DeWolf  
**Subject:** Re: C Windows Programming proposal--Question on Windows 8.1

Hi Jim,

I have to hurry to a meeting, so I send you this mail now. I will send you a second draft in a few hours.

Good news: I have implemented the Small Windows library and rewritten the Ring application in the beginning of the book, and it seems to work fine. The code will be much easier to understand with the two levels.

I am flexible when it comes to the focus on Window 8.1, I leave it up to you. The code works for Windows 7 and Windows 8. With small modification we can make it work for Windows XP, or even older versions. I also sure that the code will work with future version of Windows. We can present the book with Window 8.1 in the title, or we can just mention the versions of Windows and Visual Studio in the preface. That could possible extend the life span of the book, especially if the market for Windows 8.1 is weak at the moment.

I am also flexible regarding the title, and I leave it to you to find an appropriate title. “C++ Windows Programming” or “Windows Programming in C++” may sound a little bit too generic. Maybe “Applied Windows Programming in C++” or “Windows Application Development in C++”, or “Applied Windows Development in C++”.

*1.       Is this really an advanced book? I’m not sure that we captured that clearly in the description.  What makes it advanced?  Perhaps it is in the details of developing the seemingly simple applications.  Can you help me be more explicit in identifying the experience level of expected readers?  We need to be clear about this on Amazon, etc.*

Well, the applications are advanced, in my opinion. For instance, the Calc application uses graph theoretical search algorithms in order to keep the cells up-to-date. The expression parser uses a generic binary tree to evaluate the expressions. The Word application calculates the size and position of each individual character in the document, with regards to their font, size, and alignment. However, the Windows technology described in is not that advanced, especially since some of the types I implemented in C are available in the C++ standard library. The technology

*2.      Is the Win32 issue equally applicable when you use C++?  Does the description need to change somewhat?*

Yes, Win32 API is relevant for all Windows applications. The only difference is how much functionality is placed between the application and Win32 API. In our case, we have the Small Windows library, which is explained in the appendix and hides the details in chapter. That makes the code easy to understand, in my opinion,

*3.      Not surprisingly at TechEd, everything seemed to be based on Visual Studio.  Can you characterize why someone would benefit from your book’s approach as opposed to simply using VS?  I think this is the key issue—what are the benefits of this approach to developers?*

By just using Visual Studio you do not learn anything. You need at least a beginner’s textbook to start with.

*4.      My take on Windows 8.1 is that MS will keep releasing enhancements and “maybe” a major release next spring.  Of course this is an educated guess.  We’ll need to respond with minor updates as needed since  I think the days of major releases seem to be behind us.  This means that some minor (or not so minor) updates may be required in order to keep the book relevant.  We are facing this with many of our titles these days.*

That is not a problem. As soon as new versions of Windows or Visual Studio are released, I test the code and make all the modifications necessary (in the code and in the book text) for it to work.

Best Regards,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎onsdag‎ den ‎21‎ ‎maj‎ ‎2014 ‎00‎:‎23  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan.  I’m glad that you are still interested in moving forward with a new plan and I appreciate your patience with our efforts to produce a successful book.  I suggest that we clarify some of the objectives and target audience questions before you do a lot of revising, to make sure that we get the details right and obtain contract approval.  If we get the blueprint for the project right, I’m confident that you will write a successful book.

With so many variables having changed, I think it makes sense for both of us to revisit the proposal and make sure that we can clearly articulate where the new book will fit and what audience it will serve. I have attached a word file with the latest (unedited) description of the previous version.

The comments below are from a member of our editorial board who had some difficulty understanding the direction we were taking, possibly because of the title or my poor translation of your objectives:

I was reading the book data, I thought: this is odd, it appears to be an 8.1 book, so why does it have a generic title? Having read the thread now, I can see that perhaps the description has more 8.1 infused into it as a result.

So that would be logical on 8.1 based on the authors views that this is a Win 8.1 book - but three questions are unresolved in my mind, which are

- 8.1 market as mentioned below is not great right now

- 8.x update coming?

- the title is misleading because it is generic still?

There's also the issue that this is an API-centric book, and the title doesn't communicate that -- if it were stated to be an 8.1 book per the author, it would still be ambiguous whether it were an API-drive book, a .NET centric title, or even a JavaScript drive book.

So my impression is that we're trying to universalize this book with a title that the author doesn't mean. Author seems to have a specific goal here - Win8.1, and API 32. I think the description and title are still not really saying this, and could mislead seriously.

I am left with the following questions that need to be addressed before we can gain approval:

1.       Is this really an advanced book? I’m not sure that we captured that clearly in the description.  What makes it advanced?  Perhaps it is in the details of developing the seemingly simple applications.  Can you help me be more explicit in identifying the experience level of expected readers?  We need to be clear about this on Amazon, etc.

2.      Is the Win32 issue equally applicable when you use C++?  Does the description need to change somewhat?

3.      Not surprisingly at TechEd, everything seemed to be based on Visual Studio.  Can you characterize why someone would benefit from your book’s approach as opposed to simply using VS?  I think this is the key issue—what are the benefits of this approach to developers?

4.      My take on Windows 8.1 is that MS will keep releasing enhancements and “maybe” a major release next spring.  Of course this is an educated guess.  We’ll need to respond with minor updates as needed since  I think the days of major releases seem to be behind us.  This means that some minor (or not so minor) updates may be required in order to keep the book relevant.  We are facing this with many of our titles these days.

Again, happy to discuss any of this at your convenience either by Skype or email.  I’m on holiday for two weeks following this week but will try to keep things moving on my end.

Thanks,

Jim

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Tuesday, May 20, 2014 8:53 AM  
**To:** James T. DeWolf  
**Subject:** Re: C Windows Programming proposal--Question on Windows 8.1

Hi Jim,

I read your mail yesterday evening (Swedish time), and I have done some thinking.

Yes, C++ is a more modern language than C and a book about C++ will probably have a larger market. It is possible to rewrite the code from C to C++, and I am interested in doing so. The idea is that I rewrite the applications of chapter one to five from C to C++, taking advantages of a small C++ object-oriented library (written by me, let us call it “Small Windows” for the time being) that hides the technical Win32 API details. In this way, the book becomes divided into two layers:

        The applications of chapter one to five. The code will have the same functionality, but written in an object-oriented style in C++. For instance, the figures of the Tetris and Draw applications will be organizes in an object-oriented hierarchy. The code will focus on the application-specific functionality and rely on the Small Windows library for low-level system calls. In this way, we start directly with the application development and keep the head-first approach.

        The Small Windows C++ object-oriented library, presented in an appendix, which explains the architecture of Windows. At the moment, I thinking something like:

o   The Window class. A standard window with title, appropriate buttons, and scrollbars. It is possible to add listener to catch Windows messages regarding movement, mouse action, touch screen actions, and menu items.

o   Classes that encapsulates the standard dialogs regarding file opening and saving, colors, and fonts.

o   The application class, which encapsulates the main message pump of the application.

o   The context device class that encapsulates the device context used for drawing and writing in the window.

o   Unit classes that transforms units between the different logical and device coordinate systems.

The argument against this approach would be that there are already several object-oriented libraries in Windows. Why write another one? The answer is that Small Windows will be small, easy to use and (above all) easy to understand. The established libraries (MFC, Windows Forms, WPF, WinRT, …) are large and (more or less) easy to use, but they are certainly not easy to understand.

Moreover, I suggest we omit C crash course in appendix A and the ASCII table in appendix D altogether. As you have pointed out, the reader of the book is most surly already familiar with C and C++. The abstract data types from appendix B will be mostly omitted and replaced by C++ standard types. I am skeptical whether we shall keep the apps in appendix C. If we do, we shall probably rewrite them in C++ instead of C#.

To sum it up: the idea is still to guide the reader through the development of advanced Windows applications, with all the technical detail of the architecture available. The main different is that it will be done in C++ instead of C.

This will take some time and require some amount of work. I will as soon as possible rewrite the first application and report back to you.

If this sounds interesting to you, I suggest we agree about the legal terms and sign the contract as soon as possible.

Best Regards,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎måndag‎ den ‎19‎ ‎maj‎ ‎2014 ‎18‎:‎06  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan.  I’m back from TechEd now.  While I was there I spent considerable time tracking down people from Microsoft and my user group contacts to chat about some of the issues you and I have been discussing over the last few weeks.  I admit that I was disappointed with the comments I heard about the viability of your approach based on the size of the potential market.  It is not a reflection of the technical quality of your project, but more a matter of concern about how many people in the market would purchase a book based on C.  As one contact put it. “It’s all done in C++ now, and it really doesn’t make a difference”.   I also spoke to some presenters on related topics and they had to think hard to find someone I could reach out to who might be sympathetic to your approach.

I realize that we have been back and forth on this project quite a bit and I’m reluctant to ask you to do more work without more of a commitment from us.  With that said, I’m sure that we both want to publish a book that has reasonable sales and I’m afraid the book in its current state would have limited sales.  Do you have any interest in modifying the project to meet the larger C++ market?  I know that this represents a significant shift in your philosophy and you may prefer to continue on a different path.  I’m happy to have another Skype session to discuss if you are interested.

Thanks,

Jim

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

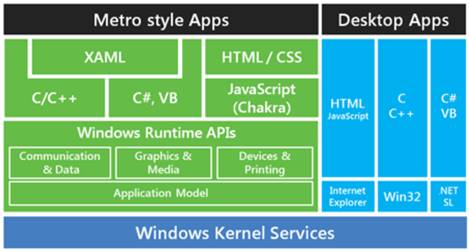
**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Friday, May 09, 2014 3:40 PM  
**To:** James T. DeWolf  
**Subject:** Re: C Windows Programming proposal--Question on Windows 8.1

Hi Jim,

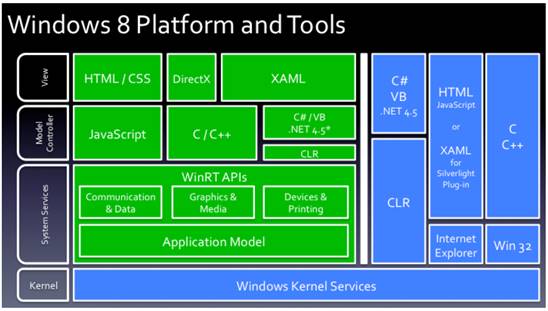
Since this mail contains pictures, I enclose it in PDF format. I have also added a Windows Store app (a Calculator) at the end of the script.

The book is intended for Windows 8.1 users, and I think we shall be clear about it. The text and code are fully adjusted to and functional in Windows 8.1. I suggest we change the title to “C Windows Programming in Windows 8.1” or “Windows API Programming in Windows 8.1”. I see no reason to mention Windows 7 anywhere in the book.

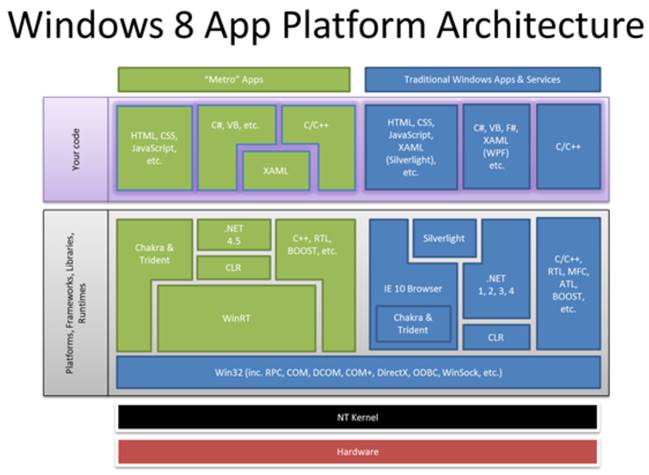
When it comes to WinRT, I have borrowed a diagram from <http://www.win8tutorial.net/getting-started/windows-runtime/>. Metro is the old name for Windows Store.



Here is another diagram from <http://dougseven.com/2011/09/15/a-bad-picture-is-worth-a-thousand-long-discussions/> that is said to be more accurate.



Here is a third diagram from <http://bitcrazed.com/post/2012/01/27/An-Accurate-Windows-8-Platform-Architecture-Diagram.aspx> that is said to be even more accurate. As you can see, there is a lot of discussion and confusion about the technical details of Windows 8.



As you can see from all three diagrams, the Windows 8 platform is divided into two parts, it does actually make sense to say that Windows 8 is two operating system in one: Windows Store (the green part) and Windows Desktop (the blue part). There is also two different development environments available: Visual Studio for Windows Desktop and Visual Studio for Windows Store (both downloadable without cost). Windows 8 Desktop is a development of Windows 7 and designed for computers while Windows Store is a new feature designed for smart phones. Microsoft want to target both the computer and smart phone markets. So to sum it up: Windows Store is the new part and Windows Desktop is the traditional part. However, Windows Desktop is just as central part of Windows 8.1 as Windows Store and necessary for development and execution of complex applications.

If you want to develop apps (for smart phones or Windows 8) to be distributed by Windows Store, WinRT is a good choice. However, according to Microsoft you cannot access Win32 API in Windows Store, and have thereby no access to the underlying system. Microsoft want the apps to be executable to as many (computer, surf pads, smart phones, …) platforms as possible.

On the other hand, if you want to develop real applications (Word, Excel, …) for Windows Desktop you cannot use WinRT. Instead, you have to use Win32 API, or any of the wrapping object-oriented libraries (MFC, Windows Forms, WPF, …). So to sum it up, when it comes to application (not app) development in Windows 8.1, we are faced with the same possibilities and decisions as in Windows 7. As I have argued before, Win32 API is the only system that gives you full access to the architecture and give you full insight in how the system works, while the object-oriented libraries hides the technical details.

Finally a few words about Win32 API and Win64 API. There are three version of the Windows API: Win16 API (from Windows 1.0 to Windows XP), Win32 API (from Windows 95), and Win64 API (from Windows NT 64-bit version). The situation today is that Windows 8.1 can be executed on both 32 bits processors (called x86, after the original 8086 processor) and 64 bits processors (called x64, after 64 bits), and thereby use both Win32 API and Win64 API. Win64 API is [backwards compatible](http://en.wikipedia.org/wiki/Backwards_compatibility) with Win32 API. The only difference is that some values (integer, pointers) hold the size of 64 bits instead of 32 bits. However, that problem is easily solved by macros in the source code. Therefore, the term Win32 API is regarded to cover both Win32 API and Win64 API, since the difference between them is very small. The source code in the book works fine on both 32 and 64 bits processors. In fact, Windows 8.1 comes in both 32 and 64 bits versions, and the installment program detects which kind of processor the computer executes on.

I suggest we present this information in a preface (I will not copy the diagrams above, but draw a similar one) in order to explain the architecture of Windows 8.1 for the reader and to show that the book is modern and up-to-date.

Best Regards,

Stefan

Skickat från Windows E-post

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎fredag‎ den ‎9‎ ‎maj‎ ‎2014 ‎11‎:‎28  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan.  The question of how to handle Windows 8.1 in the description is a valid one.  I don’t want to exaggerate the coverage or applicability of the book to 8.1 developers, but I agree with Jeff (my boss) that we need to be more clear about this as it relates to the description of coverage in the book.  Is it fair to say that the principles learned here will apply to Win 8.1 developers?  To be clear, I’m assuming that the book is primarily focused on Windows 7 but how would you characterize its relevance to WinRT and 8.1 developers? Jeff’s point is that we need to be very clear who is the ideal target audience for the book in the Amazon description because that first impression will determine whether or not a potential reader will investigate further.  Let me know how you think we should best answer this question.

Thanks,

Jim

**From:** Jeffrey Pepper   
**Sent:** Thursday, May 08, 2014 9:18 PM  
**To:** James T. DeWolf  
**Subject:** C Windows Programming proposa;

The first issue I see is that the description talks about WinRT and Windows 7.  So the question that the reader is asking is “What about Windows 8”?  And that lingering question means that anyone that thinks about that question will not buy the book.  So we need to answer that type of question in our description or lose sales.

However, in the last line of the 3rd paragraph, there is a reference to Wndows 8.1, so what is the real story?   As a consumer this disparity in description would leave me to reject the book.  BTW, I think you must at least make some play toward Win8..

Jeffrey M. Pepper

Associate Publisher

Apress

910-363-4237

[jeffrey.pepper@apress.com](mailto:jeffrey.pepper@apress.com)

Hi Jim,

I have put together some thoughts about the marketing of the book:

Title

* I would want to include the phrases “Applied”, “Application Development”, and “Windows Architecture” in the title, but maybe not “Advanced”.
  + “Applied Application Development with the Windows Architecture”

Focus on Window 8.1

* No, not really. I suggest we just focus on Windows.

Who are the readers?

* Students
  + Who want to prepare for a career as application developer
  + Who want to learn professional programming
* Programmers with experience of other platforms
  + Linux
  + Macintosh
* Professional Programmers
  + Who want to see “the whole picture” of application development.
  + Who are interested in solving more complex graphical problems:
    - How to calculate the area suitable for dragging a line or a text.
    - How to display text that are aligned to the left, right, or center, or justified.
    - How to display text where the characters have different fonts.

What will the reader learn?

* Programming Skills
  + Not just small example, but complete applications.
* Knowledge about how to use Win32 API
  + Not just small examples, but a complete library.

Best Regards,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎torsdag‎ den ‎22‎ ‎maj‎ ‎2014 ‎22‎:‎13  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan.  Yes, let’s do that. I’ll review all the revised material and be ready in the morning my time.  I am going away for the two weeks following so it would be good to get this before the ed board one more time before then.

Have a good evening

Jim

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Thursday, May 22, 2014 2:59 PM  
**To:** James T. DeWolf  
**Subject:** Re: C Windows Programming proposal--Question on Windows 8.1

Hi Jim,

I am on my way to a movie tonight, but maybe we can talk by Skype tomorrow?

Best Regards,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎onsdag‎ den ‎21‎ ‎maj‎ ‎2014 ‎22‎:‎20  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Thanks Stefan. I’m looking forward to the next installment.

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

Hi Jim,

That will be fine, I will be at the computer tomorrow.

I have just seen “X-men – days of future past”, which I can recommend.

Best,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎torsdag‎ den ‎22‎ ‎maj‎ ‎2014 ‎22‎:‎13  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Hi Stefan.  Yes, let’s do that. I’ll review all the revised material and be ready in the morning my time.  I am going away for the two weeks following so it would be good to get this before the ed board one more time before then.

Have a good evening

Jim

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

**From:** Stefan Bjornander [mailto:stefan.bjornander@outlook.com]   
**Sent:** Thursday, May 22, 2014 2:59 PM  
**To:** James T. DeWolf  
**Subject:** Re: C Windows Programming proposal--Question on Windows 8.1

Hi Jim,

I am on my way to a movie tonight, but maybe we can talk by Skype tomorrow?

Best Regards,

Stefan

**Från:** [James T. DeWolf](mailto:jamesdewolf@apress.com)  
**Skickat:** ‎onsdag‎ den ‎21‎ ‎maj‎ ‎2014 ‎22‎:‎20  
**Till:** [Stefan Bjornander](mailto:stefan.bjornander@outlook.com)

Thanks Stefan. I’m looking forward to the next installment.

Jim DeWolf

Senior Editor

Apress/Springer Science+Business Media

978-891-4110

The books below are the latest editions of three classic books. They describe in detail the basic features of compiler construction and to some extent advanced features. The structure of these books is similar to mine, one can say that I have followed these books. However, while these books give a large number of small examples, my book describes in each chapter the features necessary to be included in a C++ compiler.

Alfred V. Aho, Lam, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Compilers – Principles, Techniques, and Tools, 2nd edition. Prentice Hall, 2006.

Keith Cooper, Linda Torczon. Engineering a Compiler, 2nd edition. Morgan Kaufman, 2011.

Charles N. Fischer, Ron K. Cytron, Richard J. LeBlanc. Crafting a Compiler. Pearson Education, 2009.

The following books describe compiler construction in Java, C, and ML. They describe a compiler for a smaller language with excerpt from the code given in the book (the complete code is downloadable). However, these books are briefer than the books above, and describe the compiler features in less detail.

Andrew W Appel. Modern Compiler Implementation in Java, 2nd edition. Cambridge University Press, 2002.

Andrew W Appel. Modern Compiler Implementation in C, 2nd edition. Cambridge University Press, 2004.

Andrew W Appel. Modern Compiler Implementation in ML, 2nd edition. Cambridge University Press, 2004.

The following book could be said to be closely related to mine, as it also described the code for a C compiler. However, this book is more or less unreadable. It presents the code, which (in my option) is unstructured, and it does not include much text describing the code. Hopefully, my book will describe the topic in a clearer way. I also plan to describe the theory of a compiler, not just the code to implement it. Moreover, this book does not include the preprocessor, linker, or standard library, and the code is written in C, not Java.

Christopher W. Fraser, David R. Hanson. A Retargetable C Compiler : Design and Implementation. Benjamin Cummings, 1995.

This books deals with the more advanced parts of compiler construction; that is, compiler optimization. Even though I will include some optimization at the end of my book, it does not compete with this book.

Steven S. Muchnick. Advanced Compiler Design & Implementation. Morgan Kaufmann, 2003.

Here is a suggestion for the chapters: