

ter

telecommunications electronic reviews

Volume 4, Issue 6, June, 1997

Telecommunications Electronic Reviews (TER) is a publication of the Library and Information Technology Association.

Telecommunications Electronic Reviews (ISSN: 1075-9972) is a periodical copyright © 1997 by the American Library Association. Documents in this issue, subject to copyright by the American Library Association or by the authors of the documents, may be reproduced for noncommercial, educational, or scientific purposes granted by Sections 107 and 108 of the Copyright Revision Act of 1976, provided that the copyright statement and source for that material are clearly acknowledged and that the material is reproduced without alteration. None of these documents may be reproduced or adapted for commercial distribution without the prior written permission of the designated copyright holder for the specific documents.

Contents

- REVIEW OF: Brent E. Rector and Joseph M. Newcomer. Win32 Programming. by Alicia Abramson
- REVIEW OF: William Holderby. Real-World Networking with NT 4. by Joe Ford
- REVIEW OF: Trevor Harmon. Web Developer's Guide to Visual J++ & ActiveX. by Shirl Kennedy
- REVIEW OF: Raif O. Onvural. Asynchronous Transfer Mode Networks: Performance Issues (2nd edition). by Christinger Tomer
- About TER

 [ter issues \(/lita/publications/archive/ter\)](/lita/publications/archive/ter)

REVIEW OF: Brent E. Rector and Joseph M. Newcomer. Win32 Programming. Reading, MA: Addison-Wesley Developers Press, 1997.

by Alicia Abramson

Win32 Programming will serve as an important reference tool for Windows programmers--those already programming for the 32-bit Windows Operating System (OS) as well as those seeking to transfer their 16-bit Windows programming skills to the new model. At 1529 pages with a 247 page index, this book strives to be, and nearly is, comprehensive in its coverage of the Windows Graphical User Interface (GUI) and the Application Programming Interface (API) used in Windows 95 and Windows NT.

The authors are very thorough and cover a wide range of topics from the creation of basic, "bare-bones" windows to basic and advanced graphics creation and manipulation, the use and management of the various Windows controls, memory management, and threads management (the facility through which Windows manages synchronization between multiple processes). While the authors target advanced Windows programmers and advanced C programmers with little to no Windows programming experience, the work will no doubt be useful as a reference source to motivated beginners as well, though it should be stressed, this is not a "learning to program" text.

The book consists of 18 chapters, four appendices, and as mentioned above, an extensive index which includes instructions on how to use it effectively. The authors have done an excellent job of organizing the book--in addition to the table of contents, the front matter includes a six-page listing of tables that appear throughout the book and a five-page listing of all of the code examples. The tables listing will be especially useful for those searching for a way to quickly look up a Win32 class, object, or API function set on which they need more information. Additionally, the code examples, both uncompiled and precompiled versions, are on an accompanying CD-ROM which installs the files and programs smoothly into a separate Win32 Programming subdirectory.

The authors include a brief annotated list of further reading at the end of almost every chapter. The reference lists show a thoughtful balance between hard-core programming resources and more general works that discuss user interface development, "unauthorized" or behind the scenes Win32 reference works, and general treatises on good programming style.

Another nice touch is the ten different types of "text insets," or shaded boxes with various symbols and brief text notes, which appear throughout the book to alert the reader to issues of importance that don't fall neatly into the main discussion in the text. These insets include one for "bugs" either in the Microsoft Win32 OS documentation or software distributed to developers, a "Win16" inset which alerts 16-bit Windows programmers to differences between programming on the 16-bit and 32-bit platforms, a "Windows 95" inset that highlights features of Windows 95 that are different from Windows NT 4.X and which could lead to cross-platform incompatibility if included in code, and a similar alert for Windows NT 4.X.

The authors begin the first chapter much like an introductory text, answering the question "what is Windows?" and providing a brief history of the development of GUIs and of the evolution of the Windows environment/operating system. They lay out the conceptual model for Windows programming and briefly cover some of the main concepts key to Windows programming such as Windows operating models, the porting of 16-bit code, static and dynamic linking, and Windows memory management.

They also address the following paradox: while programming for Win32 is essentially an object oriented approach, C, the language on which the book is based, "does not provide support for direct expression of these object-oriented concepts. Therefore, you must, in effect, now write procedural code in a non-object oriented language that correctly models an object-oriented design solution." (p.12) The authors also note, however, that in writing this book, they used the Microsoft Foundation Classes (FMC) developer's environment which includes C++ libraries containing prebuilt Windows objects such as windows and scroll

bars. Readers should be aware that if they use a different developer's environment, such as Borland C++, some adjustments must be made to the code examples if they are to compile correctly. In many cases these are noted, but not always.

Chapters two through four deal with basic window creation, window styles, displaying text in windows, and all of the behind-the-scenes communication between the various windows and the Win32 OS. A basic "skeletal Microsoft Windows program" is introduced in chapter two, which serves as the foundation of many of the later programs discussed in the book. In chapter three, the "Window Explorer" program, available on the CD-ROM, is discussed. The program allows the user to select different window styles and to create windows on the fly to observe the effects of the styles selected.

Chapters five and six discuss the Windows Graphics Device Interface (GDI). GDI creates and keeps track of the context for different system graphics devices such as the screen and printer. Displaying graphics on the screen, controlling the six GDI objects (bitmap, brush, font, palette, and pen), and mapping coordinates to the screen are discussed in chapter five. Chapter six tackles advanced graphics such as drawing and manipulating ellipses, polygons, disjointed lines, bezier curves and bitmaps, creating custom brushes and pens, and creating matrices which are used to transform shapes. The "GDI Explorer" program introduced in chapter six allows the reader to experiment with these different effects.

Chapter seven is the first chapter where user input is addressed. The focus is on the keyboard, mouse and system timer. The concept of "input focus" is introduced. In addition, different character sets such as ASCII (American Standard Code for Information Interchange) and Unicode are discussed, and structured exception handling techniques are explored.

Chapters eight through ten cover the creation and use of various Windows controls such as buttons, check boxes, radio buttons, list boxes, combo boxes, scroll bars, and trackbars. Each of these controls has its own object class, and allowable styles or variations of these controls are determined by the class characteristics.

In one particularly interesting example of how a relatively small component can wreak havoc in a program, the authors detail in chapter ten their difficulty in tracking down several bugs (13 to be exact) in Microsoft's implementation and documentation of a particular control, the trackbar. The trackbar is a sliding bar that is used to indicate a value on a scale defined by the programmer. Among other things, the authors discovered that the programmer-defined frequency of tick marks on the trackbar in some cases is not retained by Windows when the program is run; that despite documentation to the contrary, negative values can be used in a trackbar range; and: "The tick marks do not form a 'set' in the mathematical sense. If you use the TBM_SETTIC message ... to set a value that is already in the tick mark array, you will get duplicate entries in the tick mark array." (p.732)

Chapters 11 and 12 cover the use of "prepackaged" Windows functions such as dialog boxes, message boxes, menus, accelerators, and icons.

Chapters 13 through 16 cover printing, memory management, fonts, and the use of Windows Dynamic Link Libraries (DLLs). The discussion of memory management in chapter 14 will be of particular interest to those familiar with 16-bit Windows programming. The authors reveal that in fact memory management has become easier in the Win32 environment--while at the same time stressing the importance of understanding how the basics of Windows memory management work such as the use of memory heaps, memory blocks and the allocation of dynamic memory. Chapter 16, which discusses Dynamic Link Libraries in depth, is essential reading for Win16 programmers, since much has changed in the implementation of DLLs in Win32.


The final two chapters (17 and 18) discuss the Windows Multiple Document Interface (MDI) and synchronization. MDI is the Windows user interface specification for the handling of multiple open windows within an application--one of the most useful features of the Windows environment. Since Windows provides built-in support for most of the MDI functions, the goal of this discussion is primarily informational, though quite an extensive code example is provided which, when compiled, is the "MDI Explorer" program.

Synchronization is the process of allocating system resources to the multiple process threads running in both the Win32 OS and active applications. The "Synchronization Explorer" is useful in demonstrating to the reader how threads run and some of the pitfalls that lie in wait for the unaware programmer.

Overall, Win32 Programming will serve as a solid cornerstone in the Addison-Wesley "Advanced Windows Series." The authors are thorough and clear and intent not only on familiarizing the reader with the overwhelming number of API calls, functions, and classes in the Win32 programming environment, but also work to instill the importance of following correct, logical procedures.

Alicia Abramson (aabrams@american.edu (mailto:aabrams@american.edu)) is the Head of Library Systems at American University in Washington DC.

Copyright © 1997 by Alicia Abramson. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at aabrams@american.edu (mailto:aabrams@american.edu).

[table of contents](#)  ter issues (/lita/publications/archive/ter)

REVIEW OF: William Holderby. Real-World Networking with NT 4. Scottsdale, AZ: The Coriolis Group, 1996.

by Joe Ford

This book is the third volume in my NT 4 collection, and occupies an honorable, if diminished, position in relation to the other entries. Given the meteoric rise in interest and use of NT since the release in mid-1996 of version 4 with its Windows 95-like interface, we can expect to see more of these thick paperback how-to-do-it-right tomes. First place in our office goes to the must-have Mastering NT 4.0 Server by Mark Minasi, which Microsoft (MS) itself gives to libraries participating in MS's Libraries Online grant program training sessions. Second place goes to MS's own Resource Kit for NT Workstation, which contains tips and software tricks which should have been in the release version of NT's desktop version. A companion volume from MS Press provides a Resource Kit for NT Server.

But, back to this volume. It provides very useful tips and tricks, but does not help a new user sort out all the vagaries of an NT 4 server installation. My own problems (debugged and repaired with the help of Minasi's fine book) are not identified in this volume, despite the apparently fairly common nature of network and peripheral card conflicts that occur in NT. This book offers very useful information on troubleshooting and performance tuning of operating NT networks, and I refer to it occasionally for that purpose. It further includes very useful information on managing NT for Internet servers. The book also introduces and describes MS's Resource Kit utilities, but MS does this better in its own publication.

The book is lengthy (600 pages) and well organized, with an emphasis on practical software issues. The format is paperback, divided into five parts and twelve chapters. Each part has a theme: Part 1 is "Understanding NT Networking," Part 2 is "Planning and Designing your NT Network," Part 3 is "Maintaining Your Network," Part 4 is "Extending Your Network," and Part 5 is "Reviewing Your Network's Performance." The prose is clear and unambiguous, and explains technical matters in coherent language. One of the book's strengths is its inclusion of tables identifying NT's large number of software modules by name, and identifying what each does in the complex NT environment. The book also includes a number of tips and case studies on performance monitoring and troubleshooting (for software only).

Holderby also does a fine job of explaining and documenting use of the Remote Procedure Call (RPC), a powerful NT feature that permits an NT computer to serve up applications to other machines. The book also does a good job on NT's Remote Access Service (RAS), including installing, configuring, and managing RAS and the RAS Administration software. Among the book's other strengths is its step-by-step approach (with illustrations of dialog boxes and screen displays) for configuring NT security, Point to Point Protocol (PPP), and other networking protocols and processes.

As noted earlier, the book focuses strictly on NT server and client software, and does not deal with the installation of equipment on the network, nor does it discuss issues in installing or operating NT on specific hardware platforms. The very important Hardware Abstraction Layer (HAL) within NT receives only the briefest discussion.

The accompanying CD-ROM provides some very useful shareware utilities, one of which I highly recommend. This virtually indispensable utility is ExecSoft's Diskeeper Lite, a very functional and effective de-fragmenter to aid in dealing with NT's penchant for scattering file fragments all over disk drives, whether the drives are formatted for traditional File Attribute Table (FAT) format or the NT-specific NT File System (NTFS). While Diskeeper Lite is available as freeware on the Internet, its presence on the CD-ROM is a real bonus, and nearly justifies the cost of the book. Except for Diskeeper, the contents of CD-ROMs from MS Service Pack 2 and NT Resource Kit from MS are superior in that they provide basic functionality or utilities of really great value for NT, but the contents of this CD-ROM have merit.

NT itself is an estimable piece of work: powerful, smooth in operation, feature-rich, secure (after MS plugs the holes the hackers have found), and well suited for work group, Internet, and intranet services. With the power in NT comes some challenges: installation and configuration, management, exploiting the power and features and extensive utilities, and troubleshooting the inevitable surprises.

A library undertaking use of NT 4 server and workstation software needs more documentation than Microsoft provides, and will be surprised to discover that purchasing NT confers no Microsoft support as part of the purchase. Instead, new owners of NT licenses must buy Microsoft's time, or third-party documentation, if they experience problems.

In those regards, books like Holderby's fill a real need, and I recommend the addition of this book to any collection of NT-related technical support documents.

Joe Ford (fordjb@wln.com (mailto:fordjb@wln.com)) is the President of Joseph Ford & Associates of Olympia, Washington.

Copyright © 1997 by Joe Ford. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at fordjb@wln.com (mailto:fordjb@wln.com).

REVIEW OF: Trevor Harmon. Web Developer's Guide to Visual J++ & ActiveX. Scottsdale, AZ: Coriolis Group Books, 1996.

by Shirl Kennedy

This is a how-to-do-it book about two somewhat controversial technologies. ActiveX is Microsoft's solution to adapting OLE (Object Linking and Embedding--a "document-centric" process allowing different software applications to communicate with each other) to the World Wide Web. ActiveX controls, as the book points out, "are self-contained programs that live inside the Web," (p. 11) or "reusable code in binary form." (p. 102) Since they can access files on a user's machine, they are more powerful than plain vanilla Java, but also a bit dicier. Various ActiveX-related security problems have been unmasked and/or decried by hackers and researchers.

While Java is a programming language, ActiveX is more accurately described as an "architecture." ActiveX controls can be written with any programming language: C++, Visual Basic, or Java itself. But plain vanilla Java on its own cannot communicate with other popular OLE-enabled software programs. ActiveX allows this to happen, so that Java needn't be isolated from other applications.

Alas, only one Java development tool facilitates this sort of tight integration. Yep, you guessed right--Visual J++, from the guys and gals in Redmond. And this is another source of controversy; some programmers and IS types feel Microsoft is in the process of turning open standards into its own proprietary ones. And yet, for those already working in a Microsoft development environment, Visual J++ is certainly worth a look.

Harmon's book is not for the clueless newbie or the faint-of-heart. If you don't already know at least the rudiments of the Java language, don't start here. I'm not a programmer, though I've played around with Visual J++ a bit. Two better book choices for the novice are:

Winters, P., Olhasso, D., Lemay, L., & Perkins, C. L. (1996). *Teach Yourself Visual J++ in 21 Days*. Indianapolis, IN: Sams.net Publishing.

Davis, S. R. (1996). *Learn Java Now*. Redmond, WA: Microsoft Press.

And even so, Java ain't the kind of thing you can teach yourself in an idle weekend like you can HTML (Hypertext Markup Language). This is serious stuff, folks!

Harmon--who has written for *The C++ User's Journal*, *The Windows Developer's Journal*, and *Visual Developer Magazine*--was, when he wrote this book, an undergraduate student at Washington University in St. Louis. He's worked for Microdyne Development Technologies as well as Washington University's Computer Science Department. Given this background, the book is less geeky than you'd expect. Harmon communicates well. Even though much of the really technical stuff was over my head, I was able to finally get a good grip on what ActiveX is and how it relates to Java. And I think I have a better idea of how this all relates to Web development--as a more efficient substitute for CGI (Common Gateway Interface), for example, that distributes processing between a user's client and a remote Web server, rather than dumping the whole burden on the server.

There's a lot of sample code in the book, and all of it is available on the CD-ROM that accompanies the book, along with a bunch of other stuff like "appealing applets and cool controls, including a plasma generator, a bitmap rotator, and a video player." (inside front cover) The book itself is divided, more or less,


into three parts. First, it introduces you to the basics of Java and ActiveX. Then it covers interfaces, integrating ActiveX controls into Java applets, writing ActiveX controls with Java, and manipulating controls and applets with HTML scripts. The third section talks about "OLE automation"--how Java and ActiveX work with other software applications, including existing databases. Security is discussed thoroughly and objectively in its own chapter, with an emphasis on code signing for authentication. There's also a good bibliography of online resources.

To his credit, Harmon does not appear to be in the thrall of Gates and Company. When he thinks Microsoft has dropped the ball on some particular feature, he doesn't hesitate to tell you about it--the limitations of Visual J++'s "Resource Wizard," for instance. This makes the book a good choice for someone interested in learning about these two Microsoft-based technologies, but who prefers not to swallow the "party line" from Redmond.

That someone, however, had better have a programming background.

Shirl Kennedy (sdk@mindspring.com (mailto:sdk@mindspring.com)) is the technical information specialist at Honeywell's space systems group in Clearwater, Florida. She writes the monthly "Internet Waves" column for Information Today (<http://www.infotoday.com/it/itnew.htm> (http://www.infotoday.com/it/itnew.htm)), and is the author of a forthcoming book about high-powered Internet research from ALA Editions.

Copyright © 1997 by Shirl Kennedy. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at sdk@mindspring.com (mailto:sdk@mindspring.com).

[table of contents](#)  ter issues (/lita/publications/archive/ter)

REVIEW OF: Raif O. Onvural. Asynchronous Transfer Mode Networks: Performance Issues (2nd edition). Norwood, MA: Artech House, 1995.

by Christinger Tomer

Asynchronous Transfer Mode (ATM) is one of the technologies, perhaps the technology, that will define the next generation or two of networking, first across network backbones and then across MANs (Metropolitan Area Networks), campus-wide networks, and LANs (Local Area Networks). Notwithstanding the virtues of competing technologies such as FDDI (Fiber Distributed Data Interface), ATM is widely viewed as the best solution for building cost-effective, multi-protocol networks in the near term. Why? Because it is considered by most experts to be the best available technology for supporting network applications that demand concurrent voice, video and data.

Where does ATM stand today? Many of the necessary baseline specifications of ATM are now complete, including numerous physical layer specifications, User-to-Network Interface (UNI) 3.0 and 3.1, LAN emulation for connection of ATM networks to existing LANs, network management capabilities and test suites. In addition, several interworking specifications have been completed, thus enabling connection of ATM networks to existing wide area network services.

On other fronts, deployment has evolved from simple trials, to sophisticated pilots, to the appearance of first-generation production networks. Users are implementing ATM to support a variety of applications, from the collection of oceanographic environmental data to satellite transmission to interactive entertainment via cable television. In the all-important realm of wide-area networking, public and private service providers alike are currently engaged in pilot projects, working to ensure successful integration and interworking with other service offerings.

Raif Onvural's *Asynchronous Transfer Mode Networks: Performance Issues* is essentially a primer, albeit a highly technical one. This book examines ATM as a technology for LAN and WAN (Wide Area Network) applications.

The strength of *ATM Networks* is in its detail. Onvural considers what ATM is, how it works, how ATM solutions are being developed, and how such solutions are likely to work. Another strength of this book is that, although it is aimed at a technologically sophisticated audience--it is, after all, a detailed analysis of a new and complicated technology set in an even more complex milieu--Onvural makes a considerable effort to admit a more general readership, mainly by prefacing each section of the book with explanations that are more broadly accessible. Early on, a reader interested in learning about ATM and the issues attending its deployment need not be a computer scientist or telecommunications engineer to derive benefit from a reading of this book.


But, as Onvural presses on, the book becomes less accessible to the reader without an engineering background. The chapters on traffic management, switching, signaling, and routing are full of mathematical formulas and references to specific electrical engineering processes. A full appreciation of these chapters will be difficult for many readers outside the engineering community.

The exception among the later chapters is the one on connectionless (or LAN-like) services. Not only is it generally more accessible than the chapters immediately preceding it, it is also particularly valuable, because descending deployment of ATM is a critical issue in the delivery of broadband services to the desktop. Specifically, Onvural examines three different approaches, the ITU-T (International Telecommunication Union, Telecommunication Standardization Sector) server method, the ATM Forum LAN emulation, and the IETF (Internet Engineering Task Force) Classical IP (Internet Protocol) over ATM, with frequent reference to legacy protocols.

The greatest weakness of Onvural's *ATM Networks* is that it is now two years old. Since it was written, ATM has been widely accepted as the network technology for delivering broadband. Its promise has been upheld through field trials, and work on key standards has been completed in some cases and extended in others. Perhaps most important of all, the major companies in the communications industry are now investing heavily in ATM as the basis for new and improved services. So, while the second edition of *ATM Networks* remains a valuable part of the literature, a new edition or an expanded version of the second edition would be welcome. (According to information posted at the Artech House Website, Onvural is working on a comprehensive guide to ATM signaling that is scheduled for publication in September, 1997. See http://www.artech-house.com/artprogs/detail_disp?0-89006-871-2 (http://www.artech-house.com/artprogs/detail_disp?0-89006-871-2).

Christinger Tomer (ct@coltrane.sis.pitt.edu (mailto:ct@coltrane.sis.pitt.edu)) is an Associate Professor at the School of Information Sciences, University of Pittsburgh. He teaches courses about information retrieval, digital libraries, and networked information services.

Copyright © 1997 by Christinger Tomer. This document may be reproduced in whole or in part for noncommercial, educational, or scientific purposes, provided that the preceding copyright statement and source are clearly acknowledged. All other rights are reserved. For permission to reproduce or adapt this document or any part of it for commercial distribution, address requests to the author at ct@coltrane.sis.pitt.edu (<mailto:ct@coltrane.sis.pitt.edu>).

[table of contents](#)  [ter issues \(/lita/publications/archive/ter\)](#)

About TER