



LETTERS

Letters to Visual Basic Programmer's Journal are welcome. To be considered for publication, letters must include your name, address, and daytime telephone number. Letters may be edited for form, fit, and style. Please send them to Letters to the Editor, c/o Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; fax them to 415-853-0230; or send them electronically on CompuServe to 74774.305 or on the Internet at 74774.305@compuserve.com.

COOPER'S CRITIC OFF THE MARK?

Mr. Nielson's comments [Letters, "Secondary Storage Not Filed Away Yet," *VBPJ* January 1996] on an article written by Alan Cooper were interesting at times, but fundamentally off the mark. Masked as a "deeper philosophical question," Mr. Nielson offers two false choices facing developers: should users be required to have some "basic knowledge" about the workings of their computers? Or should users be viewed as "impatient and conceptually impaired ninny's"?

Actually, there is a wide range of possibilities in attitudes toward users. In Mr. Cooper's article there is not one trace of disdain for or condescension toward users [Guest Opinion, "The Secondary Storage Dilemma," *VBPJ* September 1995]. Indeed, he is a champion of those who use our software; he respects their thought processes and aims to make software conform to them—not the other way around.

In Mr. Nielson's letter, he was unclear about what constitutes "basic knowledge" of computers. Of course, a user may educate himself to his heart's content. However, the degree of computer technical understanding possessed by a user is irrelevant to how well that user interacts with our software. Mr. Cooper's point is that in whatever form it is finally implemented, software design should not hinder the user in any way.

Finally, permit me to object to Mr. Nielson's ad hominem jab at Alan Cooper. Is Mr. Nielson serious when he writes: "Even Cooper seems to have a rather foggy understanding of the situation"? It should be

evident to anyone who recognizes the name Alan Cooper, that the understanding of file systems and PC hardware interactions is what Mr. Cooper has in extraordinary measure.

Edmund Jimenez
Phoenix, Arizona

THANKS FOR THE HELP

My first issue of the journal arrived today to both my surprise and delight. I started reading the editorial section first because I like to hear what other users really think about the content of the product. The commentary "New Kid on the VB Block" was very enlightening [*Letters, VBPJ* November 1995]. It's good to know that there are others who are just beginning their VB experience. I have only been in the IT field for a little over a year now. I am the Help Desk Coordinator for a small company who encourages my education and growth. My job incorporates supporting Windows users, Microsoft Office Professional, WP, Lotus, and maintaining our employee training program. I will also support the AS/400 and Mac users in the future and I have the opportunity to write the technical manuals for the custom programs our IT staff develops. The VB Professional Edition was purchased for me to increase my understanding of the Windows environment. I doubt that I will ever become a programmer, but this experience will make me more valuable to my employer and allow me to do my job better. Thank you for the section for beginners. The highly technical articles will also be enjoyed (though maybe not fully understood for a while), but the more exposure I gain to the many opinions and ideas, the faster I will ingest the understanding. This reference will be a much appreciated asset to my education.

Deveri Johnson
Cedar Rapids, Iowa

TIP CORRECTION

Somewhere, in the mad rush to put together as many VB4 tips as I could for your fine publication, I made a few mistakes, that I would like to point out and correct for your readers [Supplement to *Visual Basic Programmer's Journal*, "99 of the Hottest Tech Tips for VB Developers," *VBPJ* February 1996].

First and foremost, the tip titled "Determine When an App is Complete" (page 9) will only work from 16-bit VB4 (or VB3). This tip is still useful, in that it can tell you when a "child" app has completed, regardless if whether the child is a Win16 or Win32

VISUAL BASIC[®] PROGRAMMER'S JOURNAL

PUBLISHER & EDITOR

James E. Fawcette

ASSOCIATE PUBLISHER

Ellen Gould

ART AND PRODUCTION DIRECTOR

Michael Hollister

EXECUTIVE EDITORS

Matthew Carter, Frank Moncrief, Lee Thé

MANAGING EDITOR

Christine McGeever

ASSOCIATE MANAGING EDITOR

Nina Goldschlager

ASSOCIATE EDITOR

Amy Little

SENIOR DESIGNERS

Kristen Avery, Margaret Horoszko, Kimberly Bell

ELECTRONIC MEDIA DESIGNER

Robert W. Scoble

CONTRIBUTING EDITORS

Daniel Appleman, Chris Barlow, Andrew Brust
Carl Franklin, Roger Jennings, Deborah Kurata
Craig Leach, Mark Novisoff, Sam Patterson
Karl E. Peterson, Keith Pleas, Richard Hale Shaw
Jonathan Wood, Jonathan Zuck

CONTRIBUTORS

Michael C. Amundsen, Tom Campbell
Ken Fitzpatrick, Stephen R. Husak, Mark Pruett
Rama Ramachandran, Michael Templeman

TECHNICAL REVIEW BOARD

Daniel Appleman, Andrew Brust, Douglas Haynes
Ward Hitt, Deborah Kurata, Patrick O'Brien, Sam Patterson
Karl E. Peterson, William R. Vaughn, Phil Weber
Don Willits, Jonathan Wood, Jonathan Zuck

EDITORIAL ADVISORY BOARD

Richard Finkelstein, Craig Goren, David McCarter
Jim Lisiak, Bill Neville, Joe Robison

CIRCULATION

Tena Larson, *Circulation & Marketing Director*
Bill Hooven, *Circulation Manager*
Sherry Thomas-Zon, *Circulation & Marketing Associate*
Karen Koenen, *Associate Circulation Manager*
Shirley Modric, *Circulation Coordinator*

CONFERENCES

Elaine Kato, *Conference Director*
Michael Sturdivant, *Conference Coordinator*
Leslie Tighe, *Conference Coordinator*

ADVERTISING SALES

Robin Nakamura, *National Sales Manager*
Craig R. Altman, *Senior Regional Account Manager*
(Southwest, Southeast, South Central)
Cheri Poff, *Regional Account Manager*
(Northwest & North Central)
415-833-7100 or 800-848-5523
Carolyn Lighty & Andy Skurna, *Regional Account Managers*
(Northeast & Mid-Atlantic)
201-712-0044 or 800-685-2405
Gwen Rambis, *Sales Associate*

ANCILLARY PRODUCTS COORDINATOR

Andrew King

PROJECT MANAGER, ELECTRONIC PRODUCTS

Robyn K. Glennon

CUSTOMER SERVICE

Lara L. Ireland, Katie Hathaway, Regan Moore

ASSISTANT TO THE PUBLISHER

Trisha Merriam

ACCOUNTANT

Jena Siress

OFFICE ADMINISTRATOR

Connie Pulmano

DESIGN TEAM

Enigma Inc.
Michael Carr, Cynthia Engler
Linda Magyary, Kate Paddock, Lynn Sanford

FAWCETTE TECHNICAL PUBLICATIONS

James E. Fawcette
PRESIDENT

Wilson, Sonsini, Goodrich & Rosati
CORPORATE COUNSEL



SUBSCRIPTION SERVICES

For subscriber services, please send your magazine mailing label with your inquiry to *Visual Basic Programmer's Journal*, P.O. Box 58872, Boulder, CO 80322-8872; call us at 303-684-0365; or send us a fax at 303-661-1816.

CUSTOMER SERVICE

To subscribe to *Visual Basic Programmer's Journal* or the VBCD, or to register for VBITS, please call Customer Service at 800-848-5523 or 415-833-7100. You can also send e-mail to **75451.2343@compuserve.com** or post a message in the Customer Service section of the *VBPI* Forum on CompuServe.

EDITORIAL OFFICE

Fawcette Technical Publications
209 Hamilton Avenue
Palo Alto, CA, USA, 94301-2500
Phone: 415-833-7100 • Fax: 415-853-0230

LETTERS TO THE EDITOR, PRODUCT ANNOUNCEMENTS, AND BASIC HERO CANDIDATES

Please send or fax letters and product information to the Editorial Office listed above; address them to Letters to the Editor and New Products Editor, respectively. If you'd like us to consider your application for the Basic Heroes column of *Visual Basic Programmer's Journal*, please send us a brief description of the application along with your phone number and e-mail address; we will contact you if we decide to profile your application.

As an alternative to contacting us by mail or fax, you can also send letters, product information, suggestions for Basic Heroes, and other article ideas to **74774.305@compuserve.com** or post them in the Talk to Editors section of the *VBPI* Forum on CompuServe.

VBPI DEVELOPMENT EXCHANGE ON THE WORLD WIDE WEB

Through the *VBPI* Development Exchange, located at <http://www.windx.com>, you can subscribe to *Visual Basic Programmer's Journal* or the VBCD; register for VBITS; order books; download source code that accompanies articles in *VBPI* as well as sample applications; and browse resources for Windows developers, including listings of add-on tools for Windows developers and links to other software-industry Web sites.

VBPI FORUM ON COMPUSERVE

Typing "GO WINDX" using WinCIM will bring you to the *VBPI* Forum on CompuServe, where you can post questions about Visual Basic programming and get them answered by expert section leaders as well as other members of the Visual Basic community. You can also post messages for our customer service representatives, our conference staff, and the editors of *Visual Basic Programmer's Journal*. Code that accompanies articles in *VBPI* is available for download in the Magazine Library.

VBPI SITE ON THE MICROSOFT NETWORK

"GO WINDX" also takes you to our site on the Microsoft Network, where you can read about development tools that enhance Visual Basic and Visual C++, and send e-mail to the editors or customer service representatives.

REPRINTS AND PERMISSIONS

For all *Visual Basic Programmer's Journal* editorial and advertising reprints, contact Andrew King at our Editorial Office listed above or by e-mail at **74737.2202@compuserve.com**. If you would like to quote from an article, please specify the issue date and title of the article, the portion you would like to quote, and the purpose. You may also request permission to photocopy portions of *VBPI* for internal or personal use.

app. This was my original intent, but during my own editing, I rewrote the text incorrectly. My profound apologies for the error.

Secondly, I need to reveal my sources. A great many of my tips were shared by other VB4 developers who participate in online forums with me. Special thanks goes to Bruce McKinney, who shared a number of tips while writing his book, *Hardcore Visual Basic*, and to the MVPs of the MSBASIC Forum on CompuServe.

*A. Nicklas Malik
received on CompuServe*

THE GOOD, THE BAD, AND THE UGLY

Having used VB 3.0 for about a year in the large industrial manufacturing company where I work, the good, the bad, and the ugly are beginning to show. The good is self evident as the GUI applications are enthusiastically embraced by end users. VB is a fine product that appears capable for developing a wide array of applications from a simple desktop application to an enterprise-wide, industrial-strength client/server front end.

The bad is emerging in the IT masses. VB has a large learning curve once you get beyond a simplistic application. As John Vaughn points out, "VB is really nothing more than a conventional programming language with a hellishly powerful development environment slapped on the front" [Guest Opinion, "The True Cost of Code Maintenance," *VBPI* December 1995]. In fact, VB is a tedious, syntactically crude language that requires at least a few third-party add-ons to make it really robust. Please don't mention VB 4.0. The rationale for going with VB 3.0 was to rapidly develop GUI applications that contribute business value (productivity, profit, competitive advantage), not for the "excitement" of moving to a new version a year later. Version changes are costly to big corporations. Rollouts are bad too. It's a constant struggle getting the correct DLLs, VBXs, and API calls to work on desktops, laptops, and networks.

The ugly remains on the horizon, and it appears ominous. Again as Vaughn points out, maintenance of large-scale VB systems appears to loom as a nightmare requiring potentially huge manpower commitments to support complex systems. Microsoft is a good organization with a lot of smart folks, but they need to turn their high-powered intellect to developing tools that are more friendly to the IT professionals. Then IT professionals can get a good night's sleep just like end users.

*Darryl Miersch
Pittsburgh, Pennsylvania*

INSTALLATION WORKAROUND

I ran across an interesting problem trying to install VB 4.0 Enterprise Edition in the Windows 3.1 environment. The new ver-

sion of VB makes use of the Registry database which in Win 3.1 is limited to 64K.

As a computer consultant with a big six firm, I have an extensive list of software installed on my laptop to work in various client environments. I had previously installed Microsoft Office, MSProject, and the standard Lotus Suite software. The REG.DAT file was apparently just about maxed out. When I installed VB 4.0, I got no indications of errors during setup, but when I tried to run VB and use certain custom controls, I got "License" errors.

A call to Microsoft verified the problem and the workaround:

1. Manually remove VB 4.0.
2. Manually remove all OCXs and OCAs from \windows\system.
3. Manually remove OC25.DLL from \windows\system.
4. Rename REG.DAT to REG.OLD.
5. Remove all items from the Windows Startup Group.
6. Remove all programs auto started in the WIN.INI file with the "Load" or "Run" sections.
7. Remove all TSRs from the AUTOEXEC.BAT file.
8. If you are running a compressed drive (I was), you must free up 6 megs of space in a noncompressed volume.
9. Exit Windows and reboot.
10. Start up Windows and reinstall VB 4.0.
11. Restore your system settings.

I know of one other consultant in our firm, who does not have the extensive software list that I have, who also experienced the same installation problem.

It appears that Microsoft gave little thought to developers who must still work in the Windows 3.1 environment. Most of the clients I have talked to are not planning on migrating to Windows 95; they might move to Windows NT when they can justify the cost. I am suggesting that my clients give Borland's Delphi a good look or stick with VB 3.0 until such time that they migrate to the 32-bit environment of their choice.

*Jim Gilligan
received on CompuServe*

ON DOEVENTS AND TIME SLICES

This letter is regarding your User Tip found on page 54 of the January 1996 issue, "DoEvents() and Preemptive Multitasking." It is stated, correctly, that Windows 95 is a preemptive multitasking system, and therefore calls to DoEvents() are no longer required. This is completely true... for 32-bit VB 4.0 applications. However, it is important to understand that all 16-bit applications are one "task" for the purposes of time-slicing. As a result, if your 16-bit VB applications do not call DoEvents(), they will prevent any other 16-bit tasks from

getting any time slices.

Worse yet, the Windows 95 UI itself uses a fair amount of 16-bit code, and will not respond if your 16-bit app is tying up the system. The net result is that if your 16-bit VB 4.0 apps do not call `DoEvents()`, they will lock the system.

*George Mealer
San Francisco, California*

OPTIMAL FEATURE SUITE

I really enjoyed the articles on optimization in the January 1996 issue of *VBPI*. The

articles were well written and well presented with interesting graphics. I liked the “buttons” at the top of the first page of each article that showed the article being read. Clever.

The best tip I got from the optimization articles to increase performance of my 32-bit VB application is to change it back to a 16-bit application! Really, I’m not kidding. My 32-bit application is noticeably slower than the 16-bit app I converted it from, particularly in the time it takes to display forms.

The people developing VB would be

wise to come out with a truly compiled version in 1996 or they are going to lose a lot of developers to C++, Delphi, and PowerBuilder.

*T.N. Turner
Glenwood City, Wisconsin*

RANDOM NUMBER?

I had a question regarding Carl Franklin’s `IsFileEmpty` function in Listing 1 of the Q&A column in the December 1995 issue of *VBPI*. To the constant `nBlockSize` he had assigned the value 8192. He had also mentioned using VB4 (32-bit) to change this to larger value. I was curious about knowing why he chose 8192 and why a larger value would be optimum for VB4 (32-bit).

*Rajiv Agarwal
received by e-mail*

I selected 8192 for no particular reason other than it’s big enough to be fast when reading big files and small enough to handle small files quickly as well, and also because I have always used 8K file buffers for smallish files (less than 500K). The larger the buffer the faster the data loads, and the more RAM is used. Data loads faster because the disk is accessed less. The operating system can load a big chunk of data just as fast as a small chunk. Moving the heads to the appropriate location (seek time) is what takes the most time.

It’s a personal preference of mine to use an 8K buffer when reading small files under 16-bit Windows just because I’m used to it. With the processing power of computers today, disk-access time for reading data is not as big a concern as it was not too long ago, when using a 500-byte buffer was much slower than, say, a 4K or 8K buffer. Since there is more emphasis on conserving RAM in 16-bit Windows, it makes sense to use an 8K or 16K buffer... to me anyway.

When using a 32-bit operating system like Windows 95 there is less emphasis on conserving RAM, so therefore you might shave off a few nanoseconds of access time by using a larger buffer... say 32K or even 64K.—C.F.

CHECK YOUR WINDOWS MESSAGES

Excellent article by Jonathan Zuck on using API calls and the text box examples [Windows Programming, “Send a Message,” *VBPI* December 1995]. I have for weeks been trying to interface a text box to a DOS-based mainframe interface, and could not get over the hurdle of how Windows was handling tabs, CRLF, and so on so that I could limit the text box to a maximum of 80 characters per line, and yet still store the contents as an ASCII file. After I read the article, I looked into other API calls associated with text boxes. In the end, the answer is real simple... include the CR-CR-LF soft breaks in the file, and reduce the size of the text box if any of the lines

plus tabs are longer than 80 characters! One of the reasons this was so obtuse is that most text editors (DOS Edit included) convert all the tab-stop characters, Chr(09), to a number of spaces (ASCII 20), and if you save or it auto-saves, the tabs will become overwritten in the file, and will be lost. I do not understand why Microsoft would not expose these to VB3 or VB4!

Articles such as this, with real working code examples, are by far the most valuable parts of *VBPJ* to me. Keep up the good work, and more articles like this on programming techniques.

R.J. Falkiner
received on CompuServe

OCX CONTEST WINNERS

Congratulations to the winners! In the December 1995 issue of *Visual Basic Programmer's Journal* we challenged our readers to identify the icons of 231 OLE controls... no easy task. Even if you didn't submit your entry, we hope you had fun in the attempt. Two readers scored 100 percent: Issie Chaimovitch of North York, Ontario, Canada, and Charles Rosen of Willowdale, Ontario, Canada. These two top-scorers received various prizes donated by the vendors whose products were represented in the contest, including copies of Microsoft BackOffice. The other winners are: Hal Broadhead of Meridian, Indiana; Stephen L. Hussey of St. Louis, Missouri; Jorge Vargas de la Cruz of Lima, Peru; Dean C. Arno of Cincinnati, Ohio; Steve Lehrfeld of Cranbury, New York; Lucio F.P. de Oliveira of Dover, New Hampshire; Dave Keighan of Victoria, B.C., Canada; and Vardim Rapp of Chicago, Illinois.

We did receive one high-scoring entry without a name or address on it. If you think the entry might be yours, please contact us at 74774.305@compuserve.com. Please also feel free to write us with feedback about the contest: whether it was too difficult, too easy, and whether you'd like to see a similar contest in the future. The answers appear below.—*Eds.*

1. A9
2. I8
3. G16
4. A10
5. B9
6. E6
7. G-H 11, F-G 12
8. G14
9. G13
10. H13
11. C17, D 16-17, E 16-17
12. K8
13. D7
14. D8
15. D13
16. E 11-12, F11
17. B10, C9
18. A11-D11, A12-D12
19. K6
20. A 7-8, B7
21. H12

- | | | | |
|-----------|-----------------------|---------------------|-------------|
| 22. F14 | 36. F16 | 50. A-C 13-15, D14 | 64. J8 |
| 23. F13 | 37. F17 | 51. I-K 11-12 | 65. C6, D6 |
| 24. A16 | 38. H 14-15, I 14-15 | 52. I13 | 66. J15 |
| 25. C10 | 39. C7 | 53. I10 | 67. C8 |
| 26. D9 | 40. F10 | 54. E15 | 68. C16 |
| 27. A17 | 41. J7 | 55. F15 | 69. K16 |
| 28. G-I 6 | 42. G9 | 56. G15 | 70. H17 |
| 29. K7 | 43. G10 | 57. J9 | 71. J16 |
| 30. D10 | 44. I-K 17, A-K 18-21 | 58. F6 | 72. B17 |
| 31. D15 | 45. J6 | 59. J-K 13, K 14-15 | 73. B16 |
| 32. E9 | 46. H9 | 60. I16 | 74. B8 |
| 33. E10 | 47. H10 | 61. J14 | 75. E 13-14 |
| 34. G17 | 48. E-I 7, E-H 8 | 62. H16 | 76. K9 |
| 35. F9 | 49. I9 | 63. J10 | 77. K10 |