

Contents

Introducing VB Tech Journal

Welcome to the VB Tech family!

Product Reviews

Comparative reviews

Anatomy of a comparative review

Single-product reviews

Opinions and bugs

Anonymity--a vital resource

Becoming a reviewer

Articles that teach technique

How-to articles

1995 Editorial Calendar

VB Tech Journal Editorial Calendar

How the process works

Contacting the editors

Moving into production

The fine print

A few tips

Welcome to the VB Tech family!

VB Tech Journal is a monthly magazine for professional software developers who use Microsoft Visual Basic to write Windows applications for their customers, clients, and employers. VB Tech Journal's mission is to help Visual Basic programmers identify the very best add-on products and show them how to use those products most effectively to solve development problems.

VB Tech is a serious magazine but the presentation style is light, lively, and fun. Most magazines for software developers read as if the editors are so impressed with the gravity and difficulty of programming that they treat the topic in the driest possible way. At *VB Tech Journal*, we agree that software development is demanding work. But it's also a mental challenge and we think most Visual Basic programmers find it fun. Our attitude is simple: If you don't like programming, then maybe you should quit. The world needs somber IRS auditors and divorce attorneys.

This document outlines the kinds of articles we are looking for in *VB Tech Journal* and provides what we believe to be reliable advice about how to contact the editors and get your work into print.

VB Tech Journal purchases three kinds of articles: comparative product reviews, single-product evaluations, and how-to articles covering development techniques. This document includes guidelines for all three kinds of work.

Comparative reviews

The purpose of a comparative review is to help readers select a product to purchase from a group of several products that perform similar tasks. VB Tech Journal is committed to being the Visual Basic community's best source of 100% objective, 100% accurate comparative product reviews. Every review is based on actual hands-on experience with the products and thoughtful analysis of their features.

Refer to the sample outline for a reliable guide to the construction of a *VB Tech Journal* comparative review. Please follow this outline closely. We intend for every comparative review we publish to be based upon this framework.

Section I ("all about feebledreizers" in our sample outline) may take up 50% of the total length of the review. It starts with a statement of a programming problem and introduces the technology that addresses the problem. An explanation of the underlying technology comes next. It might be appropriate to include sample code with this section of the review.

Section I.C. ("the features that distinguish feebledreizers from each other") is the heart of the review. Your testing of the products should have resulted in a list of features or attributes. This section of the review is an evaluation of the features, not the products. You might write a paragraph like this:

Some grid controls allow developers to specify inalterable cell attributes at design time. A particular cell might be designated read-only, for instance. Another cell might be set to interpret and display its contents as a date in MM/DD/YY format. This capability generally comes at a price: The programmer who wishes to set the attributes of one cell at design time must go through the tedious process of setting the attributes of all visible cells. This probably isn't worth the effort if you're using the grid as a database browser or a little spreadsheet scratchpad—you can always set and change cell attributes at run-time, after all—but it can be a real lifesaver if you're using the grid to display custom data in read-only form.

Discuss all the features your testing led you to discover, concentrating specifically on those that are unique to the kind of product you are reviewing. General features such as speed, memory requirements, price, documentation, and so on are probably less important and less interesting, depending on the type of product you are evaluating.

Section I concludes with an optional high-level view of the products. In a review of grid controls, for example, the reviewer might note that the controls come in three general types: simple grids, "bound" grids that work with databases, and spreadsheet controls that parse and execute mathematical expressions typed into cells. The reviewer's task is to identify "feature clusters" that distinguish the groups of products from each other. Features that are desirable in a spreadsheet control may be a waste of memory and disk space in a simple grid.

You may even find that you can partition the products along two axes. There might be a natural break between expensive and inexpensive products, for instance, as well as a division between those that implement *fetch* and those that don't.

The high-level overview is often the most valuable part of a comparative review. We call it optional only because some products are so similar that it's not practical to divide them into groups. Use all your mental resources and try to provide the logical groupings. They make reviews much better.

At this point, the text of the review should be about half done. Don't be concerned if Section I takes

several pages. It's our way of sneaking tutorial content into an article about products. Laying a solid foundation in this way definitely makes the subsequent discussion of individual products clearer and more valuable.

Section II of the outline is organized product by product. (Note that **no product is to be mentioned at all until you get to Section II**. Section I is about technologies and features, not products.) For each product, write an overview that explains which of the features from Section I the product implements, and comments upon the implementation of specific features if the implementation is notable. If you have found a way to divide the products into logical groups based on the features they implement, then cover all of one group before moving to the products in the next group. The last part of Section II can be devoted to products that don't fit well into any of the logical groups if you'd like.

Section III is very short. It recaps the trade-offs that lead to feature clusters, telling readers how they can assess their own programming priorities and match them up with groups of features and products. If there are clear winners and losers in any of the groups—including products that are competitive technically but a superior value by virtue of price, support, vendor reputation, or documentation—then by all means identify them in Section III.

The outline covers the narrative of your review. We require other elements as well.

For each product, we publish a small box containing vital information: the product's (correctly spelled) name and version number, vendor information (including support policy, address, phone, fax, and e-mail contact data), the product's price, and two or three sentences summarizing the features that distinguish the product from the others. In addition, we will ask you to create BMP files containing screen captures of the product in action. Our preferred format for BMP files is 800 x 600 x 16 or [active window size] x 16. If you are having trouble creating screen-capture BMP files, get in touch and we will help you out or arrange to make the BMP files ourselves.

Anatomy of a comparative review

SIX HIGH-POWER FEEBLEDREIZERS

I. All about feebledreizers

- A. The problem space
 - 1. What is the programming problem?
 - 2. How do feebledreizers solve the problem?
- B. The technology
 - 1. What is the underlying technology?
 - 2. How do feebledreizers work?
- C. Features that distinguish feebledreizers from each other
 - 1. Blatchness--what it is
 - a. Good or bad feature?
 - b. When is it most important?
 - 2. Fretch--what it is
 - a. Good or bad feature?
 - b. When is it most important?
 - 3. Limpitude--what it is
 - a. Good or bad feature?
 - b. When is it important?
 - 4. et cetera
- D. Logical groups of products--an overview
 - 1. High blatch vs. high fretch
 - a. Some products emphasize blatch. They tend to cost more and work more slowly.
 - b. The other products emphasize fretch. They cost less and tend to be more reliable.
 - 2. High limpitude vs. low limpitude

II. How the products measure up

- A. Group A--the high-blatch feebledreizers
 - 1. Product 1
 - a. introduction
 - b. Product 1's blatchness
 - c. Product 1's fretch quotient
 - d. Product 1's limpitude
 - 2. Product 2
 - 3. et cetera
- B. Group B--the high-fretch feebledreizers
 - 1. Product 1
 - a. introduction
 - b. Product 1's blatchness
 - c. Product 1's fretch quotient
 - d. Product 1's limpitude

2. Product 2

3. et cetera

III. Which feebledreizer is for you?

A. Trade-offs in feebledreizer design

B. Matching your needs and the product
feature profiles

C. Clear winners and losers, if any.

Single-product reviews

The purpose of a single-product review is to measure a product's utility in meeting its purpose. The review may highlight the hours of work a product might save the prospective purchaser, or it may focus on the capabilities the product adds to the Visual Basic environment.

Like a comparative evaluation, the single-product review begins with a programming problem. If the problem is simple or universal, the problem may be stated briefly. (*"Did you ever wonder why command buttons always have gray faces? Did you ever wish you could replace the featureless gray button face with a bitmap or an animated series of bitmaps?"*) Complicated problems (such as the component-versioning problem addressed by Desaware's VersionStamper-VB, for instance) may require substantially more space.

Next, introduce the product as a potential solution. The balance of the review should be devoted to the ways in which the product solves the problem, plus any interesting side-notes that present themselves during testing. You may find, for instance, that the product solves the stated problem but creates new problems. Or the performance overhead imposed by the cure may be worse than the disease. The product may solve additional problems...it may even do a terrible job of curing the problem it was intended to solve while presenting a good solution to a different problem altogether.

Opinions and bugs

Most people believe that published product evaluations are based on reviewers' opinions about products. We believe this is not true of good reviews. **We do not allow opinions in VB Tech Journal reviews.**

However, we do expect you to make evaluative statements. Consider this example:

LostAt C is a terrible C compiler. It won't process #include files unless they're in the same directory as the main source file, it sometimes deletes source files during compilation, and it generates the largest, slowest executables of any compiler we tested. LostAt C accepts neither K&R nor ANSI C, requiring instead a bizarre proprietary variant. The debugger flags working code as questionable and incorrectly reports register contents. Yes, this is the least expensive of the products we tested...but it is no bargain.

In our view, this paragraph does not include a single opinion. The passage begins with an assertion of fact: "LostAt C is a terrible C compiler." This assertion is immediately followed by supporting factual detail. Based on this factual evidence, we are content to consider "terrible" a straightforward statement of truth, not an opinion.

Remember this example when you are tempted to inject an opinion into a *VB Tech Journal* review. Rephrase your opinion as a statement of fact and follow it up with supporting factual detail. A review written in this manner is dramatically more valuable to readers than an opinion-based review could ever be.

Another common misconception is that the purpose of a review is to discover and report the number, type, and severity of bugs in the tested products. This is an admirable goal, but it is not the main purpose of *VB Tech Journal's* reviews.

If you encounter a bug during testing, the first thing to do is to ensure that it is repeatable and not a random, one-time event. The Windows environment is complicated, especially when you are loading and removing multiple programming tools in quick succession. Odd behavior may be due to the interaction of your video driver, a flaky (but often benign) DLL in your word processor, and a once-in-a-lifetime global-memory fragmentation pattern.

If you can establish that the behavior is repeatable and you believe you have traced it to a product you are reviewing, contact your *VB Tech Journal* editor immediately. **Do not report the bug to the vendor!** Your editorial contact at **VB Tech** will talk with the vendor to find out if the bug is known. We may acquire a bug-fix release for you.

Anonymity--a vital resource

Whether you are writing a comparative review or a single-product evaluation, we request that you refrain from disclosing your identity to the vendor. It is common for vendors to try to contact reviewers, to offer them special help, to provide uncommon levels of support, even to try to influence the content of the review. The editors of VB Tech Journal will shield you from vendors by maintaining your anonymity. We will never disclose your name to vendors until the review appears in print with your byline.

One of the ways we preserve your anonymity is by handling all communications with vendors during the review process. The editors will request that products intended for review be shipped to us here at Oakley Publishing Co. We will then forward the products to you. Note that vendors sometimes keep track of the serial numbers of the products they send to magazines. If you are calling to assess the quality of the vendor's tech support, do not give your name or the product's serial number.

Becoming a reviewer

The editors of VB Tech Journal assign reviews to professional developers who have indicated a willingness to evaluate products. The best way to become a reviewer is to jot us a short note outlining your willingness to write reviews, your interests, and some information about your background and areas of particular expertise. We'd also like to know where you work so we can assess potential conflicts of interest in assigning product evaluations.

VB Tech Journal never accepts unsolicited reviews. The only way to get a review in the magazine is to have the evaluation assigned to you by the editors.

Use this skeleton to structure your review

SIX HIGH-POWER FEEBLEDREIZERS

- I. All about feebledreizers
 - A. The problem space
 - B. The technology
 - C. Features that distinguish feebledreizers from each other
 - D. Logical groups of products—an overview
- II. How the products measure up
 - A. Group A—the high-blatch feebledreizers
 - B. Group B—the high-fretch feebledreizers
- III. Which feebledreizer is for you?
 - A. Trade-offs in feebledreizer design
 - B. Matching your needs and the product feature profiles
 - C. Clear winners and losers, if any.

How-to articles

VB Tech Journal publishes tutorial articles in addition to product evaluations. How-to articles explain techniques for accomplishing specific programming tasks. We are interested in publishing how-to articles for novices, intermediate programmers, and wizards. In our experience, even Visual Basic wizards have gaps in their knowledge that make them novices in specific areas.

The editors of *VB Tech Journal* accept only a small fraction of the many unsolicited manuscripts they receive. Prospective authors put a lot of work into articles that, for one reason or another, do not quite meet the magazine's needs. This needn't be the case. As a potential contributor, you can stack the deck in your article's favor by applying a few simple techniques.

You've already taken the first step by acquiring these guidelines. They spell out, in broad terms, the kinds of articles *VB Tech Journal* is looking for. They explain (keep reading!) how to submit your work to the editors.

You should also study the magazine's editorial calendar, which outlines the topics that each issue of *VB Tech Journal* will focus on for the coming year. Many otherwise publishable articles are rejected with regrets due to bad timing. Deadlines for each issue are approximately 12 weeks before the cover date. Articles for the June issue, for instance, enter production in the beginning of March.

Like other programming magazines, *VB Tech Journal* publishes source code. Readers like code and understand it; a short listing is often the most effective way to illustrate a concept. However, a listing is very rarely the central focus of an article. *VB Tech Journal's* editors prefer short code samples that illustrate concepts explained in the text, not short articles that explain how lengthy program listings work. This is an important distinction.

Because Windows is a graphical environment, and because new visual programming tools substitute a "drag & drop" development process for the familiar "lines-of-code" model, screen captures are likewise valuable. The editors prefer 16-color BMP files for screen captures.

You are certainly free to write an article and submit it to the editors for consideration. If you've followed these guidelines and selected a topic from the editorial calendar, odds are good that *VB Tech Journal* will publish your work. But you can increase your article's chances further by submitting a one-page proposal or outline to the editors before you begin writing. The editors will respond by assigning you the article as proposed, declining it if the topic doesn't meet the magazine's foreseeable needs, or recommending adjustments to your outline. This is the very best way to get your work into *VB Tech Journal*.

Once your proposal is accepted, the editors will assign you a deadline and save a spot for your article in an upcoming issue.

VB Tech Journal Editorial Calendar

Issue	Theme	Product Focus
Premier 1995	Visual Basic 4.0	Grid Controls
March 1995	OLE 2	Setup Toolkits
April 1995	Graphics	Business Graphics Controls
May 1995	Group Development	Version-Control Systems
June 1995	User Interface Design	Serial Communications Toolkits
July 1995	Client-Server	Pretty Printers & Browsers
August 1995	Debugging	CASE Tools
September 1995	Mixed-Language Programming	Utilities
October 1995	Optimization	Interface Widgets
November 1995	Multimedia	Network Tools
December 1995	<u>Windows</u> Internals	Product of the Year Awards

Issue themes refer to how-to articles. The product focus column shows what kinds of products we will review in each issue.

Contacting the editors

Our editors love to work with authors and help them develop articles. They'd like nothing better than to meet you for coffee or indulge in lengthy telephone consultations to talk over your ideas.

Alas, this is simply not feasible. The editors travel a great deal and their time is in high demand when they are in the office. They cannot spend the time they would like discussing your ideas with you.

Your best bet is to reach the editors via electronic mail. VB Tech Journal's electronic home is SDFORUM, the Software Development Magazine forum on CompuServe. You can reach editor J.D. Hildebrand at 76701,32. You can also reach Hildebrand via the Internet: 76701.32@compuserve.com. Or send a fax: (503) 746-0071.

The editors will reply to your proposal within a few days. If you haven't had a response within three or four weeks, send a follow-up message. Even in these enlightened electronic days, messages sometimes go astray.

Moving into production

Once your article is complete, it will begin an arduous and unlikely journey that results in a printed magazine. The journey starts when you transmit the article to the editors. While VB Tech Journal is happy to accept your text electronically, the editors also require a hardcopy. This ensures that figures are reproduced accurately and indentation levels are maintained in source code listings.

Toss a disk containing your article into the envelope with your hardcopy and mail it to the editors at *VB Tech Journal* Submissions, P.O. Box 70167, Eugene, OR 97401-0110. If you're sending a parcel or an overnight letter, address it to us at 150 N. Fourth St., Springfield, OR 97477.

Your article will be reviewed and edited on PCs, then "poured" into our desktop-publishing system for pagination. At this point the editors will print a set of galleys and send them to you via Federal Express or facsimile. (For this reason, we need to know your fax number and your street address, not a P.O. box.)

Your job is to check the galley's for errors of fact and mistakes that may have crept in during editing.

Please check the text and listings carefully to ensure that they appear as you intended.

When you have checked the galley's, telephone the editors and report your findings.

The fine print

VB Tech Journal is pleased to be able to offer payment for articles. While this honorarium is rarely realistic compensation for the hours contributors spend on their work, it is sincerely heartfelt and generally competitive with the payment offered by other magazines.

If you wish, you can negotiate payment when you propose your article. Otherwise, you will receive an honorarium at our usual rates—usually about \$75 per published page of text (not counting advertisements and artwork that may appear on the same pages as your article). You generally get to keep any software products you review for us. *VB Tech Journal* pays somewhat more for articles that require specialized knowledge, that are delivered on time despite a short deadline, or that come from frequent contributors. Typical author payments are \$300 to \$600.

VB Tech Journal routinely purchases "all rights" to articles, which means that the editors reserve the right to reprint or republish the work at a later date. The editors will happily modify this standard practice upon request; it is not *VB Tech Journal's* intention to deprive authors of rights to their work.

Code published in the magazine (and supplemental listings provided by authors but not printed in the magazine) may be made available to readers on disk, on our CompuServe forum, and via the magazine's network of private BBSs. We are not lawyers, but we have consulted lawyers and arrived at this conclusion: Once you have published source code in any magazine, readers may use the code in their own projects without restriction. This is true no matter what copyright or other restrictive statements you embed in the code, no matter what assurances the magazine's editors give you to the contrary. Here's a commonsense rule: **If you don't want others to use your code, don't publish it in a magazine that's read by tens of thousands of programmers.**

A few tips

Don't worry too much about dangling participles or proper use of the subjunctive. *VB Tech Journal* employs a staff of professional editors whose joy it is to polish the rough edges in contributed articles. Your important contribution is the content of the article or review. Write conversationally, erring on the side of clarity.

VB Tech Journal prefers articles in the 2,500 to 3,500 word range. It's difficult for the editors to find room for an article that exceeds 4,000 words in length, even if it is a very fine article. Keep listings to 100 lines or so. Many articles suffer from over-ambition. It's better to cover a limited topic fully than to describe a very broad topic superficially. If you find yourself approaching 4,000 words, consider tightening your article's focus.

Comparative reviews may run from 3,000 to 10,000 words, depending on the number of products and the depth of the evaluation. Your editor will work with you on a target length.

The most reliable guide to the editors' tastes is *VB Tech Journal* itself. The best contributors are those who read the magazine.

Glossary of Terms

anonymity

calendar

comparative review

conflicts of interest

copyright

dangling participles

deadline

documentation

features

figures

galleys

IRS auditors

J.D. Hildebrand

magazine

novices

Oakley Publishing Co.

opinion

over-ambition

overnight letter

professional software developers

program listings

proposal

SDFORUM

single-product review

subjunctive

tech support

VB Tech Journal

Windows

anonymity

We protect your anonymity during the review process so there is no chance vendors will contact you. This makes your life easier and ensures that vendors cannot influence the review process.

calendar

The editorial calendar lists the topics that will be covered in future issues of the magazine.

comparative review

A comparative review is an evaluation of several products, all designed to perform the same task. The reviewer's task is to help readers understand which products will best meet their specific needs.

conflicts of interest

The editor's responsibility is to serve only the readers' interests. A conflict of interest occurs when the editor or reviewer is subjected to temptation. If you owned stock in a company and that company's product was included in a comparative review, there would be a conflict between the readers' interest in knowing the full truth and your own interest in having the vendor's products sell well so your stock's value will rise.

copyright

When your article is accepted you will receive a "Warranty of Originality and Assignment of Rights" form that expands upon these points. Basically, you must guarantee that you own the article and can sell us permission to print it, and you must outline what specific publication rights you are granting us. This form is routine but important.

dangling participles

Afraid for her life, the car turned toward her and accelerated.

deadline

Your deadline is simply your due date. Our deadlines are approximately 12 weeks before the issue cover date...but we appreciate work that comes in early!

documentation

The hallmark of a bad review: excessive space devoted to evaluation of the installation procedure and documentation. It's a sure sign that the reviewer spent insufficient time (or maybe no time at all) actually using the product!

features

Features are the capabilities of the program you are testing. Print preview is a feature, for instance. Price and availability are not commonly thought of as features, but they are.

figures

A figure may be a .BMP file containing a screen capture of a program in operation, or it may be a diagram illustrating a process or the architecture of your app. We generally redraw diagrams to match the look and feel of the magazine, so don't invest a lot of time polishing the rough edges. We do appreciate perfect screen capture files, however!

galleys

The term "galleys" refers to your article after it has been edited and converted into a desktop publishing file. We ask authors to review their work at this point to ensure that we have not introduced errors during editing. This is your last chance to look over your article, so please review your galleys carefully!

IRS auditors

You don't think we'd be stupid enough to say something disparaging about IRS auditors, do you?

J.D. Hildebrand

J.D. Hildebrand is the editor of VB Tech Journal and the editorial director of Oakley Publishing Co. Reach him on CompuServe at 76701,32.

magazine

The word "magazine" is derived from an Arabic word meaning "storehouse." Magazines are storehouses of information.

novices

We don't consider "novice" a disparaging term. Everybody's a novice at something. We publish both advanced and novice-level articles.

Oakley Publishing Co.

Oakley Publishing Company is the leading publisher of periodicals for professional Windows programmers, including *Windows Tech Journal*, VB Tech Journal, *NT Developer*, and *VC++ Professional*. Contact Oakley Publishing at P.O. Box 70167, Eugene, OR 97401, (800) 234-0386, (503) 747-0800, fax (503) 746-0071.

opinion

Opinions are not allowed in VB Tech Journal reviews.

over-ambition

"How to Write Good Visual Basic Programs"

overnight letter

Writing for us shouldn't cost you money. If you promise to clear your long-distance phone calls and overnight mailings with us in advance, we promise to reimburse you for them promptly.

professional software developers

Professional developers are programmers who (a) are paid to write software as part of their work, and (b) write software that other people use.

program listings

VB Tech Journal publishes source code listings. We prefer a two-space indentation level. Sometimes supplemental files are made available online even though they aren't printed in the magazine.

proposal

A proposal can be as short as a few sentences, or it may expand to several pages. We need to know what your idea is, why it's useful, and how much space you will need to explain it.

SDFORUM

Also known as CLMFORUM (named after the defunct Computer Language magazine), SDFORUM is the electronic home of Software Development, PC Techniques, Windows Tech Journal, Windows/DOS Developer's Journal, Embedded Systems Programming, and other publications. Check it out!

single-product review

The purpose of a single-product review is to help readers effectiveness of a product at performing its intended task.

subjunctive

If he continues to eat this way, by Christmas he will have become fat!

tech support

The ideal review does include an assessment of vendor support, but preserving your anonymity is more important than evaluating support.

VB Tech Journal

VB Tech Journal is a monthly magazine for professional software developers who use Microsoft Visual Basic.

Windows

In this document, "Windows" refers to a member of the Microsoft Windows family of operating environments: Windows 3.x, Windows 95, and Windows NT.

To subscribe, contact Oakley Publishing Co., P.O. Box 70167, Eugene, OR 97401, (800) 234-0386, (503) 747-0800, fax (503) 746-0071

Visual Basic is a Microsoft product, but VB Tech Journal is a completely independent publication. Oakley Publishing Co. is not affiliated with Microsoft Corp.

Opinions are not allowed in VB Tech Journal reviews.

An assertion of fact is an opinion stated as fact and supported by factual detail.

The vendor is the publisher of the software you are reviewing.

The editor's responsibility is to serve only the readers' interests. A conflict of interest occurs when the editor or reviewer is subjected to temptation. If you owned stock in a company and that company's product was included in a comparative review, there would be a conflict between the readers' interest in knowing the full truth and your own interest in having the vendor's products sell well so your stock's value will rise.

Different publications have different philosophies about editorial calendars. Some editors devote every word of the issue to the topic on the calendar. We prefer to reserve space for 1 to 3 articles related to the issue theme, and to fill the rest of the issue with the best, timeliest articles available, even if they are not related to the topics listed on the calendar.

VB Tech Journal's electronic home is VB Tech Online: CompuServe's CLMFORUM, section 9. You'll find the authors there as well as all the code to accompany each issue of the magazine.

Contact VB Tech Journal at (800) 234-0386 or (503) 747-0800. Outside working hours, we answer the "spouse line": (503) 746-0070. Or fax us at (503) 746-0071.

VB Tech Journal pays upon publication, which means you receive your check along with several copies of the issue that contains your article or review.

