NET Gotchas: 75 Ways to Improve Your C# and VB.NET Programs, Venkat Subramaniam, O'Reilly Media, Inc., 2005, 0596009097, 9780596009090, 372 pages. Like most complex tasks, .NET programming is fraught with potential costly, and time-consuming hazards. The millions of Microsoft developers worldwide who create applications for the .NET platform can attest to that. Thankfully there's now a book that shows you how to avoid such costly and time-consuming mistakes. It's called .NET Gotchas. The ultimate guide for efficient, pain-free coding, .NET Gotchas from O'Reilly contains 75 common .NET programming pitfalls--and advice on how to work around them. It will help you steer away from those mistakes that cause application performance problems, or so taint code that it just doesn't work right. The book is organized into nine chapters, each focusing on those features and constructs of the .NET platform that consistently baffle developers. Within each chapter are several "gotchas," with detailed examples, discussions, and guidelines for avoiding them. No doubt about it, when applied, these concise presentations of best practices will help you lead a more productive, stress-free existence. What's more, because code examples are written in both VB.NET and C#, .NET Gotchas is of interest to more than 75 percent of the growing numbers of .NET programmers. So if you're a .NET developer who's mired in the trenches and yearning for a better way, this book is most definitely for you.

DOWNLOAD http://archbd.net/IrA2RK

ADO.NET Cookbook, Bill Hamilton, 2003, Computers, 605 pages. Designed in the highly regarded O'Reilly Cookbook format, ADO.NET Cookbook is strikingly different from other books on the subject. It isn't bogged down with pages of didactic.


The Domain Book of Intuitive Home Design How to Decorate Using Your Personality Type, Judy George, Todd Lyon, 1998, , 192 pages. An innovative new approach to home design explains how to incorporate the four personality archetypes--Visionary, Artisan, Adventurer, or Idealist--into an individualized.


A Manual of American Ideas Designed 1st. For the Use of Schools. 2nd. For the Instruction of Foreigners Seeking Naturalization. 3rd. For the Use of Citizens, Caspar Thomas Hopkins, 1873, United States, 382 pages.


No fluff, just stuff anthology the 2006 edition, Neal Ford, Scott Davis, Jun 13, 2006, Computers, 240 pages. Presents a collection of articles covering such topics as web services, DSLs, Shale, legacy code, and CSS.

Picture Books Plus 100 Extension Activities in Art, Drama, Music, Math, and Science, Sue McCleaf Nespeca, Joan B. Reeve, 2003, Education, 133 pages. The idea of literature extension--using hands-on projects from math, science, music, drama, and art to expand on literary works--is an increasingly popular teaching method.

C# & VB.NET Conversion Pocket Reference, Jose Mojica, 2002, Computers, 139 pages. Designed
for quick and easy access for developers working with both C# and Visual Basic.NET, a compact reference explores the similarities and differences between the two ....

The net power structure in ten villages, Bangladesh Rural Advancement Committee, 1980, Political Science, 113 pages. .

Visual Basic .NET Codemaster's Library , Matt Tagliaferri, Apr 4, 2002, Computers, 376 pages. Microsoft® D.â„¢s new .NET version of Visual Basic is its most radical revamping of the product since it was first released. Even if you know your way around VB6, the latest version ....

C# developer's headstart , Mark Michaelis, Philip Spokas, 2001, , 225 pages. An overview of C# programming basics compares C# to C++, Java, and Visual Basic.NET while covering polymorphism, threading, garbage collection, inheritance, .NET framework ....

Visual Basic 2005 in a Nutshell , Paul Lomax, Ron Petrusha, Feb 6, 2006, Computers, 746 pages. The classic Nutshell guide to Microsoft's Visual Basic programming language is completely revised and reorganized to cover the forthcoming VB 3005 version, as well as VB .NET 1.1 ..


Fast Track C# , K. Scott Allen, Neil Avent, Julian Templeman, Jon D. Reid, Syed Fahad Gilani, May 1, 2002, Computers, 448 pages. C# is a modern, object-oriented language, specifically designed by Microsoft to be the language of choice for programming the .NET Framework, the first development platform ....

NET multithreading , Alan Dennis, Jan 1, 2003, Computers, 360 pages. Covers topics such as using Asynchronous Delegates, controlling thread execution, handling errors, protecting data, coordinating threads, and using ThreadPools..

Like most complex tasks, .NET programming is fraught with potential costly, and time-consuming hazards. The millions of Microsoft developers worldwide who create applications for the .NET platform can attest to that. Thankfully there's now a book that shows you how to avoid such costly and time-consuming mistakes. It's called .NET Gotchas.

The ultimate guide for efficient, pain-free coding, .NET Gotchas from O'Reilly contains 75 common .NET programming pitfalls--and advice on how to work around them. It will help you steer away from those mistakes that cause application performance problems, or so taint code that it just doesn't work right.

The book is organized into nine chapters, each focusing on those features and constructs of the .NET platform that consistently baffle developers. Within each chapter are several "gotchas," with detailed examples, discussions, and guidelines for avoiding them. No doubt about it, when applied, these concise presentations of best practices will help you lead a more productive, stress-free existence.

Dr. Venkat Subramaniam (venkats@agiledeveloper.com) is the founder of Agile Developer, Inc. (a software training, mentoring and development firm), and a co-founder of DuraSoft. Working with .NET since its Beta 2 pre-release, he's used the platform extensively on commercial projects for his clients. Venkat offers several courses on developing applications using .NET, and has trained and mentored more than 3,000 software developers in the US and Europe. A frequent speaker at software development conferences, he is also an adjunct professor for the practice of computer science at University of Houston, and teaches at Rice University's School for Continuing Studies. Venkat holds a BS in Computer Engineering, an MS in Electrical Engineering, and PhD in Computer Science. He is recipient of the 2004 UH Computer Science Department Teaching Excellence award.
If I had just two words to describe this book they would be "Experience Talking". Read this only if you have played with .net for a year or more. You will recount how you were stuck in an issue and spent anything from hours to days figuring out what the right way to do it. And now it is out there for you in this book.

The real question about this book is, "What is the point?" I purchased this book based on early reviews. I encourage you not make my mistake; you should take a look at the contents first. I believe you will quickly decide you have several better book choices from Amazon. The book is a collection of essentially unrelated code snippets in C# and VB.NET, which describe how not to write code. Everything in the book is well-known and available in the C# / VB.NET online documentation. If you are specifically interested in programming mistakes then this book may be of interest to you. But if you are looking for practical coding techniques you should look somewhere else.

Most C# and VB.Net programmers know there are many quirks and unforeseen results that hide within the .NET Framework. Sometimes we think we know how something should behave, and will spend hours trying to make it work -- all the while, the actual behavior was working correctly, it was our assumptions that were incorrect.

In this book, the author takes you through 75 of the most common quirks, gotchas and pitfalls that even the most experienced .NET programmer sometimes will get stuck up on. This is not a book about errors within .NET, rather one that shows you the more common differences between perceived and expected results.

The book is organized into sections covering different areas of the framework -- from the CLR itself, to garbage collection, to the Visual Studio environment, and beyond. One of the more helpful parts about the book is the author in many places will illustrate code snippets with "gotcha" or incorrect code, and then have a sample of the correct way to code a given item to get back the intended results. In addition, a wrap-up at the end of each gotcha puts the entire thing into a nutshell-quote to make it (hopefully) easy to remember and apply.

I love O'Reilly books in general but this book has me asking what has happened there. This gotchas book is a weak entry in the O'Reilly lineup. Not so much from a technical point of view; overall the book is competent, but from a "why do I want this book"? On the positive side the book is interesting because it presents some real arcane features of the CLR, but on the negative side there is very little practical value. Like any book, check it out first to see if you will find it useful --- I did not.

This whole book is for those of us that find things that we can never get to work and now the truth is out. They really don't work. I started this by reading the compiler gotchas and that made me feel a lot better. Being new to the visual studio environment I was very unsure of my code writing since the last time I had to write code we were using simple VB ver.4 scripting so I was not very sure of what I was doing.

There are very few programming books that gives you a sense of "Oh Really ?" after reading them and this is definitely one of those. First of all, be aware that the intended audience for this book is definitely VB.net or C# programmers with experience. If you're a newbie to .Net, then you need to know some stuff before opening this book and therefore it would not be a good value addition for newbies.

Okay, I read only Chapter-5 (Garbage Collection Gotchas) in this book so far but from just reading that chapter alone, I got the sense of "Oh Really !". Rarely do you see Garbage Collection discussed in such detail in .Net books, but it is a class act in this book. The concept or Gotcha is explained very clearly without any typos and the code is presented in the book in both VB & C#. Therefore, it is highly useful for both the audience (VB & C#) and also it gives you a sense of what is possible or not-possible in the other language. For example, C# uses ~ for destructor whereas in VB you have to explicitly code the Finalize method. Also, there is no equivalent of "using" in VB although .Net 2.0 introduces it in VB. As far as Chapter-5 is concerned, I had to read the gotchas in order since they kind of build up on the previous one. But, Iam really hoping that I can just read the other Gotchas in
Are you among the many programmers who have come to appreciate how powerful Microsoft's .NET Framework can be as a platform for development? If you do, this book is for you! Author Venkat Subramaniam, has done an outstanding job of writing a great book that shares his .NET experiences with developers, to help them avoid the gotchas! Subramaniam begins by discussing the features in the CLR and the Framework that can impact the behavior and performance of your application. Next, the author focuses on Visual Studio- and compiler-related gotchas. Then, he delves into gotchas at the language and API level of the .NET platform. The author continues by discussing the issues of language operability gotchas. In addition, the author next focuses on concerns related to garbage collection, and how to write code that handles it effectively. He also discusses, the things you need to be aware of in the areas of inheritance and polymorphism so you can make the best use of these important concepts. Next, the author addresses the general problems with threading, the thread pool, asynchronous calls using delegates, and threading problems related to Windows Forms and Web Services. Finally, he focuses on details you should be aware of to make interoperability work for you. With the preceding in mind, the author has also done an excellent job of writing a book that focuses on the .NET Framework and language features that have consistently exhibited behavior that is not obvious to the programmer. So, why should you be interested in learning about these unexpected features? Because, knowing these little gotchas will help you avoid mistakes!

At my work, we have a pretty rigorous application process. Once of the most difficult aspects of our application process is the series of very difficult .NET questions, which determine the applicant's understanding of C# and the .NET CLR. If you read this book before taking our .NET test, not only will you pass, but you'll probably be able to put some of the testers to shame. This is one of the best books I've read on the under the hood aspects of the CLR. This book is filled with tips (e.g. gotcha #12 compiler warnings may not be benign), which are then followed by an explanation of what the problem is, and how it can be resolved. In the example above, the author describes how .NET will allow you to do things that you probably shouldn't be allowed to do. In this particular example, the programmer used instead of , and failed to mark a method in a derived class with the override keyword. This book also does an excellent job describing some of the differences between C# and VB.NET. For example, the author illustrates that the sequence of initialization is not the same between the two languages. In terms of organization, when the author identifies a gotcha, there is a clear code example provided that exploits the problem. The author even displays a large X next to the code, so you know not to use it. After discussing how to resolve the problem, the author usually provides code that is safe to use (which is identified with a large check mark next to it). Without a doubt, this is the best book I have read on the subject of things to look out for in the .NET framework. The issues discussed in this book should be known by every .NET developer. At least read the book before your next job interview.

A nice, random access book on C# and VB.NET programming. You are expected to know the rudiments of at least one of these languages. It might be worthwhile to have this book around, as a possible quick time saver, if you're debugging. Looking over the gotchas, one sees how the ambitious scope of .NET has sometimes caused problems. In any given language, you know that certain names cannot be used for variables or classes, because they are reserved keywords. But .NET lets you mix source code from different languages, because they are reserved keywords. But say, effectively expand to include keywords in those other languages. Something you never had to worry about before. Another gotch arises due to arrays. In C#, the array size is one more than the maximum allowed index. In VB, the array is made with that maximum value. Inconsistent. The book has other items that are more intricate. Some of these might be new to you.

I buy a lot of Oreilly books and this is a typical example of the reasons why i think they are by far the best programming publishers. The writing is fluent, with the difficult stuff presented in a way that makes it easy to understand. Venkat is quite witty at times, which makes the reading that much more pleasurable. All the stuff in it is real-world stuff, and he works through the problems from a variety of perspectives, so you get a thorough understanding. Excellent.