

Chapter 1

What It's All About: Visual Basic Express Takes a Bow

In This Chapter

- ▶ Understanding why VB Express exists
 - ▶ Discovering VB Express's goals
 - ▶ Organizing via the user interface
 - ▶ Programming for the Internet
-

Visual Basic Express is the only popular computer programming language available today. By *popular*, I mean “for the people” — novices, small-business people, amateurs — anyone other than professional programmers. VB Express is the language for the rest of us.

There are far more small-business people, beginners, and enthusiasts than there are professionals, just as amateur cooks outnumber professional chefs. That's why VB Express's predecessor, Visual Basic, was for a decade the world's most popular computer language by a wide margin.

Small-business people need an efficient, understandable programming language to write quick utilities to solve problems unique to their work. A surfboard maker may often need to calculate polyester catalyst ratios, or dad might want to write up a quick history quiz to help with Laura's homework. A hobbyist may enjoy creating a coin collection management program.

Whatever your personal needs, knowing how to program a computer — and thus how to perfectly *customize* its behavior — is a useful and often enjoyable skill.

Something Happened in Orlando

Before you get started writing programs with VB Express, you may find it helpful to understand why Visual Basic Express was even created. After all, Visual Basic .NET already existed, so why this new express version of VB? In the following sections, I explain how VB has changed over the years, so that you can understand just how VB Express fits into the picture and what VB Express can do for you.

Visual Basic through the years

When Microsoft introduced Visual Basic in 1991, VB was primarily a procedure-oriented language (organizing its programs via Events, subroutines, and functions).

To make it easily understood, Visual Basic was designed to be as close to English as possible. Its punctuation, diction, and syntax are familiar — easily understood, remembered, and read — because VB is like a natural human language. Someone creating VB would choose the word *stop*, for example, when they wanted the computer to stop. Makes sense to me.

But in July 2000, something astonishing happened in Orlando. Visual Basic .NET was unveiled in front of an audience of many of the world's best Visual Basic programmers. But VB .NET was so unlike what Basic has always stood for that I heard people around me gasp as the .NET features were described.

VB .NET changed Basic from its traditional role as the fastest route from idea to application to a more powerful — but less organized — language. Where previously simplicity had been the guiding principle, now OOP (object-oriented programming) became a primary goal. And in the process, some needless complexities were introduced.

Visual Basic today

Visual Basic *was* the world's most popular programming language for over a decade, but when Visual Basic .NET appeared, things changed. Because of

the .NET framework's added complexity, Visual Basic became less appealing to many for beginning, amateur, and small-business programmers. Its popularity has declined. In this newest iteration of Visual Basic — Visual Basic Express 2005 — Microsoft has tried to win back this segment of programmers.

Visual Basic 2005 comes in two versions: Visual Basic 2005 for Developers and Visual Basic 2005 Express Edition. Because you bought this book, I assume you have the Express Edition.

VB for Developers focuses on technologies that assist people working in groups on the same program (OOP has many such features). But for people programming alone, VB Express should be everything you need. VB Express is a subset of VB for Developers, but little of significance is left out for those programming solo.

What Does Visual Basic Express Have to Offer?

Nobody knows whether it's too little, too late, but VB Express is designed to rescue Visual Basic, and maybe it will. I hope so. The world needs a computer language for the rest of us — for those who aren't professional programmers. (For professionals, complexity and obscurity often help ensure job security.)

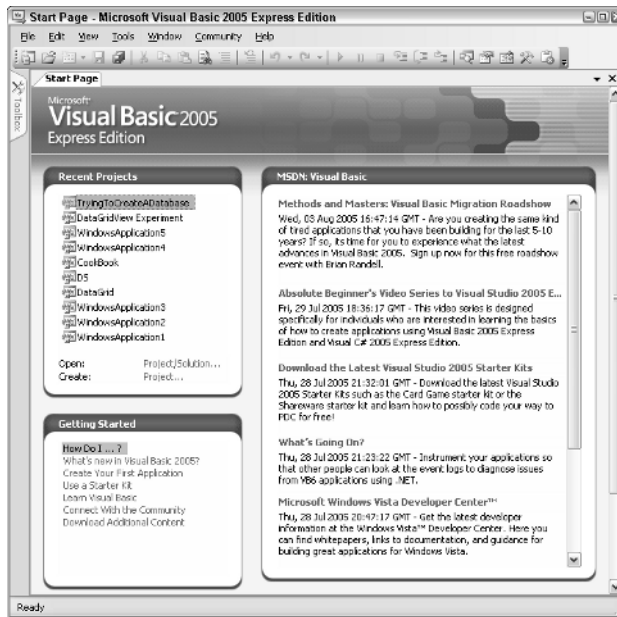
VB Express *is* simpler on the surface than its big brother Visual Basic .NET, but the entire, massive .NET Framework (library of objects) is at your disposal in VB Express. So you'll find all the power you need under the hood.

The Express initiatives

I believe Microsoft understands that there is a problem (how many copies of VB .NET are selling?). Is VB Express the answer? I hope so. VB Express interacts with beginning programmers differently than VB .NET does, but with varying degrees of success:

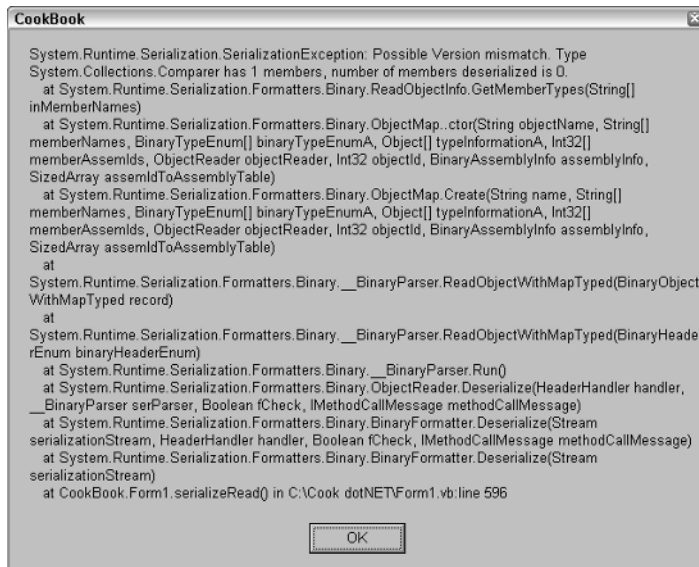
- ✓ The `My` object achieves some abstraction (reducing your need to write huge amounts of code to accomplish some common tasks), as you can see in Chapter 6.
- ✓ The startup screen (see Figure 1-1) appears to contain welcomed simplifications and assistance. But I find much of the content behind the links on this “Welcome to Visual Basic Express” portal deceptive. The “Create Your First Program” step-through leads you to think that you can create “powerful programs . . . quickly and easily.” The example browser trick is hardly a *program*. The My Movie “starter kit” is far from useful or easy as a learning tool. It’s quite advanced actually, and certainly off-putting to all but the Einsteins among novice programmers.

Figure 1-1:
Looks
inviting, but
are the
examples
understand
able and
promises
kept?



- ✓ The menus in VB Express are abbreviated. For example, the macro feature available on the Tools menu in ordinary VB .NET is missing from VB Express. This is similar to what happens when you first install Microsoft Word — presumably not to frighten and confuse beginners with too many features all at once. But in Word you can opt to restore the full menu. In VB Express, the shortening of menus actually represents the removal of features. For example, the macro feature isn't available. Perhaps you're expected to get comfortable using VB Express and then move on up to the more advanced version if you want features such as macros. But don't be too concerned about this — you'll likely find everything you really need in VB Express.
- ✓ Error messages are being improved. Some of them are now more precisely descriptive of the actual error (rather than offering vague, misleading statements about OOP complexities). And now, useful suggested fixes to the code are sometimes offered. But many error messages remain simply alarming and/or useless. Take a look at Figure 1-2

Figure 1-2:
This is a
VB error
message.
Can you
understand
what it's
trying to
tell you?



As you can see in Figure 1-2, this monstrously unhelpful mass of jargon cannot be considered a useful message to the ordinary programmer. Sometimes more is less.

Yet I remain somewhat hopeful. Though not yet accomplished, the initiatives listed here are worthy goals. Perhaps VB Express will evolve into a popular language.

Finding help

Although efforts have been made to improve the VB Help feature since the introduction of VB .NET, unfortunately, much remains to be done before Help is uniformly helpful. In my view, *writers* didn't write the Help examples (neither the code nor the narratives that describe them). Programmers did. And what's worse, I think that many if not most of those programmers are familiar with C languages and only vaguely, if at all, acquainted with Visual Basic. As a result, you often find bizarre code examples in the Help documentation that are a mixture of VB and C styles, accompanied by explanations that defy understanding. As a result, Help too often remains very little help indeed.

For example, imagine a beginner trying to better understand what a *subroutine* is. After all, subs are a major feature in every VB program. So our apprentice — with dewy eyes and a hopeful heart — looks it up in the VB Express Help index and is inundated with confusing jargon, some words that even most advanced VB programmers don't understand: *containing class, interface, structure, Implements keyword, access modifiers, attributes, protected, derived class, ProtectedFriend, assembly, overloading, overriding, redeclares, NotOverridable, MustOverride, shadows, generic procedure*, and on and on. And you're treated to "explanations" like this (send me a translation if you know what they're talking about):

```
The Implements statement must include the interface specified by
interface. However, the name by which the interface defines the Sub (in
definedname) does not have to be the same as the name of this procedure
(in name).
```

Get it? And this nonsense is supposed to be a *help* document for beginners and novices, the purported audience for VB Express. Sounds to me more like page 439 of the government handbook on assembling a card table.



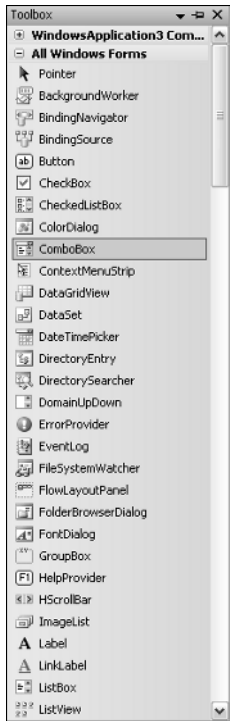
My advice to you is to use this book as your guide instead (and also take advantage of the many code examples in the extensive online Appendix described in this book's Introduction). This book and the huge online Appendix are written by me, a Visual Basic programmer and a writer. You can understand this book and the Appendix. Also, although the VB Express Help system doesn't yet have much to offer (keep checking online to see if it improves), some of VB's features can be quite useful. Chapter 3 introduces some excellent built-in helpers such as IntelliSense, and Chapter 10 explains how to use debugging tools to track down errors.

A couple years ago I offered myself to Microsoft's VB team, like the Aztec virgin that in my heart I am. And I've written them lengthy suggestions, talked to them in person, and had e-mail conversations back and forth with them. They've assured me that they're working on improving the Help system and making other changes.

Taking a First Look

Usually when you start writing a Windows program in VB Express, you begin by adding components (controls from the Toolbox, see Chapter 4) to a form. This creates the user interface on the *form* — the window that the user interacts with. There can be other windows, but many shorter programs have only a single form. You can move the toolbox, shown in Figure 1-3, like most features of the editor (or IDE, see Chapter 2), but most people leave it on the left side.

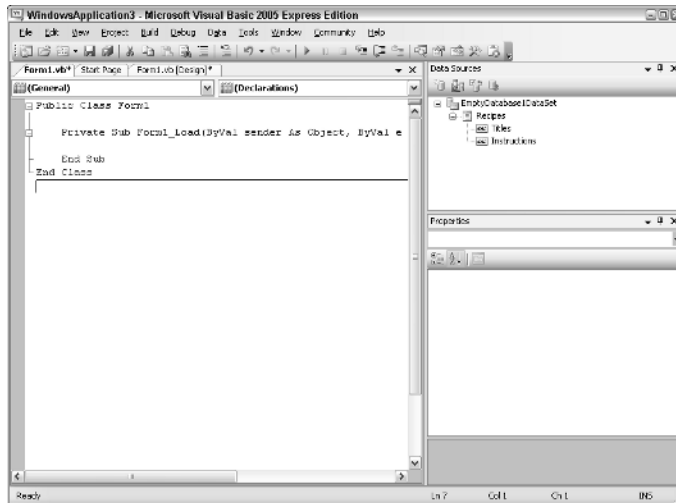
Figure 1-3:
The Toolbox
holds lots of
powerful,
prebuilt
components
that you just
drag and
drop onto
your VB
Express
forms.



Creating a user interface helps organize your project

Adding components and thereby defining what the user sees helps you organize your programming into logical categories. A button that you label `Open File` becomes the location where you write code to load a file from the user's hard drive. If you put a Button control on a form and then double-click that button, a little VB Express magic happens. You're switched to the code window "underneath" the form — or behind the scenes. The user doesn't see the code, but they certainly appreciate that you wrote some programming behind the button so it actually does its job. The Code window is illustrated in Figure 1-4.

Figure 1-4:
You write
your pro-
gramming
here in the
code
window.



Programming for the Web

Internet programming differs somewhat from ordinary Windows programming, but VB Express offers some great tools for this task, too. Part IV covers these topics. You use an editor similar to the IDE for Windows programming, but optimized for Web use — including a specialized set of controls in the Toolbox. The Visual Web Developer Express Edition — a free download — is shown in Figure 1-5.

Figure 1-5:
This editor
is optimized
for designing
and program-
ming Web
pages.



