2

Visual Basic, Controls, and Events

2.1 An Introduction to Visual Basic 2010

Visual Basic 2010 is the latest generation of Visual Basic, a language used by many software developers. Visual Basic was designed to make user-friendly programs easier to develop. Prior to the creation of Visual Basic, developing a friendly user interface usually required a programmer to use a language such as C or C++, often requiring hundreds of lines of code just to get a window to appear on the screen. Now the same program can be created in much less time with fewer instructions.

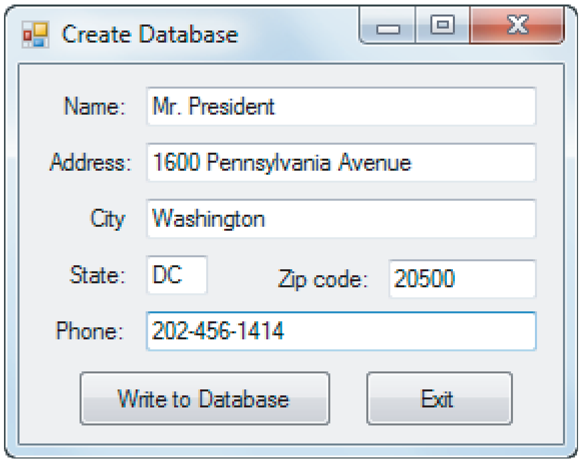
Why Windows and Why Visual Basic?

What people call **graphical user interfaces**, or GUIs (pronounced "gooies"), have revolutionized the computer industry. Instead of the confusing textual prompts that earlier users once saw, today's users are presented with such devices as icons, buttons, and drop-down lists that respond to mouse clicks. Accompanying the revolution in how programs look was a revolution in how they feel. Consider a program that requests information for a database. Figure 2.1 shows how a program written before the advent of GUIs got its information. The program requests the six pieces of data one at a time, with no opportunity to go back and alter previously entered information. Then the screen clears and the six inputs are again requested one at a time.

```
Enter name (Enter EOD to terminate): Mr. President
Enter Address: 1600 Pennsylvania Avenue
Enter City: Washington
Enter State: DC
Enter Zip code: 20500
Enter Phone Number: 202-456-1414
```

FIGURE 2.1 **Input screen of a pre-Visual Basic program to fill a database.**

Figure 2.2 shows how an equivalent Visual Basic program gets its information. The boxes may be filled in any order. When the user clicks on a box with the mouse, the cursor moves to that box. The user can either type in new information or edit the existing information. When satisfied that all the information is correct, the user clicks on the *Write to Database* button. The boxes will clear, and the data for another person can be entered. After all names


Create Database

Name: Mr. President
Address: 1600 Pennsylvania Avenue
City Washington
State: DC
Zip code: 20500
Phone: 202-456-1414

FIGURE 2.2 **Input screen of a Visual Basic program to fill a database.**

have been entered, the user clicks on the Exit button. In Fig. 2.1, the program is in control; in Fig. 2.2, the user is in control!

How You Develop a Visual Basic Program

A key element of planning a Visual Basic program is deciding what the user sees—in other words, designing the user interface. What data will he or she be entering? How large a window should the program use? Where will you place the buttons the user clicks on to activate actions in the program? Will the program have places to enter text (text boxes) and places to display output? What kind of warning boxes (message boxes) should the program use? In Visual Basic, the responsive objects a program designer places on windows are called *controls*. Two features make Visual Basic different from traditional programming tools:

1. You literally draw the user interface, much like using a paint program.
2. Perhaps more important, when you're done drawing the interface, the buttons, text boxes, and other objects that you have placed in a blank window will automatically recognize user actions such as mouse movements and button clicks. That is, the sequence of procedures executed in your program is controlled by "events" that the user initiates rather than by a predetermined sequence of procedures in your program.

In any case, only after you design the interface does anything like traditional programming occur. Objects in Visual Basic recognize events like mouse clicks; how the objects respond to them depends on the instructions you write. You always need to write instructions in order to make controls respond to events. This makes Visual Basic programming fundamentally different from traditional programming. Programs in traditional programming languages ran from the top down. For these programming languages, execution started from the first line and moved with the flow of the program to different parts as needed. A Visual Basic program works differently. Its core is a set of independent groups of instructions that are activated by the events they have been told to recognize. This event-driven methodology is a fundamental shift. The user decides the order in which things happen, not the programmer.

Most of the programming instructions in Visual Basic that tell your program how to respond to events like mouse clicks occur in what Visual Basic calls *event procedures*. Essentially, anything executable in a Visual Basic program either is in an event procedure or is used by an event procedure to help the procedure carry out its job. In fact, to stress that Visual Basic is fundamentally different from traditional programming languages, Microsoft uses the term *project* or *application*, rather than *program*, to refer to the combination of programming instructions and user interface that makes a Visual Basic program possible. Here is a summary of the steps you take to design a Visual Basic program:

1. Design the appearance of the window that the user sees.
2. Determine the events that the controls on the window should respond to.
3. Write the event procedures for those events.

Now here is what happens when the program is running:

1. Visual Basic monitors the controls in the window to detect any event that a control can recognize (mouse movements, clicks, keystrokes, and so on).
2. When Visual Basic detects an event, it examines the program to see if you've written an event procedure for that event.
3. If you have written an event procedure, Visual Basic executes the instructions that make up that event procedure and goes back to Step 1.
4. If you have not written an event procedure, Visual Basic ignores the event and goes back to Step 1.

These steps cycle continuously until the program ends. Usually, an event must happen before Visual Basic will do anything. Event-driven programs are reactive more than active—and that makes them more user friendly.

The Different Versions of Visual Basic

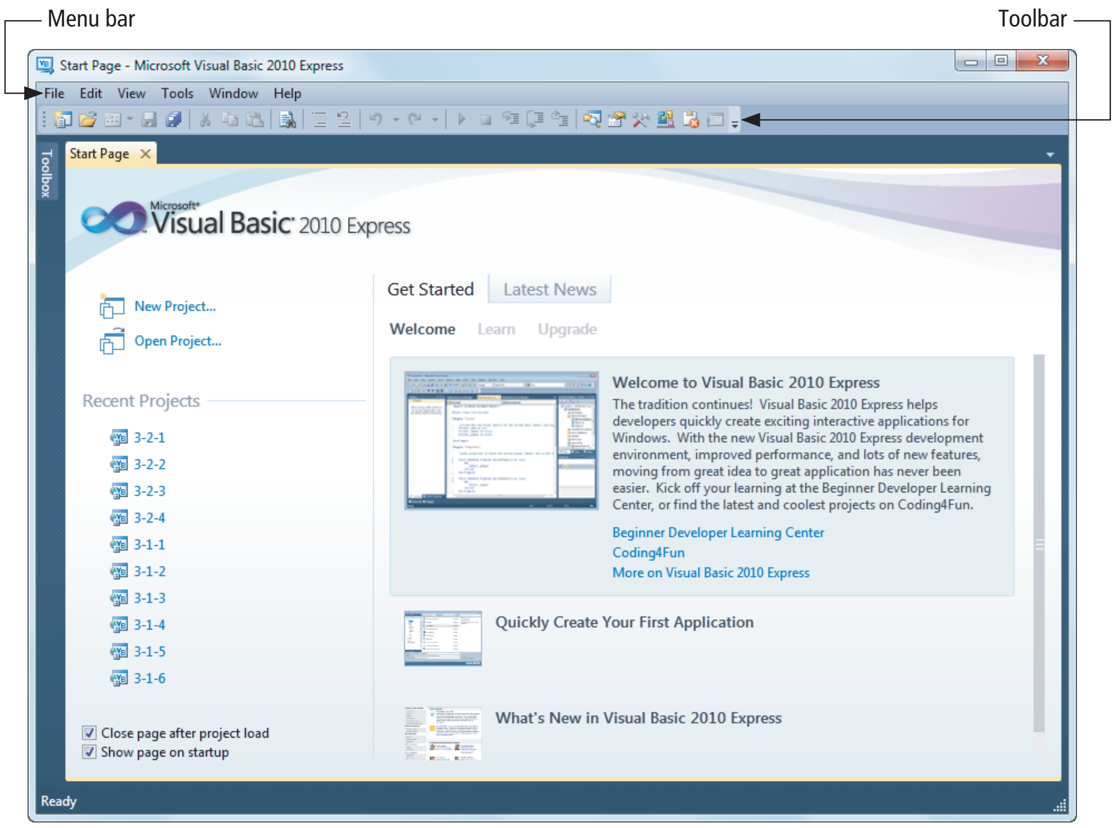
Visual Basic 1.0 first appeared in 1991. It was followed by version 2.0 in 1992, version 3.0 in 1993, version 4.0 in 1995, version 5.0 in 1997, and version 6.0 in 1998. VB.NET, initially released in February 2002, was not backward compatible with the earlier versions of Visual Basic. It incorporated many features requested by software developers, such as true inheritance. Visual Basic 2005, released in November 2005, Visual Basic 2008, released in November 2007, and Visual Basic 2010, released in April 2010 are significantly improved versions of VB.NET.

2.2 Visual Basic Controls

Visual Basic programs display a Windows-style screen (called a **form**) with boxes into which users type (and in which users edit) information and buttons that they click to initiate actions. The boxes and buttons are referred to as **controls**. In this section, we examine forms and four of the most useful Visual Basic controls.

Starting a New Visual Basic Program

Each program is saved (as several files and subfolders) in its own folder. Before writing your first program, you should use Windows Explorer to create a folder to hold your programs.


Menu bar
Toolbar
Start Page - Microsoft Visual Basic 2010 Express
File
Edit
View
Tools
Window
Help
Toolbox
Start Page
Microsoft Visual Basic 2010 Express
New Project...
Open Project...
Recent Projects
3-2-1
3-2-2
3-2-3
3-2-4
3-1-1
3-1-2
3-1-3
3-1-4
3-1-5
3-1-6
Close page after project load
Show page on startup
Get Started
Latest News
Welcome
Learn
Upgrade
Beginner Developer Learning Center
Coding4Fun
More on Visual Basic 2010 Express
Quickly Create Your First Application
What's New in Visual Basic 2010 Express
Ready

Welcome to Visual Basic 2010 Express

The tradition continues! Visual Basic 2010 Express helps developers quickly create exciting interactive applications for Windows. With the new Visual Basic 2010 Express development environment, improved performance, and lots of new features, moving from great idea to great application has never been easier. Kick off your learning at the Beginner Developer Learning Center, or find the latest and coolest projects on Coding4Fun.

FIGURE 2.3 **Visual Basic opening screen.**