The Timer Control As we wrap up our consideration of flow control tools and techniques in this chapter, you should also consider the benefits of using the Visual Studio Timer control, which you can use to execute a group of statements for a specific period of time or at specific intervals. The Timer control is essentially an invisible stopwatch that gives you access to the system clock in your programs. The Timer control can be used like an egg timer to count down from a preset time, to cause a delay in a program, or to repeat an action at prescribed intervals.

Although timer objects aren’t visible at run time, each timer is associated with an event procedure that runs every time the timer’s preset interval has elapsed. You set a timer’s interval by using the Interval property, and you activate a timer by setting the timer’s Enabled property to True. Once a timer is enabled, it runs constantly—executing its event procedure at the prescribed interval—until the user stops the program or the timer object is disabled. Your job as a programmer is to conceive of how to use time in your programs creatively. In other words, what events in a program (or in life) happen at regular intervals? Can you predict or envision the passage of time so that it can be integrated into your code?

**Creating a Digital Clock by Using a Timer Control**

One of the most straightforward uses for a Timer control is creating a custom digital clock. In the following exercise, you’ll create a simple digital clock that keeps track of the current time down to the second. In the example, you’ll set the Interval property for the timer to 1000, directing Visual Studio to update the clock time every 1000 milliseconds, or once a second. Because the Windows operating system is a multitasking environment and other programs also require processing time, Visual Studio might not update the clock precisely every second, but it will always catch up if it falls a bit behind. To keep track of the time at other intervals, such as once every tenth of a second, you simply adjust the number in the Interval property.

**Create the Digital Clock program**

**1**. On the File menu, click the New Project command, and create a new Windows Forms Application project named My Digital Clock. The new project is created and a blank form opens in the Designer.

**2**. Resize the form to a small rectangular window (one that’s wider than it is tall). You don’t want the clock to take up much room.

**3**. Double-click the Timer control on the Components tab of the Toolbox. This is the first time that you have used the Components tab and the Timer control in this book. (The Components tab provides a number of interesting controls that work “behind the scenes” in your programs.) Visual Studio creates a small timer object in the component tray beneath your form.

Certain Visual Studio controls don’t have a visual representation on the form, and when objects for these controls are created, they appear in the component tray beneath the form. (This was the case for the MenuStrip and ToolStrip controls.) However, you can still select controls in this special pane and set properties for them, as you’ll do for the timer object in this exercise.

**4.** Click the Label control in the Toolbox, and then draw a very large label object on the form—a label that’s almost the size of the entire form itself. You’ll use the label to display the time in the clock, and you want to create a very big label to hold the 24-point type you’ll be using. Note When you first create the label object, it resizes automatically to hold the text Label1 in the default size. But when you set the AutoSize property to False in the next step, the label object is restored to the size you originally created.

**5**. Open the Properties window, and set the following properties for the form and the two objects in your program:

|  |  |  |
| --- | --- | --- |
| Object | Property | Setting |
| Label | Auto Size  Text  Text Align | False  *Empty*  Middle Center |
| Timer1 | Enabled  Interval | True  1000 |
| Form1 | Text | Digital Clock |

**Tip**. If you’d like to put some artwork in the background of your clock, set the BackgroundImage property of the Form1 object to the path of a graphics file.

Now you’ll write the program code for the timer.

**6.** Double-click the timer object in the component tray. The Timer1\_Tick event procedure appears in the Code Editor. This is the event procedure that runs each time that the timer clock ticks.

**7.** Type the following statement:

Label1.Text = TimeString

This statement gets the current time from the system clock and assigns it to the Text property of the Label1 object. (If you’d like to have the date displayed in the clock as well, use the System.DateTime. Now property instead of the TimeString property.) Only one statement is required in this program because you set the Interval property for the timer by using the Properties window. The timer object handles the rest.

**8.** Click the Save All button on the Standard toolbar to save your changes. Be sure to name it something so that you will be able to find it in the future.

**9.** Click the Start Debugging button on the Standard toolbar to run the clock. The clock will appear displaying the time.

If you used the System.DateTime.Now property, you’ll also see the date in the clock. If your system clock information appears wrapped, close the program and resize your label and form.

**10.** Watch the clock for a few moments. Visual Basic updates the time every second.

**11.** Click the Close button in the title bar to stop the clock. The Digital Clock program is so handy that you might want to compile it into an executable file and use it now and then on your computer. Feel free to customize it by using your own artwork, text, and colors.

**12.**  Get Ticked-Off.