

## TIMERS – How to Make Things Happen on a Regular Basis

Many times we will want something happening on a regular basis – perhaps every “year” that passes, my Tomagachi pet degrades a little bit.

In other words, every so often (on some regular time interval), something needs to happen. What we need is some sort of timer that will let us know when the “every so often” has occurred, i.e. the time interval has passed. We frequently refer to this *timer interval* as the *timer tick*.

What we want is to set up a timer and have it tick on the desired interval, letting us know when it ticks.

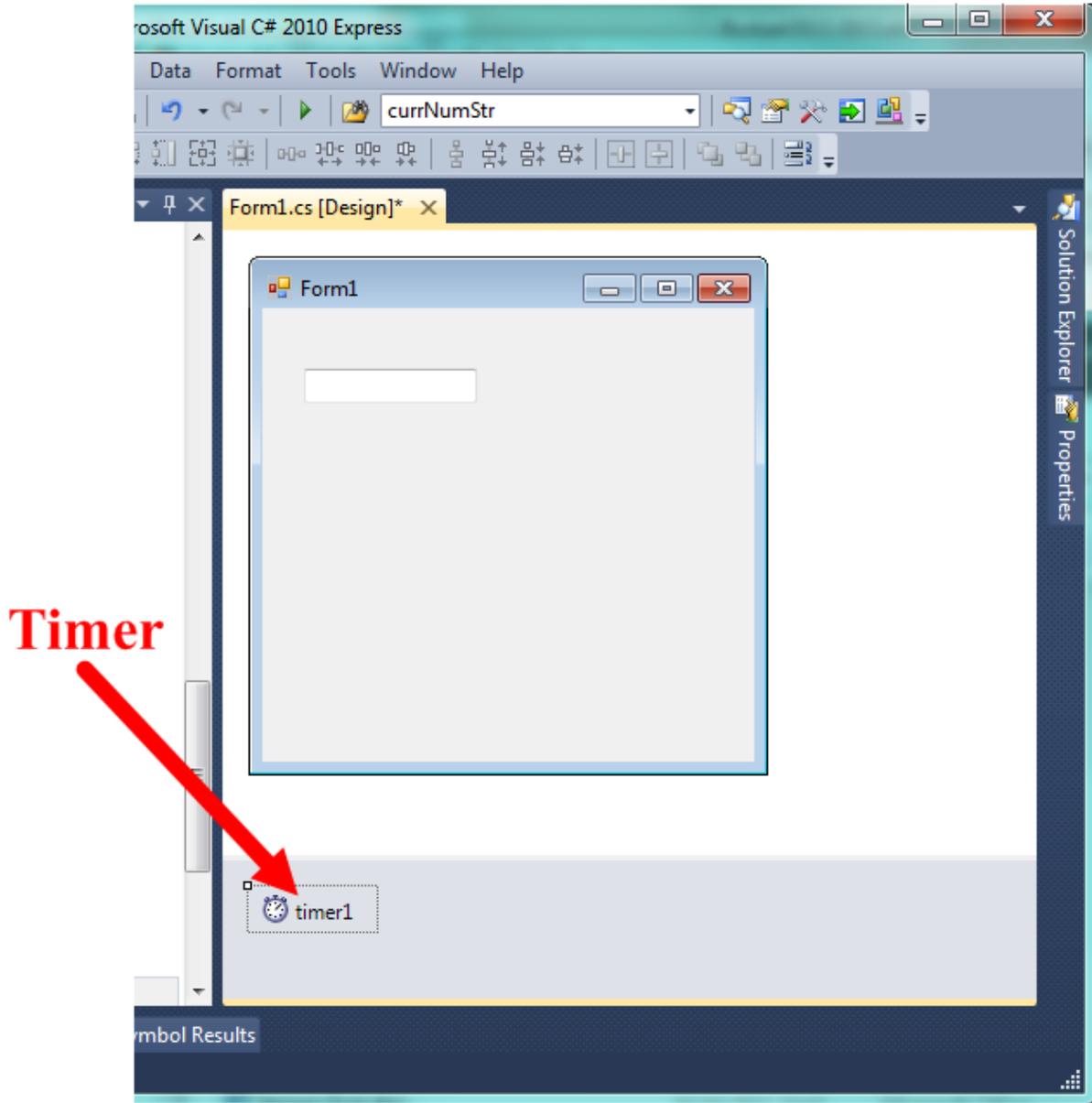
Well, the IDE gives us a timer and makes it very easy to use! Let’s learn how! For this demo, let’s have a display of seconds on the GUI. To do so, we’ll:

1. Add a textbox (to hold the display of seconds),
2. Add a timer, configure it to tick every second,
3. Create the tick event handler method that will get called every time the timer ticks, and
4. Add code to the event handler to display a count of seconds.

Here we go!

## TIMERS – How to Make Things Happen on a Regular Basis

1. Let's start off by creating a new *Windows Form Application*.
2. Drag a **TextBox** and a **Timer** onto the GUI in the design window.
  - You can find the **TextBox** in the *Common Controls* section of the toolbox.
  - You can find the **Timer** in the *Components* section of the toolbox.
  - ...or you can find both in the *All Windows Forms* section. ☺
3. Notice when you drag the timer onto the GUI, it does not appear on the GUI itself:
  - It appears in a section at the bottom of the IDE.



## TIMERS – How to Make Things Happen on a Regular Basis

4. In the properties panel, set the name for these to:

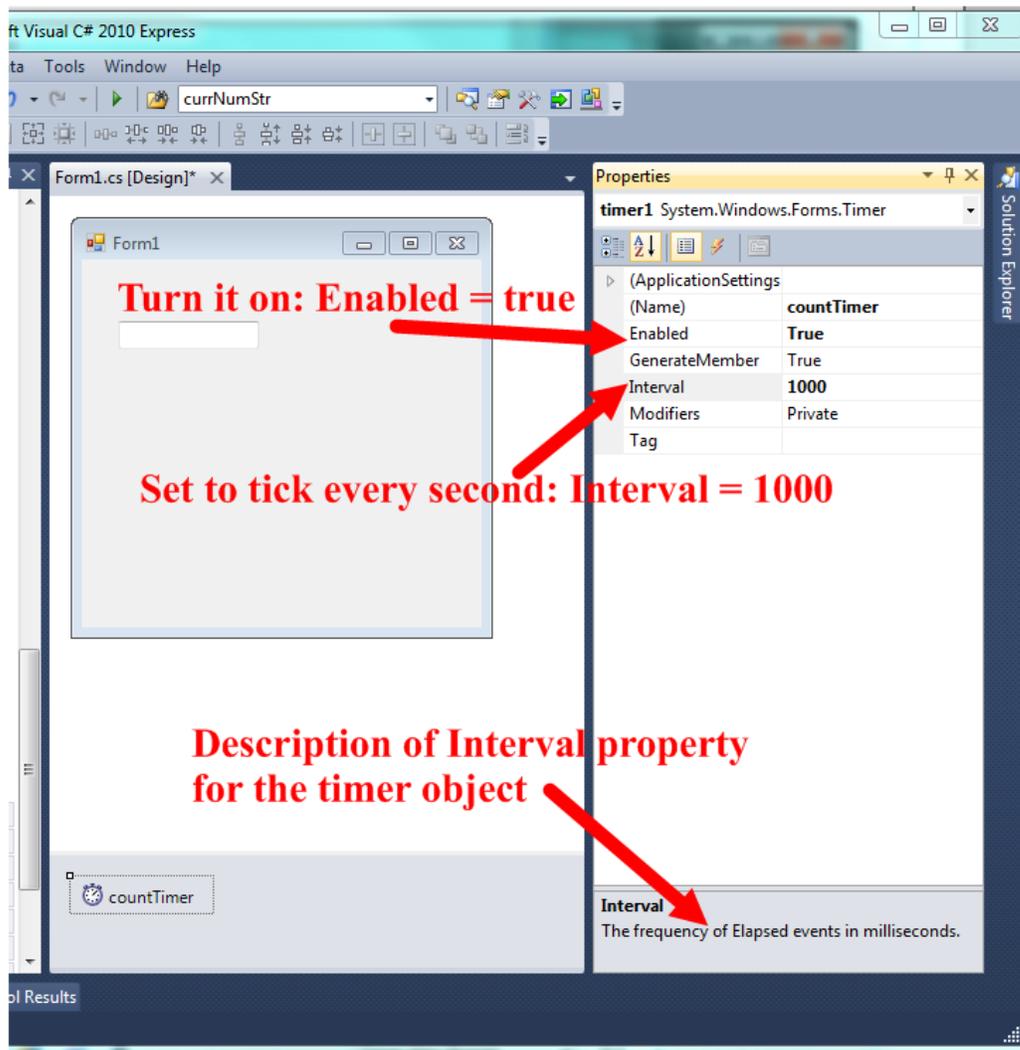
- countTextBox
- countTimer

5. In the properties panel, set up the *Timer*:

- Set *Enabled* to true
- Set the interval to 1000

Click on the timer. In the properties panel, select the *Interval* property (you just set it to 1000).

Notice at the bottom of the properties panel there is a brief description of the timer interval property: it says “The frequency of Elapsed events in milliseconds.”

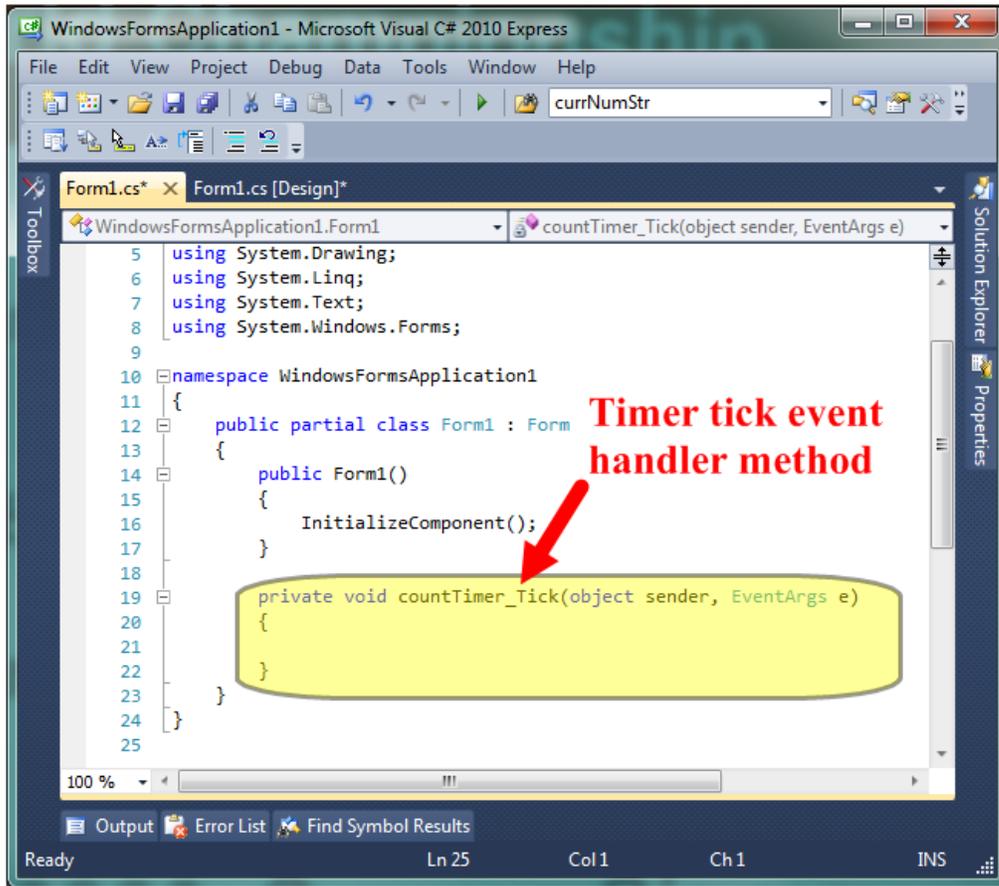


This means the unit of measurement for the timer tick is milliseconds. There are 1000 milliseconds in one second. When we set the interval to 1000, we were saying we want the timer to tick every second (1000 milliseconds). If you wanted the timer to tick every 5 ½ seconds, you would set the interval to 5500 ( $5.5 * 1000$ ).

## TIMERS – How to Make Things Happen on a Regular Basis

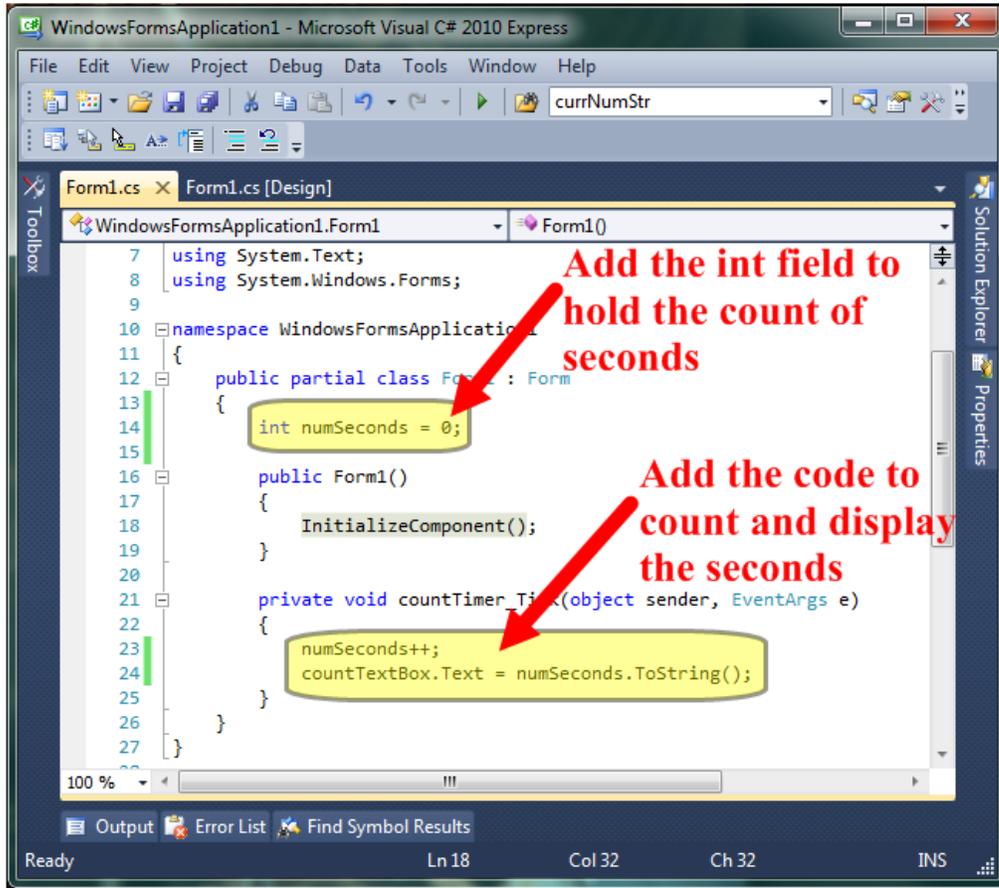
We now have the timer configured to tick every second.

The next step is set it up so the timer will tell us when it ticks; to do so simply double-click on the timer object at the bottom of the Form1.cs (Design) screen. This will open the code window and create the event handler method for the timer tick event. This is where we will add the code to do whatever needs to happen every time the timer ticks.



## TIMERS – How to Make Things Happen on a Regular Basis

Next let's create a class variable (*field*) to hold the count of seconds. In the timer tick event handler we will increment (add 1 to) this field ... this will count the number of seconds that have gone past. Here is the code:

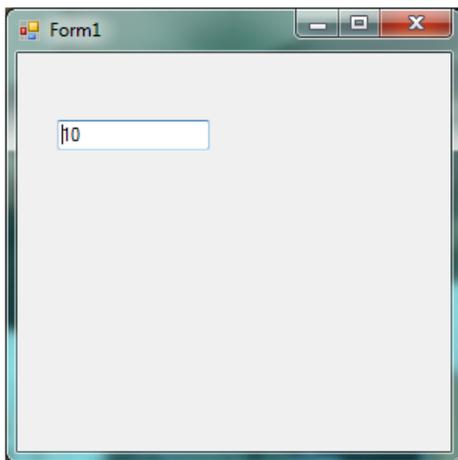


```
7 using System.Text;
8 using System.Windows.Forms;
9
10 namespace WindowsFormsApplication1
11 {
12     public partial class Form1 : Form
13     {
14         int numSeconds = 0;
15
16         public Form1()
17         {
18             InitializeComponent();
19         }
20
21         private void countTimer_Tick(object sender, EventArgs e)
22         {
23             numSeconds++;
24             countTextBox.Text = numSeconds.ToString();
25         }
26     }
27 }
```

**Add the int field to hold the count of seconds**

**Add the code to count and display the seconds**

Run the code and you will see the second count updating in the textbox:



You now know how to use a timer in your GUI! All you need to do is decide how often you want the timer to tick, set the interval appropriately and add the event handler code.