



C# Tips & Tricks for Beginners

Reference guide to all C# articles available on
HowToIdeas.net

Released By
TheWindowsClub.com

Introduction & Disclaimer

This book is the first book in our series of books for “C# Tips and Tricks” for beginners and this book has been released through [TheWindowsClub](#). I have included First 20 Tips and tricks which I have written for [HowToIdeas](#). I have tried my level best to make the language of this book as simple so that a novice can also understand.

I have personally tried and tested all of the tweaks discussed in this guide. However, I request fellow readers to be cautious while trying it out with your system and we hold no responsibility for the damage if happen to your PC. Always take a backup copy of all important data/registry before attempting to change the system and registry settings.

Last but not the least- Learn, Share & Grow. Add your knowledge to what I have and share it unconditionally with the people who are looking for it.

Table of Contents

<u>HOW TO CREATE SQL CONNECTION IN C#</u>	<u>3</u>
<u>HOW TO WRITE TEXT TO A TXT FILE IN C#</u>	<u>5</u>
<u>HOW TO DELETE COOKIE USING C#</u>	<u>6</u>
<u>HOW TO SEND EMAIL USING YOUR GMAIL ACCOUNT IN C#</u>	<u>7</u>
<u>HOW TO READ TEXT FROM A TXT FILE</u>	<u>8</u>
<u>HOW TO CHECK IF A KEY IS PRESSED IN C#</u>	<u>9</u>
<u>HOW TO RENAME A FILE USING C#</u>	<u>10</u>
<u>HOW TO DETERMINE WHICH .NET FRAMEWORK VERSION IS INSTALLED</u>	<u>11</u>
<u>HOW TO DISABLE RIGHT CLICK IN C# TEXTBOX</u>	<u>13</u>
<u>HOW TO READ TEXT FROM A FILE</u>	<u>14</u>
<u>HOW TO ADD SELECT ALL BUTTON OR CHECKBOX IN CHECKEDLISTBOX</u>	<u>15</u>
<u>HOW TO CREATE A NEW FOLDER USING C#</u>	<u>17</u>
<u>HOW TO GET LIST OF ALL THE RUNNING PROCESSES IN C#</u>	<u>18</u>
<u>HOW TO DOWNLOAD A FILE USING C#</u>	<u>20</u>
<u>HOW TO FADE OUT WINDOWS FORM BEFORE CLOSING</u>	<u>21</u>
<u>HOW TO CREATE A NUMERIC TEXTBOX IN C#</u>	<u>23</u>
<u>HOW TO CREATE AUTO COMPLETE TEXTBOX IN C#</u>	<u>24</u>
<u>HOW TO WRITE A XML FILE USING C#</u>	<u>26</u>
<u>HOW TO STOP NOT RESPONDING PROGRAMS USING C#</u>	<u>28</u>
<u>HOW TO SELECT SOME TEXT FROM A TEXTBOX</u>	<u>29</u>

How to Create SQL Connection in C#

Instructions:

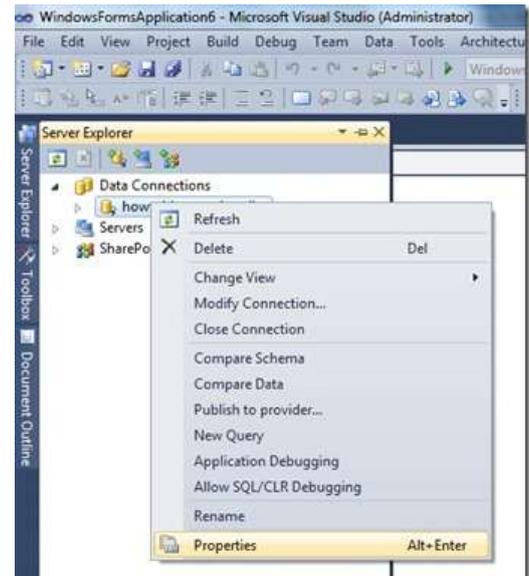
1. In the code behind file, add a new namespace.

```
using System.Data.SqlClient;
```

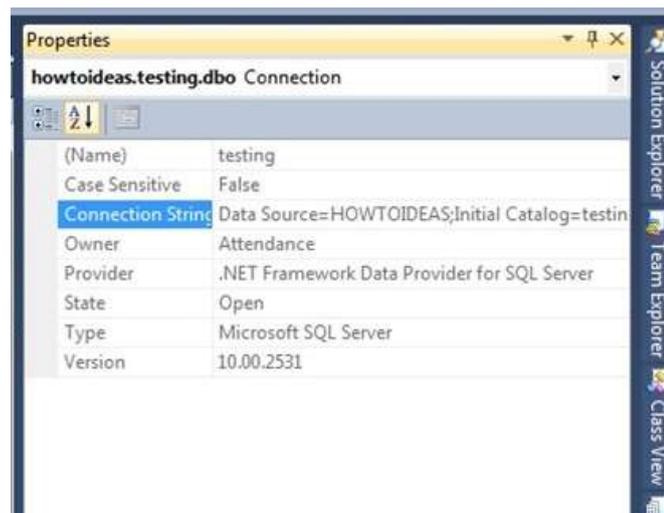
2. Use the following code to create a connection with SQL Server Database in any method.

```
SqlConnection con = new  
SqlConnection();
```

3. Now we have to provide an address of our database to this connection. For that, you have to create a new data connection to your database in visual studio. If you don't know how to do that, see [my article](#) on this topic. After that, when you have data connection ready, right click on it in "Server Explorer" and choose properties.



4. From the "Properties" window, copy the value of *connectionString*.



5. Now add the following code

```
con.ConnectionString = "paste your connection string here";
```

paste the connectionString value you just have copied in the double quotes in upper line of code.

6. After that you just have to open your connection with database. Do so by adding following line of code.

```
con.Open();
```

7. Test your application. If it doesn't show any error, then it means your connection has succeeded. You now can read, insert, update, delete data from that database.

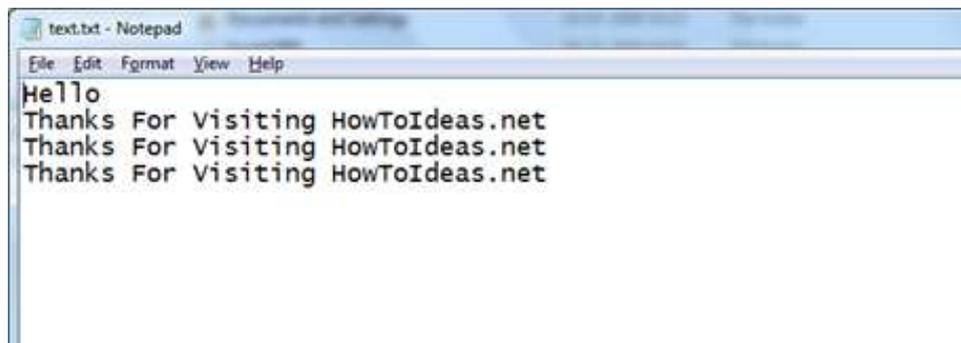
How to Write Text to a Txt File in C#

Instructions:

1. Open Any Project or create a new one.
2. Use the following code to write text to any File

```
FileInfo f = new FileInfo("C:\\text.txt"); //Give address of  
your file here  
StreamWriter sw = f.AppendText(); //It will place your cursor at  
the end of file  
sw.WriteLine("Thanks For Visiting HowToIdeas.net"); //This will  
ad the string to your file  
sw.Close();
```

3. It will Append the text at the end of the file you will mension, every time this code will run. See the result of this code after running three times



How to Delete Cookie Using C#

Instructions:

1. Check if the cookie already exists and if it, then create a new cookie having the same name as the name of the cookie you want to delete.
2. After that set the expiry date of this new cookie to any past time say 2 day back from now.
3. Then just add the cookie to Response object and you are done.

Code:

```
if (Request.Cookies["yourCookieName"] != null)
{
    HttpCookie myCookie = new HttpCookie("yourCookieName");
    myCookie.Expires = DateTime.Now.AddDays(-2d);
    Response.Cookies.Add(myCookie);
}
```

Just replace “yourCookieName” with the name of your cookie you want to delete.

How to Send Email Using Your Gmail Account in C#

Instructions:

1. Add the following namespaces to your Code-Behind file.

```
using System.Net.Mail;
using System.Net;
using System.Configuration;
```

2. Add the following lines of code in Code-Behind file where you want to send an email.

```
System.Net.Mail.MailMessage mail =
new System.Net.Mail.MailMessage("yourEmailId",
"destinationEmailId");

mail.Subject = ""; //Enter the text for the subject of the mail
in quotes.

mail.Body = ""; //Enter the text for the body of mail within the
quotes.

mail.IsBodyHtml = true;

SmtpClient client = new SmtpClient("smtp.gmail.com");

NetworkCredential cred = new NetworkCredential("yourEmailId",
"yourPassword");

client.EnableSsl = true;
client.Credentials = cred;
try
{
    client.Send(mail);
}
catch (Exception)
{
}
```

How to Read Text from a TXT File

Instructions:

1. Start by adding following namespace to your code file.

```
using System.IO;
```

2. After that add the following code, where you want to read text from a file. This code will read all the text present in the file, line by line.

```
try
{
using (StreamReader sr = new StreamReader("file.txt")) // replace
the file.txt with the exact location of your file.
{
String line;
while ((line = sr.ReadLine()) != null) //sr.ReadLine() reads text
line by line and every time it reads a line,
it places curson in the next line. So we will read the text to
the last
line and after that this function return null value.
{
Console.WriteLine(line);
}
}
}
catch (Exception ex)
{
Console.WriteLine("The file could not be read:");
Console.WriteLine(ex.Message);
}
```

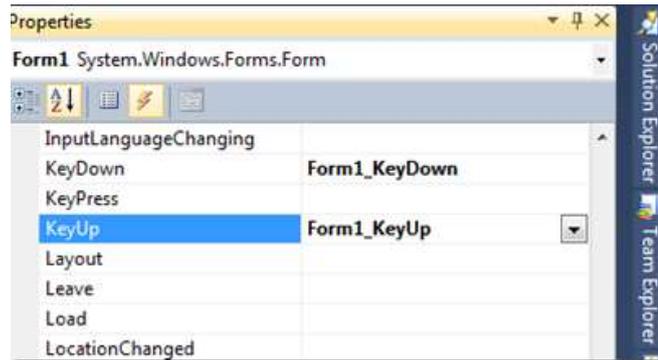
3. If you want to check that whether the file you specified exists or not, you can use the following code for that purpose.

```
if (!File.Exists("file.txt")) // replace the file.txt with the
exact location of your file.
{
return;
}
```

How to Check If a Key Is Pressed In C#

Instructions:

1. Open the project, click on any empty area on the form.
2. Ensure that Form is selected not a control if present on your Form.
3. In the Properties Window, click on the *Lightening* icon to open events for the Form.
4. Double click on *KeyUp* and *KeyDown* events to create event methods for these two.



5. Use the following code to detect if required key is pressed.

```
// This Event will occurs when a key is pressed
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Up)
    {
        // Write your code for this key
    }
    if (e.KeyCode == Keys.Down)
    {
        // Write your code for this key
    }
}

// This Event will occurs when a key is released
private void Form1_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Up)
    {
        // Write your code for this key
    }
    if (e.KeyCode == Keys.Down)
    {
        // Write your code for this key
    }
}
```

How to Rename a File Using C#

Instructions:

The only way you can use to do this is first copy the file with required name in the same directory and then delete the first file. This is a bit long procedure for this problem, as you can rename the file just by moving the file from one directory to another(in our case the destination directory is same to source directory) with changed name. You just enter the following line of code to rename your file.

```
File.Move(@"C:\oldfile.txt", @"C:\newfile.txt");
```

This code will rename *oldfile.txt* to *newfile.txt* which is present in *C:/* directory.

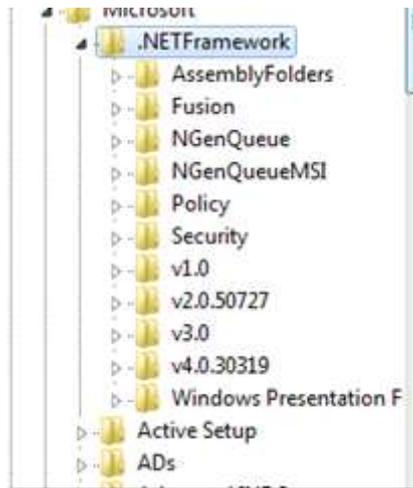
Note: In this method old file is moved from one directory to another. So, time taken to complete the process is depend on file size.

How to Determine which .NET Framework Version is Installed

Instructions:

1. Using Registry

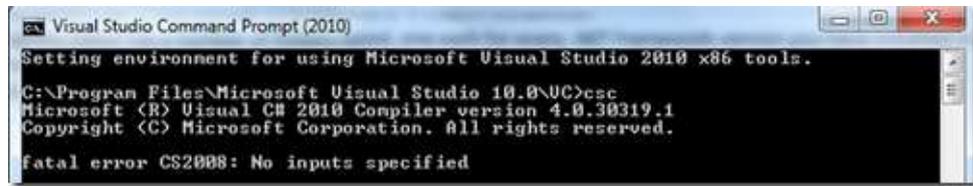
- Open your Registry Editor by typing *regedit* in Start menu search box.
- In the Registry Editor Move to the following location
HKEY_LOCAL_MACHINE->SOFTWARE->MICROSOFT->.NETFRAMEWORK
- This Key in Registry Editor has a number of folders listed, one each for every .NET Framework version you have installed in your Computer.
- If your Registry Editor does not have the above Location, then try the following location
HKEY_LOCAL_MACHINE->SOFTWARE->MICROSOFT->NET FRAMEWORK SETUP/NDP



2. Using Visual Studio Command Prompt

- If you have installed the latest .NET Framework, then you can find a Visual Studio Command Prompt, which is available from the Visual Studio Tools folder which can be used to compile our Code files.
- You can find the Visual Studio Command Prompt in the following directory
 - **For .NET Framework 4.0 or Later**
Start -> All Programs -> Microsoft Visual Studio -> Visual Studio Tools, and then click Visual Studio Command Prompt
 - **For Earlier Version Of .NET Framework**
Start -> All Programs -> Microsoft Visual Studio -> Visual Studio Tools, and then click Visual Studio Command Prompt. (Only if you have Visual Studio 2005 or later is installed)
or
Start -> All Programs -> Microsoft .NET Framework SDK v2.0, and then click SDK Command Prompt. (Only if you have .NET Framework 2.0 or higher is installed)

- If we just type `csc` in that Visual Studio Command Prompt and hit Enter, It will give an error and also tell us about the Latest .NET Framework installed in the machine.



3. Using C# Code

- You don't need to have Visual Studio installed too use this method. Just open any text editor like Notepad and add the following code in it.
- Now save this file with the any name but extension must be “.cs” and you must select “All Files” in the “Save As Type:” box while saving the file. I am saving this file with the name “Framework.cs” in D:\ directory.



- Now open the Visual Studio Command Prompt which I had described in the 2nd method.
- Move to the directory where you have saved the file, in my case, it is in D:\ directory.
- Type the following command in there and hit Enter, It will compile your file and make a .exe file from it.
`csc Framework.cs`
- It might give you a warning but don't worry it will compile your file correctly.
- Now type just `Framework.exe` there and hit Enter. It will show you the Framework Version as output of the program.



How to Disable Right Click in C# Textbox

Instructions:

1. Say you have a textbox with ID textBox1, then add the following line in the Form Load Event

```
textBox1.ContextMenu = new ContextMenu();
```

2. It will Re initialize your Right Click Context Menu for that textbox having no items in that menu. So, now if you try to right click on that textbox, you won't see any Right Click Context Menu.

How to Read Text from a File

Instructions:

1. Start by adding following namespace to your code file.

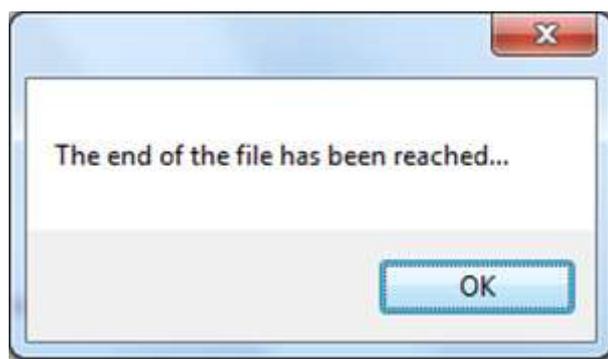
```
using System.IO;
```

2. Add a Label to your form and add following code in the Form_Load method

```
string file = @"D:\test.txt";

if (!File.Exists(file)) //check if file exists or not
{
    MessageBox.Show("File {0} does not exist.", file);
}
else
{
    using(StreamReader sr = File.OpenText(file))
    {
        label1.Text = "";
        String text;
        while ((text = sr.ReadLine()) != null)
        {
            label1.Text += text + "\n";
        }
        MessageBox.Show("The end of the file has been reached...");
    }
}
```

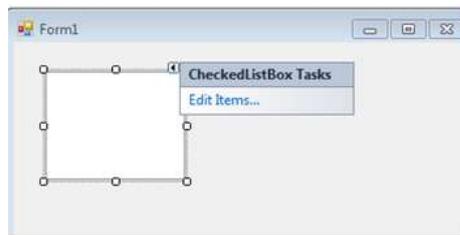
3. After reading all the text from that file, you will get following notification and when you click OK, your label will be having all the text from that file.



How to Add Select All Button or Checkbox in CheckedListBox

Instructions:

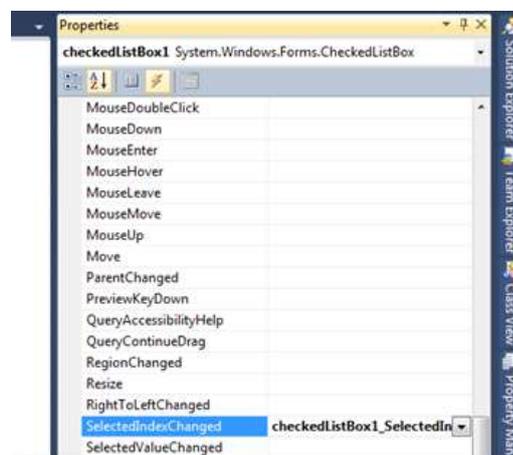
1. Open the Windows form, in which you want to add a Select All checkbox.
2. From toolbox add CheckedListBox into your Form.
3. Click on the “Small Arrow” pointing toward right in the CheckedListBox and select “Edit Items...”.



4. Enter the options you want to have in CheckedListBox, but you must add “Select All” option at the top.

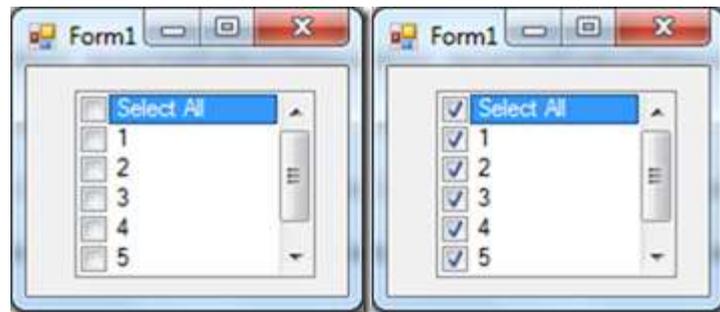


5. Click Ok and then open Properties Window for this CheckedListBox. In the Properties Window, click on Events icon and then double click on SelectedIndexChanged event.



6. This will create an event function for this event. Now add the following code in the newly created function

```
if (checkedListBox1.SelectedIndex == 0)
{
    for (int i = 1; i < checkedListBox1.Items.Count; i++)
    {
        checkedListBox1.SetItemChecked(i,
checkedListBox1.GetItemChecked(0));
    }
}
else
{
    if
(!checkedListBox1.GetItemChecked(checkedListBox1.SelectedIndex)
)
    {
        checkedListBox1.SetItemChecked(0, false);
    }
}
```



How to Create a New Folder Using C#

Instructions:

If you want your application to create folders or directories dynamically at run time to save any kind of data or files, you can do so by using any of the following command/method.

```
System.IO.Directory.CreateDirectory(@"D:\hello")
```

```
System.IO.DirectoryInfo dinfo = new DirectoryInfo(@"D:\hello");  
dinfo.Create();
```

If you want to check whether a directory with the name you specified already exists or not then you can try the following method

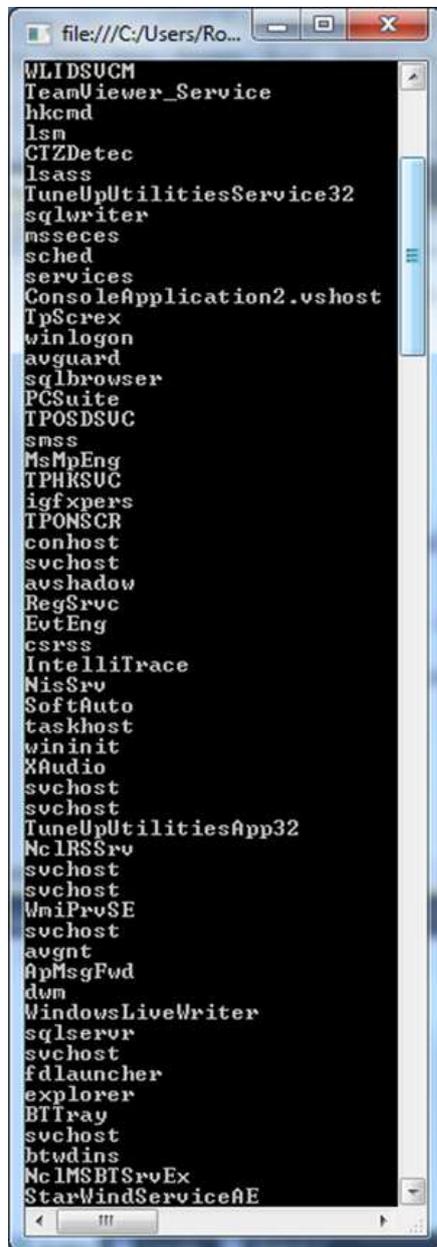
```
System.IO.DirectoryInfo dinfo = new DirectoryInfo(@"D:\hello");  
if (!dinfo.Exists) //checks if the directory already exists or not  
{  
    dinfo.Create();  
}
```

How to Get List of All the Running Processes in C#

Instructions:

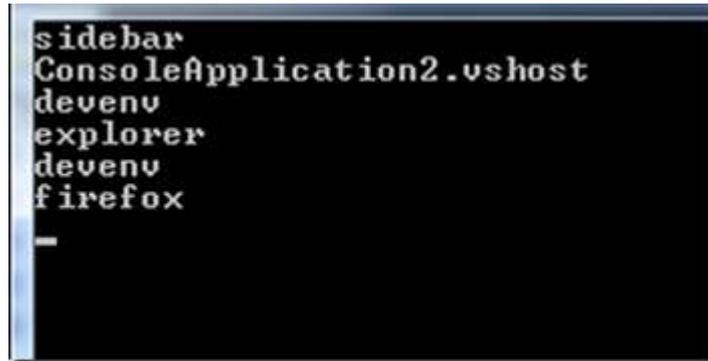
1. Open the project in Visual Studio in which you want to list all the currently running processes.
2. Add the following code to list all the processes in Console Window.

```
foreach(Process p in Process.GetProcesses())  
{  
    Console.WriteLine(p.ProcessName);  
}  
Console.ReadKey();
```



3. If you just want to get the process started by User not by the Windows then use the following code

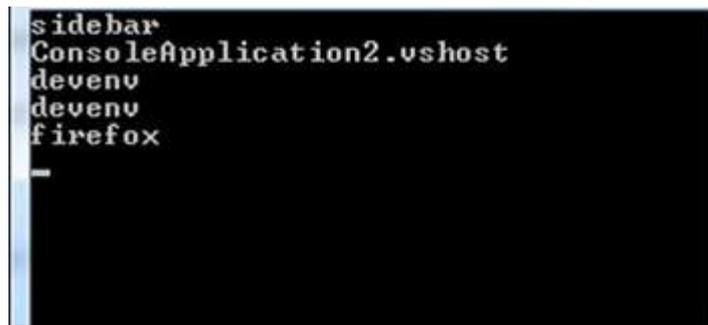
```
foreach (Process p in Process.GetProcesses().Where(p =>
p.MainWindowHandle != IntPtr.Zero))
{
    Console.WriteLine(p.ProcessName);
}
Console.ReadKey();
```



```
sidebar
ConsoleApplication2.vshost
devenv
explorer
devenv
firefox
-
```

4. If you just want to get the names of programs not the opened folders, then use the following code

```
foreach (Process p in Process.GetProcesses().Where(p =>
p.MainWindowHandle != IntPtr.Zero && p.ProcessName !=
"explorer"))
{
    Console.WriteLine(p.ProcessName);
}
Console.ReadKey();
```



```
sidebar
ConsoleApplication2.vshost
devenv
devenv
firefox
-
```

How to Download a File Using C#

Instructions:

1. Add the following namespace to your code file
using System.Net;
2. Now add the following code where you want to save the file.

```
string address =  
"http://www.google.com/intl/en_ALL/images/srpr/logo1w.png";  
string path = @"D:\logo.png";
```

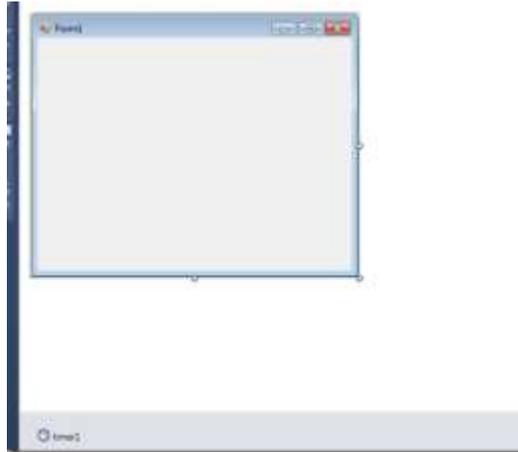
```
WebClient w = new WebClient();  
w.DownloadFile(address, path);
```



How to Fade Out Windows Form Before Closing

Instruction:

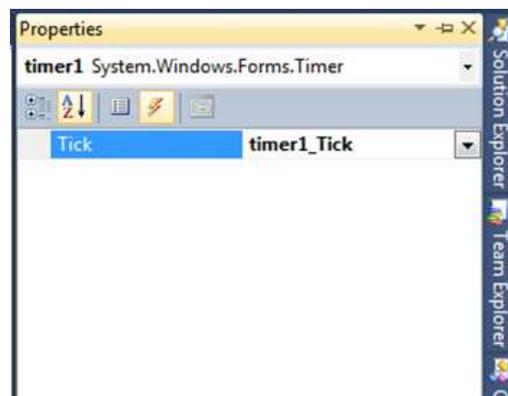
1. In your form, insert new Timer control from toolbox. You will see timer1 control at the lower bar as i have in the following image.



2. Now double click on Form, which will take you to the form_load event, add the following code there. Here we are telling .NET to disable the timer, so the timer will not tick and we are setting its tick interval to 200 millisecond. So, when we enable this timer, it will tick after every 200 millisecond.

```
timer1.Enabled = false;  
timer1.Interval = 200;
```

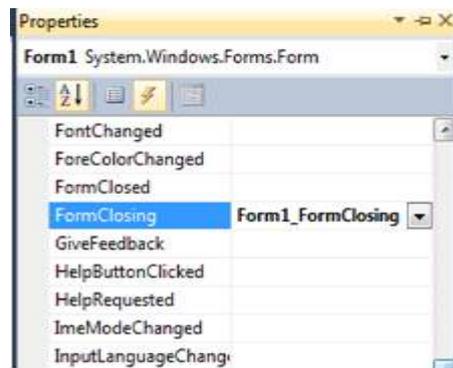
3. Now select the timer from the designer and then open its Properties by pressing F4, now go to its events and then double click on Tick event, which will create an event for timer tick. And this event will occur every 200 millisecond when the timer is enabled, but currently the timer is disabled. So, the code we write there won't work for now, but will work when we enable the timer.



- Now in the timer1_Tick event, add the following code. This code will decrease the opacity of form by 0.05 every time the timer ticks. and when the opacity of form decreases to less than 0.05, we will close the form.

```
if (this.Opacity > 0.05)
{
    this.Opacity -= 0.05;
}
else
{
    timer1.Enabled = false;
    this.Close();
}
```

- We have written code to decrease the opacity of form by 0.05 after every 200 millisecond when the timer ticks, but this will happen only if the timer will be enabled, which is currently disabled. So, now we just have to enable the Timer, but where should we enable it. And the answer is: when user tries to close the form... So, when user click on the close button, we enable the timer, which will tick after every 200 millisecond and opacity of our form will decrease by 0.05 every time till it has opacity more than 0.05, after which we finally close the form.
- Now Select the Form and then open its Properties by pressing F4, now go to its events and then double click on Form Closing event there, which will create an event for Form Closing in the code file.



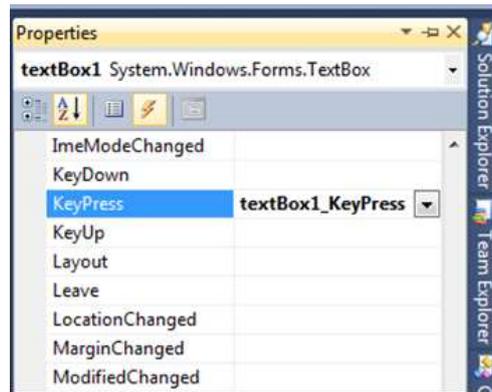
- In the Form closing event, add the following code. This code will enable the timer, if opacity of form is more than 0.05 and will cancel the closing event. But when the opacity of form decreases to less than 0.05 this code will not work and our form will get closed.

```
if (this.Opacity > 0.05)
{
    timer1.Enabled = true;
    e.Cancel = true;
}
```

How to Create a Numeric Textbox in C#

Instruction:

1. Open your project in which you want to create a numeric textbox.
2. If you already have textbox placed in your Form then select it, otherwise create one now and select that textbox.
3. Open Properties for that textbox by pressing F4 key and the select Events tag.
4. Double click on Key Press event for this textbox, which will create event `textBox1_KeyPress` in the code file.



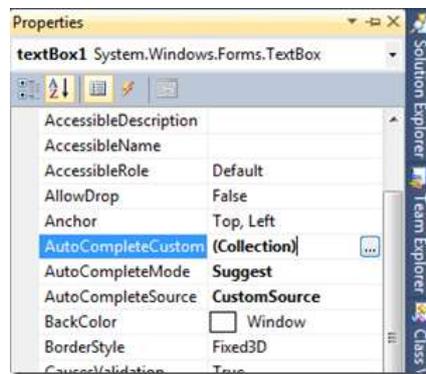
5. Now we want user to enter only numbers or decimal or negative sign. So enter the following code in the KeyPress event. This code will check whether the key pressed in decimal or negative sign. If it is then it's OK. Otherwise in default section, it will check whether the key entered is digit or not, if it is not then handle the event i.e. don't allow to enter this key.

```
switch (e.KeyChar)
{
    case '.':
        break;
    case '-':
        break;
    default:
        if (!Char.IsDigit(e.KeyChar))
        {
            e.Handled = true;
        }
        break;
}
```

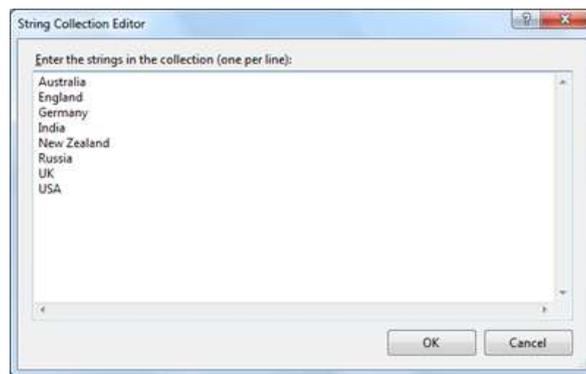
How to Create Auto Complete Textbox in C#

Instruction:

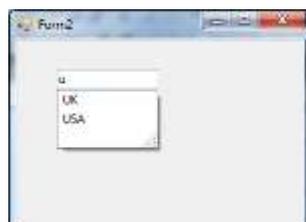
1. Open any project or create a new one.
2. Add a textbox to your Form from Toolbox.
3. Select the Textbox you have added and then press F4, which will open its properties.
4. Change the value of “AutoCompleteMode” from “None” to “Suggest”, at the end of this post, I will talk about other options as well.
5. Change the value of “AutoCompleteSource” from “None” to “CustomSource”.
6. Now click on the ellipses “...” button for “AutoCompleteCustom” .



7. It will open “String Collection Editor” dialogue.
8. Just Enter their some Country names line by line.



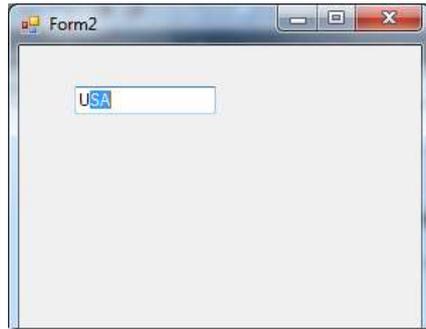
9. Click OK and then run your project.
10. Start typing any Country name from that list and you will get a pop down showing you suggestions.



11. You can scroll through these suggestions using up down arrow keys which will complete the suggested word for you in textbox. You can also double click on any suggestion to insert into textbox.

Now What is the difference between three “AutoCompleteMode”, “Suggested”, “Append”, “SuggestAppend”.

- If you choose Suggested, It will show you a dropdownlist of all the suggested values.
- If you choose Append, It will append the suggested word to textbox and if you have multiple suggestions, then you can scroll through them using UP DOWN arrow keys.



- If you choose SuggestAppend, then it will suggest you in both the way



How to Write A XML File Using C#

Instructions:

As we know XML is a mark-up language, So it has tags like we see in HTML files. So I will be creating an XML file having following structure at the end.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<Employees Total="4">

<Employee Name="abc">Developer</Employee>

<Employee Name="def">Designer</Employee>

<Employee Name="ghi">Writer</Employee>

<Employee Name="jkl">Developer</Employee>

</Employees>
```

In this article, I will be creating an XML file in my D:\ directory using C# in which we write XML using C# code. To accomplish the task I will be using two built in classes

1. XmlWriter – to create new XML file and to write all the data in that
2. XmlWriterSetting – to define general styles and settings to the XML we write using upper class

Defining Settings For Our XML File

Write the following code in C# which will create settings for our XML file. In this we are telling the .NET to enable, Indenting in our XML file and to use \r\n characters for new lines. Which will help us create an XML file with nice design.

```
XmlWriterSettings settings = new XmlWriterSettings();

settings.Indent = true;

settings.NewLineChars = "\r\n";
```

Creating New XML File

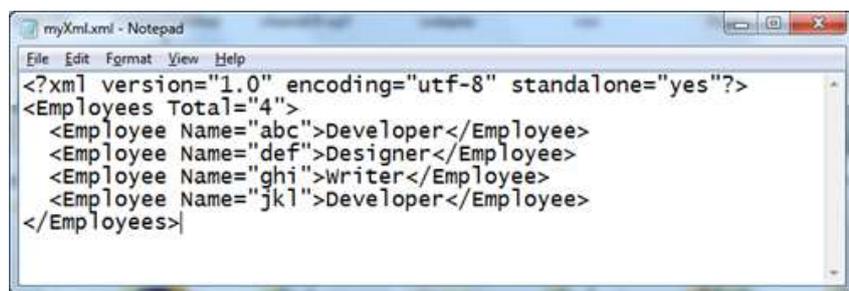
Use the following code to create new instance of XmlWriter class by providing path of our XML file and also settings variable which we have created above, which will tell this XmlWriter instance to do decoration for us in the XML file. This line will create a new file at the path we provide, if there is no such file exists. If there is some file, then it will overwrite that file.

```
XmlWriter writer = XmlWriter.Create(@"D:\myXml.xml", settings);
```

Now write the following code which will write the desired Xml in our XML file.

Code for writing XML in a File

```
string[] text = new string[] { "Developer", "Designer", "Writer",  
"Developer" };  
  
string[] names = new string[] { "abc", "def", "ghi", "jkl" };  
  
writer.WriteStartDocument(true);  
  
writer.WriteStartElement("Employees"); // start employees element  
  
writer.WriteAttributeString("Total", "4"); // write Total attribute  
and its value for employees element  
  
for (int i = 0; i < 4; i++)  
{  
  
writer.WriteStartElement("Employee"); // start employee element  
  
writer.WriteStartAttribute("Name"); // write name attribute to  
employee element  
  
writer.WriteString(names[i]); // write value of Name attribute  
  
writer.WriteEndAttribute(); // Close Name attribute  
  
writer.WriteString(text[i]); // write inner text to employee element  
  
writer.WriteEndElement(); // close employee element  
  
}  
  
writer.WriteEndElement(); // close Employees element  
  
writer.Close(); // close the writer which will write all the xml to  
our file
```



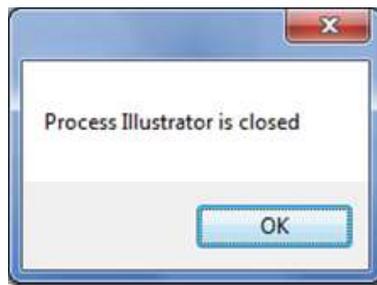
How to Stop Not Responding Programs Using C#

Instructions:

1. Open Visual Studio and in the code file, write the following code to close all the not responding programs.

```
foreach (Process p in Process.GetProcesses())
{
    if (!p.Responding)
    {
        p.Kill();
        MessageBox.Show(String.Format("Process {0} is closed",
p.ProcessName));
    }
}
```

2. This will not only close that program but also displays message box saying which process is closed.



How To Select Some Text From A TextBox

Instructions:

1. Open Visual Studio and Create a New project.
2. In the new project, you will be having a form.
3. Insert two textboxes onto that form.
4. First textbox will be having a lot of text from which we select just a part of the whole text of that control which was given by second textbox.
5. Add a button and Change its Text to "Find Text".
6. Double Click on the button and add the following code in button_click event.

```
textBox1.Select(textBox1.Text.IndexOf(textBox2.Text), textBox2.Text.Length);  
textBox1.Focus();
```

7. Run the program.
8. Enter a line of text in first textbox and a part of that line in 2nd textbox.

Hit the button and desired text got highlighted in first textbox.

About The Author

Rohit Garg started programming in C# Language last year and has successfully accomplished couple of Real Time Applications. He has his expertise in other technologies too, which include Adobe products like Photoshop, Illustrator, Dreamweaver, Flash, C# .NET, SQL Server, Oracle, etc.

He is currently pursuing bachelor's degree in Information Technology from PEC, University of Technology, Chandigarh, India. He has been doing regular research work on .NET technologies and writes on HowToIdeas.net as his hobby and to earn his pocket money. You can connect with him at rohit@howtoideas.net.