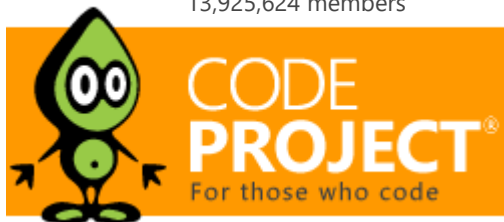


13,925,624 members

1 **646** Member 14164976



[home](#) [articles](#) [quick answers](#) [discussions](#) [features](#) [community](#) [help](#)

Search for articles, questions, tips



Ask a Question

[All Questions](#) [All Unanswered](#) [Mine](#) [FAQ](#)

Follow



## Clear all textboxes in form with one Function

**See more:** [★ C#](#) [★ .NET](#) [★ Visual-Studio](#) [★ VS2010](#) [★ OOP](#)

Rate: ★★★★★

I have 3 Windows forms let Suppose

Form1

Form2

Form3

Form1 has 3 textBox Control

Form2 has 6 textBox Control

Form3 has 14 textBox Control

I want a Function that Accepts form object and clear all textboxes in that form.

I use this code but its not working (No Error, Just No Output).

[Hide](#) [Copy Code](#)

```
public void ClearTextBoxes(Form form)
{
    foreach (Control control in form.Controls)
    {
        if (control.GetType() == typeof(TextBox))
        {
```

### Top Experts

#### Last 24hrs

[OriginalGriff](#)

140

[Dave Kreskowiak](#)

75

[Patrice T](#)

50

[Gerry Schmitz](#)

40

#### This month

[OriginalGriff](#)

1,189

[Richard Deeming](#)

605

[Gerry Schmitz](#)

345

[F-ES Sitecore](#)

290

```
        control1.Text = "";  
    }  
}  
  
//Calling this Function  
var fm1 = new Form1();  
ClearTextBoxes(fm1);  
  
//Smilarly From Form2 and So on  
var fm2 = new Form2();  
ClearTextBoxes(fm2);
```

Please Help me so i can acomplish this task..

**Posted** 31-Aug-14 0:25am  
Ghalib Mirza



Add a Solution

## Comments

Thomas Daniels 30-Aug-14 13:47pm [Reply](#)

Do you have any nested TextBoxes (TextBoxes in other controls such as a Panel)?

Ghalib Mirza 30-Aug-14 14:04pm [Reply](#)

Yes I have Group Box as Parent of textboxes..

Have a Question or Comment?

## Also Read

**Public, Private, and Hybrid  
Cloud: What's the difference?**

## Related Questions

[clear all textboxes except one.](#)

[cleared the textbox value](#)

[textbox clear in forms](#)

[How do I clear a textbox of one form by clicking a button on a different form in C#](#)

[code for clearing all textbox in a form c#](#)

[How to call clear screen function](#)

[unable to clear textbox](#)

[How to clear textbox on postback](#)

[clear all textboxes using a single function](#)

[How to clear cache memory for login form?](#)

## 4 solutions

Top Rated

Most Recent

## Solution 1

Rate: ★★★★★ 

In your comment, you said:

**Quote:**

Yes I have Group Box as Parent of textboxes..

Your method does not look at nested controls (controls inside controls). Try a recursive solution:

[Hide](#) [Copy Code](#)

```
public void ClearTextboxes(Control.ControlCollection ctrlCollection)
{
    foreach (Control ctrl in ctrlCollection)
    {
        if (ctrl is TextBoxBase)
        {
            ctrl.Text = String.Empty;
        }
        else
        {
            ClearTextboxes(ctrl.Controls);
        }
    }
}
```

To use the method on your form, use this code:

[Hide](#) [Copy Code](#)

```
ClearTextboxes(this.Controls);
```

**Posted** 31-Aug-14 1:09am **Updated** 31-Aug-14 16:11pm [v4](#)  
[Thomas Daniels](#)

## Comments

Ghalib Mirza 30-Aug-14 14:31pm [Reply](#)

Thankyou Sir "@ProgramFOX",  
Your example is working absolutely fine for me..  
This encourages me and enhance my knowledge. Thnx again..

Ghalib Mirza 30-Aug-14 14:48pm [Reply](#)

Sorry to disturb you sir..  
i wanna ask you one more thing if there is a ComboBox insted of a TextBox and what if i want all combo boxes set to Zero Index...?

Sergey Alexandrovich Kryukov 31-Aug-14 1:58am [Reply](#)

How about doing a bit by yourself using the idea. Do you know how many classes in .NET FCL? Are you going to ask the same question on different classes?  
And better use Solution 2, it is more correct, especially if you take into account my comments.  
—SA

Thomas Daniels 31-Aug-14 5:13am [Reply](#)

You can actually do almost the same; just replace **TextBoxBase** with **ComboBox** and inside the if statement, set the index to zero. Note that you need to cast the Control to a ComboBox before you can do that.

Sergey Alexandrovich Kryukov 31-Aug-14 1:54am [Reply](#)

Voted 4. It's not good to compare by System.Type, because it would filter out objects of the classes derived from TextBoxBase. Solution 2 is better (is of "is" also, use of string.Empty,), but still there is one problem.  
—SA

Thomas Daniels 31-Aug-14 5:11am [Reply](#)

Thanks for your comments; I improved my answer.

Ghalib Mirza 31-Aug-14 7:43am [Reply](#)

Thnx for improving... Now this code is also Going perfect for me..  
Hence, Thanx Both of you Sir @ProgramFox and Sir @DineshMaind..  
Now its Hard for me to select Efficient method that requires less memory. Kindly Guide me through the process

Thomas Daniels 31-Aug-14 9:32am [Reply](#)

I have no idea which one requires less memory, but .NET has automatic memory management so I don't think you should care too much about it.

Sergey Alexandrovich Kryukov 31-Aug-14 13:15pm

Reply

Sed then... :-)

—SA

Thomas Daniels 31-Aug-14 13:16pm

Reply

Thank you!

Sergey Alexandrovich Kryukov 31-Aug-14 13:42pm

Reply

However... still could be better.

Also, your use of "is" is less than optimal; and having more than one child control type needs generic.  
Please see my answer.

—SA

Have a Question or Comment?

## Solution 2

Rate: ★★★★★ 

If you have nested controls you have to make it recursive try following function.

Hide Copy Code

```
public void EmptyFormControls(Control control)
{
    if (control is TextBox)
    {
        ((TextBox)control).Text = string.Empty;
    }

    for (int i = 0; i < control.Controls.Count; i++)
    {
        EmptyFormControls(control.Controls[i]);
    }
}
```

You can call it as follows.

```
Form1 obj = new Form1();  
EmptyFormControls(obj);
```

**Posted** 31-Aug-14 1:19am  
[DineshMaind](#)

## Comments

[Ghalib Mirza](#) 30-Aug-14 14:35pm [Reply](#)

Sir @DineshMaind Sorry to say you, there is a logical problem in your code your code is not working for me. Hence i would analyse your code and make it correct thnx for your kind response...

[Sergey Alexandrovich Kryukov](#) 31-Aug-14 2:00am [Reply](#)

Where do you see a problem?  
—SA

[Sergey Alexandrovich Kryukov](#) 31-Aug-14 1:57am [Reply](#)

This solution is much better than Solution 1, but I voted 4, too: you should have used the class System.Windows.Forms.TextBoxBase, not System.Windows.Forms.TextBox, because the classes which could be considered text box classes are derived from TextBoxBase, moreover, it is recommended to use TextBoxBase as an immediate base class (which is designed to be used that way), not TextBox.

Besides, the loop will be simpler if you change it to foreach.

—SA

[Ghalib Mirza](#) 31-Aug-14 2:43am [Reply](#)

@Sergey Alexandrovich This code is not working for me even i ammend this code as:

```
public void EmptyFormControls(Control control)  
{  
    if (control is TextBoxBase)  
    {  
        ((TextBoxBase)control).Text = string.Empty;  
    }  
  
    for (int i = 0; i < control.Controls.Count; i++)
```

```
{  
    EmptyFormControls(control.Controls[i]);  
}  
}
```

```
///  
Calling this function  
var obj = new Form1();  
EmptyFormControls(obj);
```

Still its not working..

Have a better solution Kindly provide me with code examples.. Thnx

---

**DineshMaind** 31-Aug-14 3:09am

Reply

If you want to use it for your current form just call EmptyFormControls(this); it will work definitely. As I am using it in my live projects...

---

**Ghalib Mirza** 31-Aug-14 7:37am

Reply

Yes now its works fine for both textboxes and Comboboxes Thanx Sir @Dinesh and Sir @Sergey Alexandrovich thnx for making me correct..

---

**Sergey Alexandrovich Kryukov** 31-Aug-14 13:14pm

Reply

You are welcome.  
—SA

---

**DineshMaind** 31-Aug-14 13:38pm

Reply

Thanks @Sergey Alexandrovich Sir for suggesting much better solution by using System.Windows.Forms.TextBoxBase class.

---

**Sergey Alexandrovich Kryukov** 31-Aug-14 14:54pm

Reply

You are welcome.  
You you then also accept Answer 3 formally?  
—SA

---

**Sergey Alexandrovich Kryukov** 31-Aug-14 13:41pm

Reply

Also, your use of "is" is less than optimal; and having more than on child control type needs generic.  
Please see my answer.  
—SA

[Have a Question or Comment?](#)

## Solution 3

Rate: ★★★★★ **Ghalib Mirza wrote:**

Yes now its works fine for both textboxes and Comboboxes Thanx Sir @Dinesh and Sir @Sergey Alexandrovich thnx for making me correct...

Well, no, then the answers published so far are not universal enough. If you really need some action for those two or more different types, you rather need one common generic method. I wrote it for you:

[Hide](#) [Copy Code](#)

```
using System;
using System.Windows.Forms;

//...

static void ClearControls<CTRL>(Control parent, Action<CTRL> clearMethod) where CTRL: Control
{
    foreach (Control child in parent.Controls) {
        CTRL targetChild = child as CTRL;
        if (targetChild != null)
            clearMethod(targetChild);
        ClearControls(child, clearMethod);
    } //Loop
} //ClearControls
```

Usage:

[Hide](#) [Copy Code](#)

```
static void Test(Control parent) {
    ClearControls<TextBox>(parent, new Action<TextBox>((child) => {
        child.Text = string.Empty;
    }));
    ClearControls<ComboBox>(parent, new Action<ComboBox>((child) => {
        if (child.Items.Count > 0)
            child.SelectedIndex = 0;
    }));
} //test
```



Note the use of the operator **as**, instead of **is**. If is good when you need both type check and type cast: the type check actually is the comparison of the type cast result with null.

Of course, the action can be anything of this profile, not only "clear". You can also add parameters, for something else.

—SA

**Posted** 1-Sep-14 0:39am  
[Sergey Alexandrovich Kryukov](#)

**Updated** 1-Sep-14 0:44am [v3](#)

## Comments

[Thomas Daniels](#) 31-Aug-14 13:44pm

Reply

+5, good answer!

**Sergey Alexandrovich Kryukov** 31-Aug-14 14:52pm

Reply

Thank you very much.  
—SA

[DineshMaind](#) 31-Aug-14 13:53pm

Reply

my +5, very nice and generalized solution. Highly useful for me.

**Sergey Alexandrovich Kryukov** 31-Aug-14 14:52pm

Reply

Thank you very much.  
—SA

[Ghalib Mirza](#) 1-Sep-14 6:47am

Reply

I m speechless for your Solution .. amazing solution.. but unfortunately i know nothing about Generic Class (Sad but true). The word Generic is new for me, now i add it to "Have to Learn" Sticky note..  
Thanx for the solution.

**Sergey Alexandrovich Kryukov** 1-Sep-14 11:18am

Reply

You cannot do .NET development without at least just using them (which is really easy though), and you cannot do good programming without defining them, but it's not at all hard to learn. Especially if it optimizes and ease up your work.

—SA

DineshMaind 1-Sep-14 7:21am [Reply](#)

You did right thing now. This solution actually deserves best solution ever.

Sergey Alexandrovich Kryukov 1-Sep-14 11:19am [Reply](#)

Thank you, Dinesh.

—SA

Ghalib Mirza 1-Sep-14 7:24am [Reply](#)

Thanx

Hope it would be a best one!

Sergey Alexandrovich Kryukov 1-Sep-14 11:20am [Reply](#)

You are welcome.

Will you accept the answer formally them (green "Accept" button)?

—SA

Ghalib Mirza 1-Sep-14 14:58pm [Reply](#)

Off Course i accept it...

i have checked it, works perfect..though i not really understand the code... :(

Sergey Alexandrovich Kryukov 1-Sep-14 15:16pm [Reply](#)

Very good. In this code the following features are used: 1) generic method, 2) anonymous method used as the actual parameters for clearMethod, 2a) more exactly, anonymous method in the lambda form, 3) Action is the declaration from .NET FCL, System.Action (several declarations with different numbers of parameters). Just read on the topic, it's pretty easy to understand, especially in terms of usage. Read the whole C# manual with related .NET features (I don't say "all" of .NET); it will pay off pretty soon, and if you won't, it can hurt you pretty soon.

Wish you the best of luck.

—SA

Ghalib Mirza 2-Sep-14 1:10am [Reply](#)

Thank you!

I just learn about .NET FCL as soon as i can.. thanx again.

Sergey Alexandrovich Kryukov 2-Sep-14 2:20am [Reply](#)

You are welcome.

—SA

## Solution 4

Rate: ★★★★★👍

To make it really generic to use, I would create a Dictionary to set the clear method for each type, and then have a non-generic method that can clear an entire control hierarchy by using the actions defined in the dictionary. This also gives a clear separation of defining actions and running them.

Hide Expand ▼ Copy Code

```
//...

private Dictionary<Type, Action<object>> actions = new Dictionary<Type, Action<object>>
{
    { typeof(TextBox), ctrl => ((TextBox)ctrl).Text = string.Empty },
    { typeof(ComboBox), ctrl => ((ComboBox)ctrl).SelectedIndex = -1 },
};

//...

private void ClearChildControls(Control parent)
{
    foreach (Control child in parent.Controls)
    {
        var controlType = child.GetType();

        if (actions.ContainsKey(controlType))
        {
            actions[controlType](child);
        }


        ClearChildControls(child);
    }
}

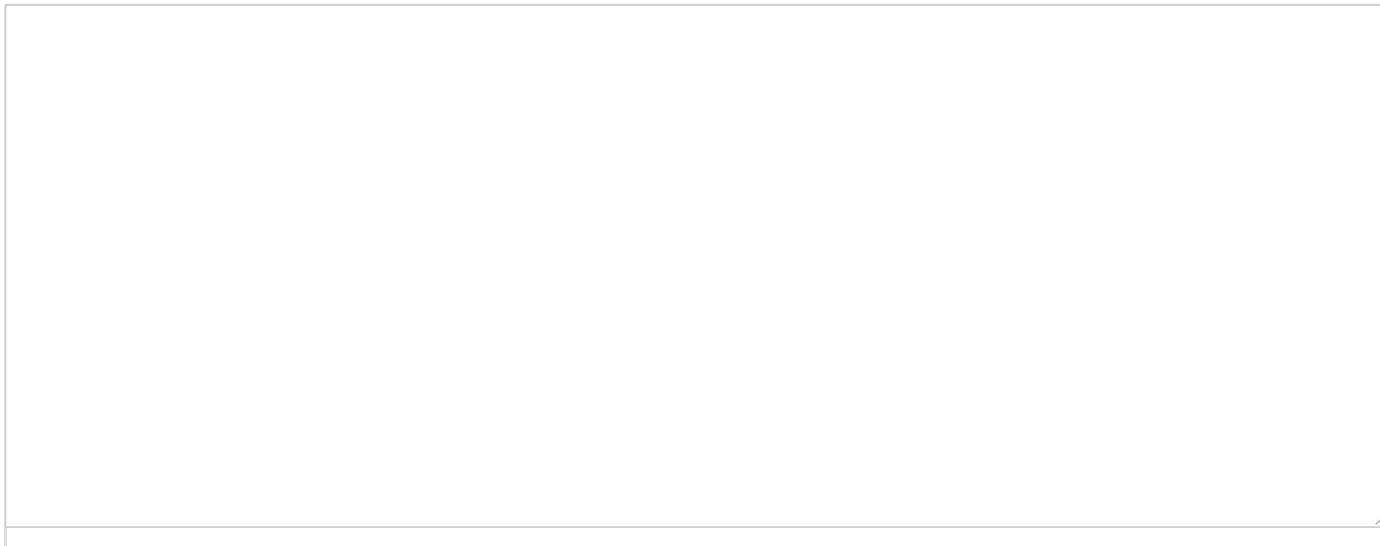
//...
```

**Posted** 1-Sep-14 0:57am  
[kbrandwijk](#)

Have a Question or Comment?

Add your solution here

**B** / U  small BIG code **var** < > & link [^] encode untab case indent outdent



Preview



[Submit your solution!](#)

When answering a question please:

1. Read the question carefully.
2. Understand that English isn't everyone's first language so be lenient of bad spelling and grammar.
3. If a question is poorly phrased then either ask for clarification, ignore it, or **edit the question** and fix the problem. Insults are not welcome.
4. Don't tell someone to read the manual. Chances are they have and don't get it. Provide an answer or move on to the next question.

Let's work to help developers, not make them feel stupid.

This content, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)