

How to clear all form fields from code-behind?

[Ask Question](#)

5



3

HTML has an input button type to reset all fields in a form to their initial state in one step: `<input type="reset" ... />` .

Is there a similar simple way to reset all form fields of an aspx page from code-behind? Or is it necessary to reset all controls one by one with `TextBox1.Text=string.Empty` , `TextBox2.Text=string.Empty` , etc. ?

Thanks in advance!

Update:

Context is a simple Contact/"Send us a message" page with 8 `asp:TextBoxes` on the page (where the user enters the name, address, phone, email, message, etc.). Then he clicks on submit, the `OnClick` message handler in code-behind sends an email to some administrator, and all the form fields the user filled in should be emptied and he gets a notification in a label ("Message sent blabla..."). I want to have the form fields cleared to avoid that the user clicks again on submit and the same message is sent a second time.

[asp.net](#)[webforms](#)

edited Apr 10 '10 at 15:11

asked Apr 10 '10 at 14:52



Slauma

150k

51

359

386

- 1 Depending on what the scenario is, there could be several ways to clear them. Could you explain how you want this to work in your scenario? – [Mikael Svenson](#) Apr 10 '10 at 14:54

@Mikael Svenson: I have added an "Update" to my question to answer your question. Hope this makes things clearer. – [Slauma](#) Apr 10 '10 at 15:12

In that case I propose to turn off the viewstate for the controls you want to clear as I have written in my answer. – [Mikael Svenson](#) Apr 10 '10 at 15:17

5 Answers



You need only write a fork for each type of control unless one of the control has something special that needs to be done to reset it.

11



```
foreach( var control in this.Controls )
{
    var textbox = control as TextBox;
    if (textbox != null)
        textbox.Text = string.Empty;

    var dropDownList = control as DropDownList;
    if (dropDownList != null)
        dropDownList.SelectedIndex = 0;
    ...
}
```

ADDITION You asked how to clear controls even ones that are buried. To do that, you should create a recursive routine like so:

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams

Q&A for work


[Learn More](#)

```
private void ClearControl( Control control )
{
    var textbox = control as TextBox;
    if (textbox != null)
        textbox.Text = string.Empty;

    var dropDownList = control as DropDownList;
    if (dropDownList != null)
        dropDownList.SelectedIndex = 0;
    ...

    foreach( Control childControl in control.Controls )
    {
        ClearControl( childControl );
    }
}
```

So, you would call this by passing the page:

```
ClearControls( this );
```

edited Apr 10 '10 at 18:46

answered Apr 10 '10 at 15:01



Thomas

56.5k 8 80 124

You should probably include some recursion if you have databindings, controls inside templates and custom controls. – [Mikael Svenson](#) Apr 10 '10 at 15:13

Nice simple thing! (will be especially simple in my case since I only have TextBoxes on the page). Thank you! – [Slauma](#) Apr 10 '10 at 15:14

There is a problem: My TextBoxes are inside of the Content element for a ContentPlaceHolder because my page is derived from a MasterPage. The this.Controls collection contains only one element which is the MasterPage itself (or something derived from it) with the ID=ctl00 . I

was searching around in debugger but can't find any collection of objects which represent my TextBoxes. – [Slauma](#) Apr 10 '10 at 17:06

@Slauma - I've amended my response so that you can recursively clear each control. – [Thomas](#) Apr 10 '10 at 18:07

@Thomas: Great, it works! Thanks! (I only had to replace the anonymous var in the foreach loop by `Control` since the compiler complained that `childControl` is an object and not a `Control`). – [Slauma](#) Apr 10 '10 at 18:37

Refer this link for more information

7

<http://www.freshcodehub.com/Article/3/clear-all-fields-like-textbox-dropdownlist-checkbox-radiobutton-label-after-form-submission-in-aspnet-c>

```
public void ClearControls(Control parent)
{
    foreach (Control c in parent.Controls)
    {
        if ((c.GetType() == typeof(TextBox))) //Clear TextBox
        {
            ((TextBox)(c)).Text = "";
        }
        if ((c.GetType() == typeof(DropDownList))) //Clear DropDown
        {
            ((DropDownList)(c)).ClearSelection();
        }
        if ((c.GetType() == typeof(CheckBox))) //Clear CheckBox
        {
            ((CheckBox)(c)).Checked = false;
        }
    }
}
```

```

if ((c.GetType() == typeof(CheckBoxList))) //Clear CheckBoxList
{
    ((CheckBoxList)(c)).ClearSelection();
}
if ((c.GetType() == typeof(RadioButton))) //Clear RadioButton
{
    ((RadioButton)(c)).Checked = false;
}
if ((c.GetType() == typeof(RadioButtonList))) //Clear RadioButtonList
{
    ((RadioButtonList)(c)).ClearSelection();
}
if ((c.GetType() == typeof(HiddenField))) //Clear HiddenField
{
    ((HiddenField)(c)).Value = "";
}
if ((c.GetType() == typeof(Label))) //Clear Label
{
    ((Label)(c)).Text = "";
}
if (c.HasControls())
{
    ClearControls(c);
}
}
}

```

edited Jun 18 '16 at 12:11

answered Jun 26 '15 at 11:41



Kamal Pratap

129 1 3



3

using the manual approach of `String.Empty` for each and every Textbox or any other field will be cumbersome, also by using `Response.Redirect()`; it will be difficult to show any confirmation

message or same. So, on reading so many blogs i have found a reliable approach so far:

```
Public void reset(Control control)
{
    foreach (Control x in control.Controls)
    {
        if (x is TextBox)
        {
            (x as TextBox).Text = String.Empty;
        }
        if (x is DropDownList)
        {
            (x as DropDownList).SelectedIndex = 0;
        }
        .
        .
        reset(x);
    }
}
```

use this code as `reset(this);` in your page wherever you want to reset or clear the values. At end of the `if` conditions do not forget to use the function recursively using the same `Control` object `x`.

answered Aug 1 '12 at 19:56



Shaminder S Aujla

1,133 2 15 27

2

Using `form.Controls.Clear()` is not such a good approach because it will clear the entire form and you will even lose all the buttons on the form.

Instead if you just want to clear all the form fields like text fields and radio buttons I would recommend you try the following: If you have a Reset button "Button1" then on click call a function `reset();`

In the reset function:

```
protected void resetButton_Click(object sender, EventArgs e)
{
    TextBox1.Text=""; //set equal to empty string to all fields
}
```

Or redirect to same page by terminating the previous page

```
protected void resetButton_Click(object sender, EventArgs e)
{
    Response.Redirect("~/Test2.aspx", true);
}
```

edited Aug 1 '12 at 3:41



Jack

10.1k 12 42 64

answered Aug 1 '12 at 2:49



JASKIRAT KHEHRA

21 1



1

For your scenario the easiest way to clear the fields, in my opinion, is to turn off the ViewState (`EnableViewState=false`) of the controls you want to appear blank after the submit.



Or perhaps for the whole page unless there is some state you need.


answered Apr 10 '10 at 15:15



Mikael Svenson

32.3k 4 61 68

Hm, I've just tested it (for one of TextBoxes and also the option to turn off the ViewState of the whole page). But it doesn't work: All TextBoxes are

still filled after the postback. – [Slauma](#) Apr 10 '10 at 15:27 

If you do a regular postback (not ajax) with `EnabledViewState=false` on the page or a control, there should be no way those fields should have values, unless you set them in code somewhere. – [Mikael Svenson](#) Apr 10 '10 at 15:43

I've created a test web site (one TextBox+one Button, `EnableViewState="false"` on page level and also for the TextBox): Same behaviour, Values in the TextBox are preserved after PostBack. I've googled a bit and found explanations: 1) dotnetfunda.com/articles/... and 2) geekswithblogs.net/AaronLi/archive/2007/05/20/112615.aspx – [Slauma](#) Apr 10 '10 at 16:22

1 @Slauma Learn something new every day :) – [Mikael Svenson](#) Apr 10 '10 at 18:14
