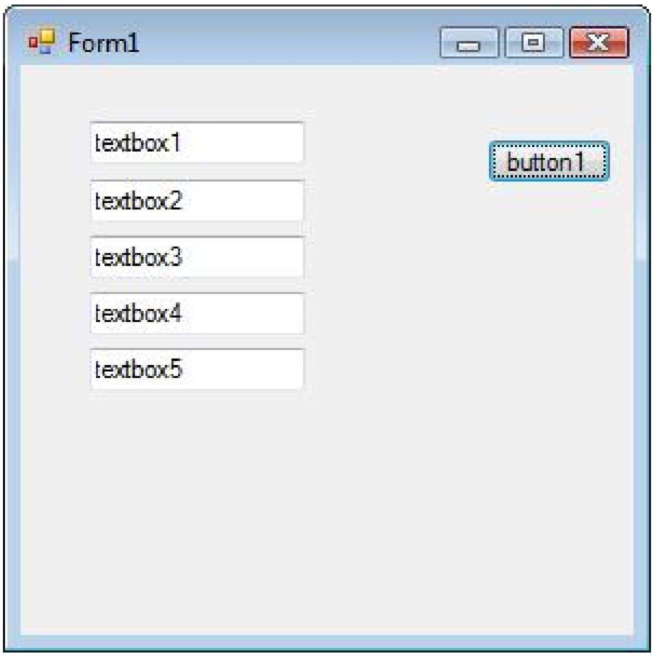
How do you return the focus to the last used control after clicking a button in a winform app?

Asked 16 years, 3 months ago Modified 10 years, 6 months ago Viewed 13k times

7

I'm working on a windows forms application (C#) where a user is entering data in a form. At any point while editing the data in the form the user can click one of the buttons on the form to perform certain actions. By default the focus goes to the clicked button so the user has to click back on to the control they want to edit in order to continue modifying the data on the form. What I need to be able to do is return the focus to the last edited control after the button click event has been processed. Here's a sample screenshot that illustrates what I'm talking about:



The user can be entering data in textbox1, textbox2, textbox3, etc and click the button. I need the button to return the focus back to the control that most recently had the focus before the button was clicked.

I'm wondering if anyone has a better way of implementing this functionality than what I've come up with. Here's what I'm doing right now:

```
public partial class Form1 : Form
 {
     Control _lastEnteredControl;

     private void textBox_Enter(object sender, EventArgs e)
     {
         _lastEnteredControl = (Control)sender;
     }

     private void button1_Click(object sender, EventArgs e)
```

```
        {
            MessageBox.Show("Do something here");
            _lastEnteredControl.Focus();
        }


    }
```

So basically what we have here is a class variable that points to the last entered control. Each textbox on the form is setup so the textBox_Enter method is fired when the control receives the focus. Then, when the button is clicked focus is returned to the control that had the focus before the button was clicked. Anybody have any more elegant solutions for this?

c# .net winforms

Share

Improve this question

Follow

edited Feb 8, 2017 at 14:07

Community Bot

1 ● 1

asked Sep 1, 2008 at 0:48

Joe Barone

3,140 ● 3 ● 24 ● 20

7 Answers

Sorted by: Highest score (default)

16

✓

For a bit of 'simplicity' maybe try.

```
public Form1()
    {
        InitializeComponent();

        foreach (Control ctrl in Controls)
        {
            if (ctrl is TextBox)
            {
                ctrl.Enter += delegate(object sender, EventArgs e)
                              {
                                  _lastEnteredControl = (Control)sender;
                              };
            }
        }
    }
```

then you don't have to worry about decorating each textbox manually (or forgetting about one too).

Share Improve this answer Follow

answered Sep 1, 2008 at 1:11

Joshua Turner

1,049 ● 2 ● 10 ● 16

2

You could do the following

Change the button to a label and make it look like a button. The label will never get focus and you don't have to do all the extra coding.

Share Improve this answer Follow

answered Nov 6, 2008 at 16:52

eric

1

I think what you're doing is fine. The only thing I could think of to improve it would be to store each control into a stack as they are accessed. That would give you a complete time line of what was accessed.

Share Improve this answer Follow

answered Sep 1, 2008 at 0:51

Ethan Gunderson

11.4k ● 8 ● 31 ● 29

1

Your approach looks good. If you want to avoid having to add an the event handler to every control you add, you could create a recursive routine to add a GotFocus listener to every control in your form. This will work for any type of control in your form, however you could adjust it to meet your needs.

```
private void Form_OnLoad(object obj, EventArgs e)
{
    AddGotFocusListener(this);
}

private void AddGotFocusListener(Control ctrl)
{
    foreach(Control c in ctrl.Controls)
    {
        c.GotFocus += new EventHandler(Control_GotFocus);
        if(c.Controls.Count > 0)
        {
            AddGotFocusListener(c);
        }
    }
}

private void Control_GotFocus(object obj, EventArgs e)
{
    // Set focused control here
}
```

Share Improve this answer Follow

answered Sep 1, 2008 at 1:07

Dan Herbert

103k ● 51 ● 192 ● 221

0

Your implementation looks good enough -- what I do want to know is why you want to do this in the first place? Won't it be preferrable for the focus to cycle back to the first entry? Is the data in the last text box so malleable that once they click the button it is "remembered"? Or do you have some sort of operation that the button does to that specifici text box data -- in that case shouldn't the focus go to a subsequent control instead?

I'm interested in finding out why you want to do this in the first place.

Share Improve this answer Follow

answered Sep 1, 2008 at 0:51

Jon Limjap

95.3k ● 15 ● 103 ● 153

Yeah, I admit the requirement is a bit unusual. Some of the information that the users will be entering into this application exists in scans of old documents that are in a couple of different repositories. The buttons facilitate finding and opening these old

0

docs. It's difficult to predict where the users will be on the form when they decide to pull up a document with more information to enter on the form. The intent is to make the UI flow well in spite of these funky circumstances.

Share Improve this answer Follow

answered Sep 1, 2008 at 1:13

Joe Barone

3,140 ● 3 ● 24 ● 20

0

Create a class called CustomTextBox that inherits from TextBox. It has a static variable called stack. When the textbox loses focus push onto the stack. When you want to find the last focused control then just pop the first item from the stack. Make sure to clear the static Stack variable.

Share Improve this answer Follow

answered Jun 18, 2014 at 19:14

Denis

12.1k ● 16 ● 93 ● 155