Visual Web Development

Terry Marris September 2007

2 Text Input Output

We see how a user may enter text, and how the text along with a message may be displayed.

2.1 Use Case

A use case describes the interaction between a system and its user.

Use Case	TextIO
Goal	to display a personalised greeting
Pre-condition	none
Post-condition	a personalised greeting is displayed
Initiating Actor	the user

Main Success Scenario

1 system prompts user for their identity or name
2 user enters their id or name
3 system displays a greeting
4 exit success

The use case is named TextIO. TextIO represents the system, e.g. a program.

The goal is the expected outcome. The expected outcome is that a personalised greeting is displayed.

Pre-conditions are what must be true if the post-conditions are to be met. In our example there are no explicit pre-conditions.

Post-conditions are what must be true if the pre-conditions are met. The post-condition here is that a greeting is displayed.

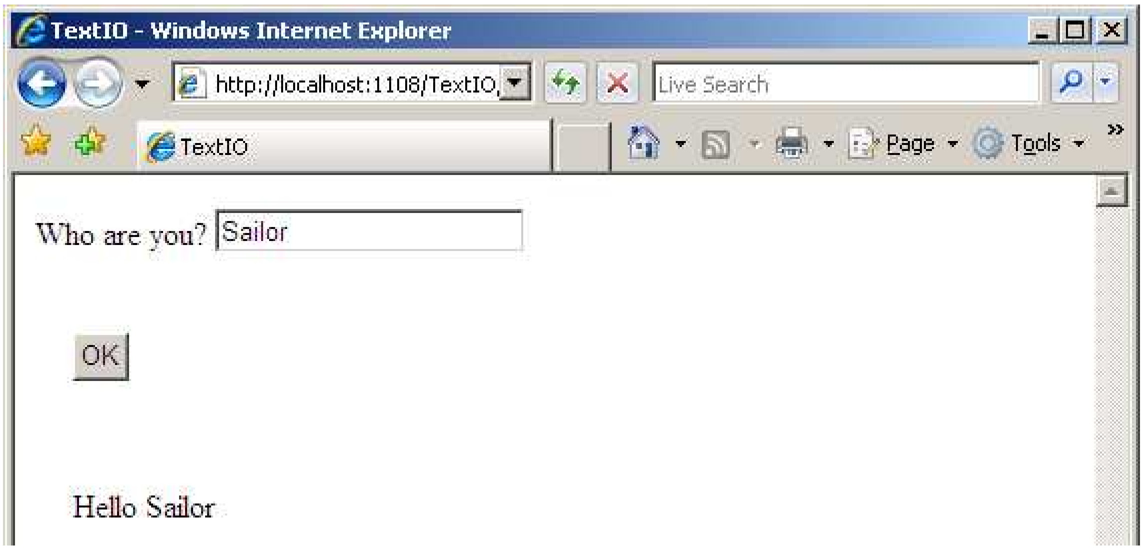
The initiating actor is the person interacting with the system. The initiating actor could be another system, but in our example it is the human user.

The main success scenario shows the step-by-step actions that accomplish the goal.

Use cases are the essential first step in planning and writing computer programs. Without a use case (in some form or another) you have no idea what to do. They can be as detailed, or as sparse as you like.

2.2 Required Result

The result we are looking for is something like:


TextIO - Windows Internet Explorer
http://localhost:1108/TextIO
Live Search
TextIO
Page
Tools

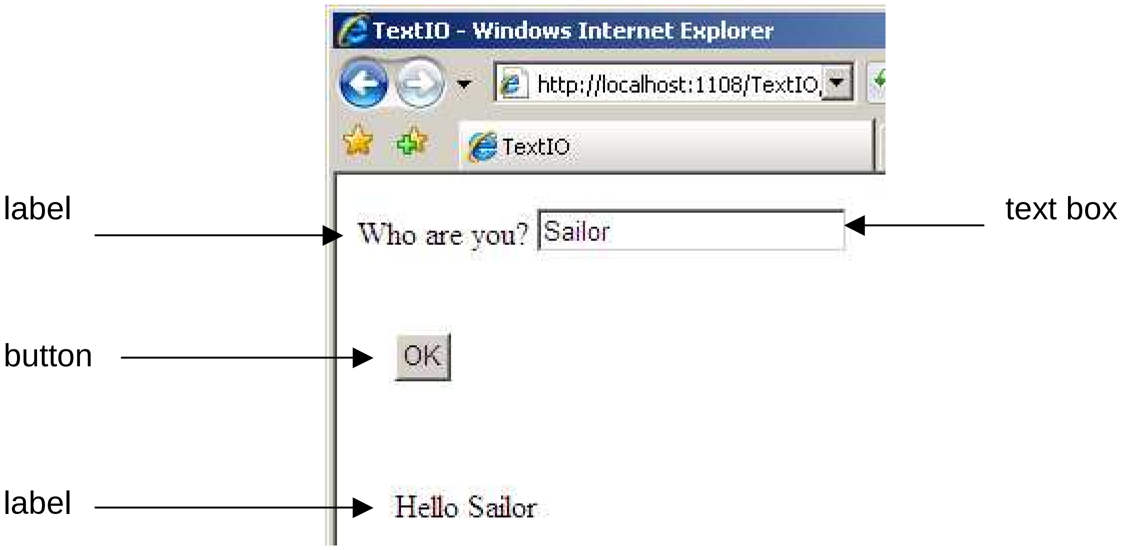
Who are you? Sailor

OK

Hello Sailor

The prompt "Who are you?" is displayed. The user enters "Sailor" and clicks OK. The computer then displays "Hello Sailor".

2.3 The Controls


label
button
label
text box
TextIO - Windows Internet Explorer
http://localhost:1108/TextIO
TextIO
OK

Who are you? Sailor

Hello Sailor

We use:

- Labels for text output
- Text boxes for text input (and output)
- Buttons for click events. For example, when the user clicks OK the text they entered is added onto "Hello" and displayed.

2.4 Properties and Values

Control	Property	Value
Form	File Name Title	TextIO.aspx TextIO
Label	ID Text	lblPrompt Who are you?
Text Box	ID	txtName
Button	ID Text	btnOK OK
Label	ID Text	lblGreeting <blank>

Notice how:

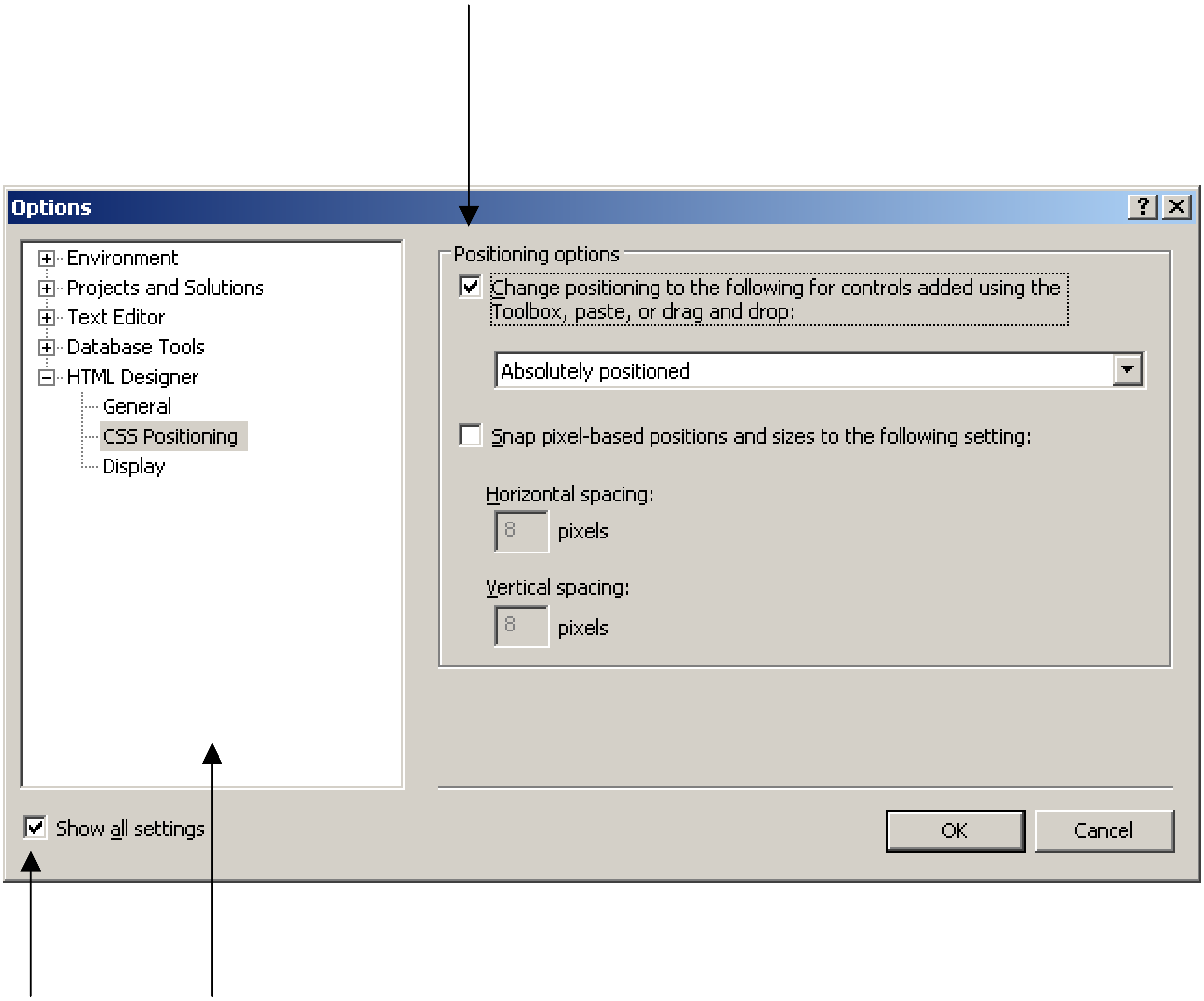
- label ids begin with lbl
- text box ids begin with txt
- button ids begin with btn

Notice also that the names chosen for ids are descriptive i.e. convey their purpose.

2.5 Form Layout

Drag and drop controls onto the form as shown in the diagram in §2.3 above. If the controls remain on one line and cannot be dragged to where you want them select:

- Layout, Position, Auto-position options
- Ensure Show All Settings is selected
- Set Positioning options as shown below. OK.



Options

- Environment
- Projects and Solutions
- Text Editor
- Database Tools
- HTML Designer
  - General
  - CSS Positioning
  - Display

Positioning options

☑ Change positioning to the following for controls added using the Toolbox, paste, or drag and drop:

Absolutely positioned

□ Snap pixel-based positions and sizes to the following setting:

Horizontal spacing: 8 pixels

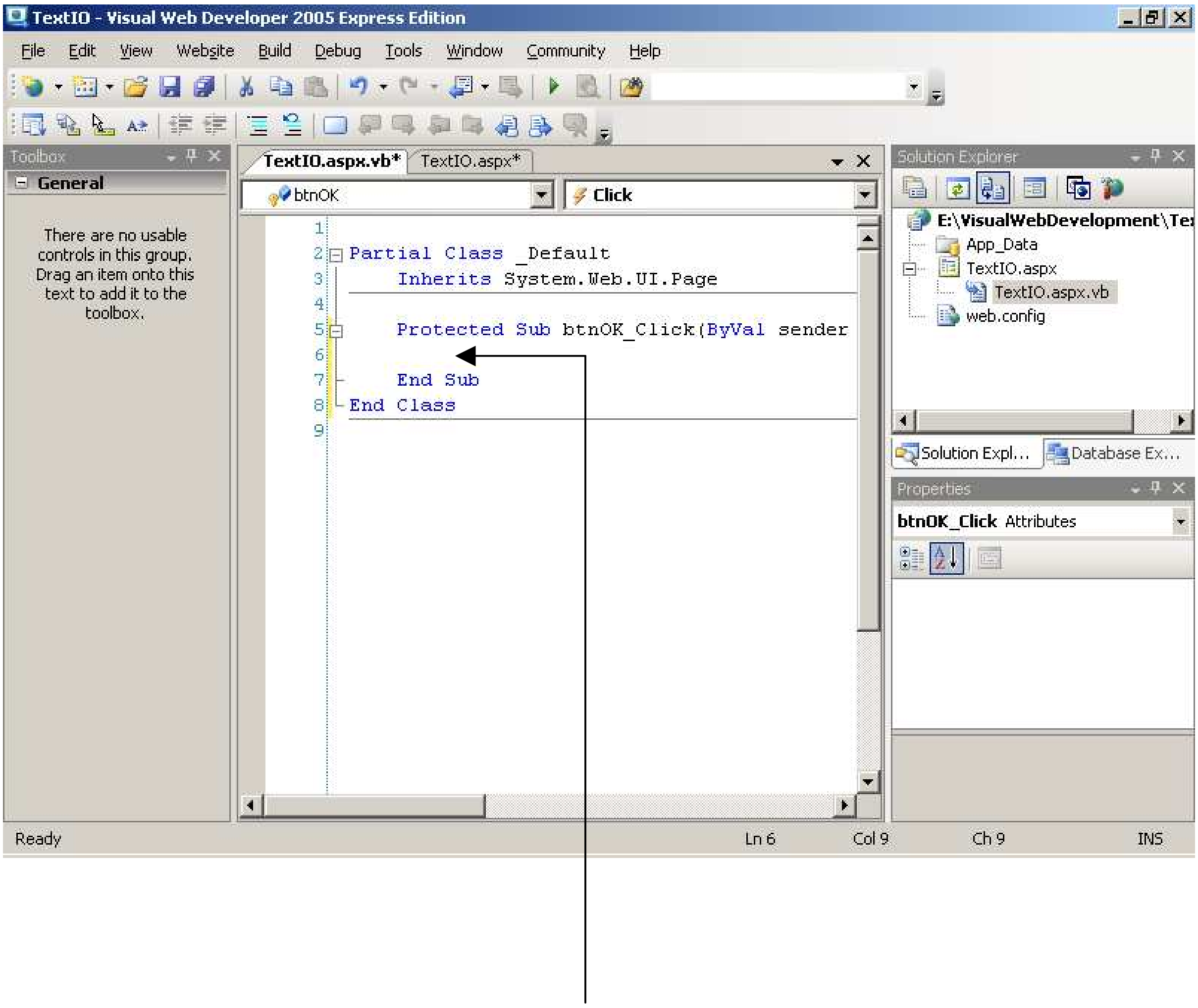
Vertical spacing: 8 pixels

☑ Show all settings

Set the properties for each control as shown in the table in §2.4 above.

2.6 The Code Behind the Button

Rapid double click on the OK button. A code window opens (eventually!) onto the Sub btnOK_Click procedure.

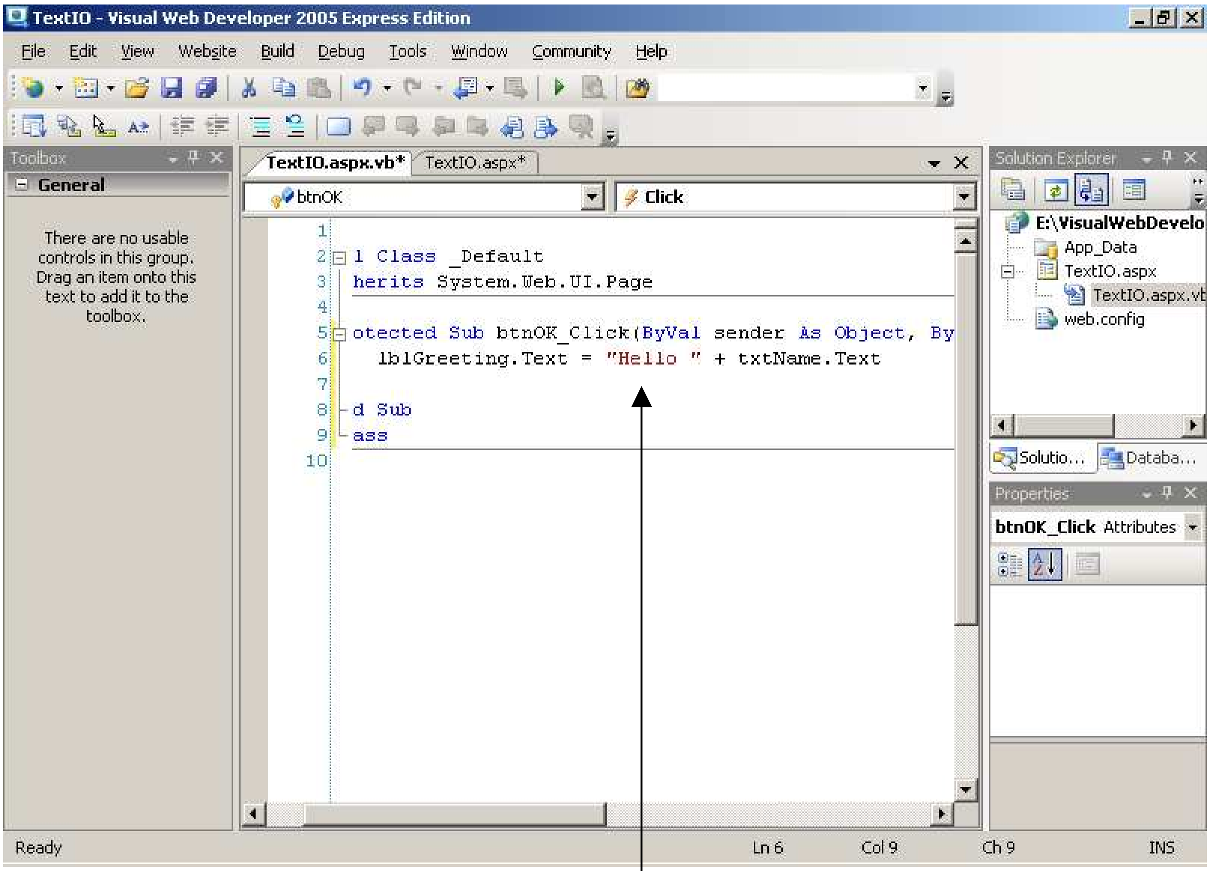


There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

```

Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub btnOK_Click(ByVal sender

    End Sub
End Class

```

We enter VB code to instruct the computer on what to do when the button is clicked:

- take the value entered by the user in the name text box
- add this value onto the end of Hello
- display the result in the greeting label



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

```
l Class _Default
herits System.Web.UI.Page

otected Sub btnOK_Click(ByVal sender As Object, By
  lblGreeting.Text = "Hello " + txtName.Text

d Sub
ass
```

The Visual Basic line

lblGreeting.Text = "Hello " + txtName.Text

works from right to left.

The = sign is known as the assignment operator and copies the value on its right into the container on its left.

In this context, the + symbol is known as the concatenation operator; its joins the text on its right onto the end of the text on its left. Notice the space after Hello and before the quotation mark; this guarantees a space after the word Hello when the program runs.

2.7 Exercise

1. Try out the program, shown above, that prompts the user to enter their name and then displays a personal greeting.