

ASP VALIDATION

Rehab AlFallaj

CT1501: Development of Internet Application

VALIDATION

- Why?
 - To check that the information users enter is valid.
- ASP.NET provides a set of validation controls that provide an easy-to-use but powerful way to check for errors and, if necessary, display messages to the user.

TYPES OF VALIDATION FOR ASP.NET SERVER CONTROLS

<i>Type of validation</i>	<i>Control to use</i>	<i>Description</i>
Required entry	RequiredFieldValidator	Ensures that the user does not skip an entry
Comparison to a value	CompareValidator	Compares a user's entry against a constant value, against the value of another control (using a comparison operator such as less than, equal, or greater than), or for a specific data type.
Range checking	RangeValidator	Checks that a user's entry is between specified lower and upper boundaries. You can check ranges within pairs of numbers, alphabetic characters, and dates.
Pattern matching	RegularExpressionValidator	Checks that the entry matches a pattern defined by a regular expression. This type of validation enables you to check for predictable sequences of characters, such as those in e-mail addresses, telephone numbers, postal codes, and so on.
User-defined	CustomValidator	Checks the user's entry using validation logic that you write yourself. This type of validation enables you to check for values derived at run time.

CONT'.

- ⦿ A related control, the **ValidationSummary** control, does not perform validation, but is often used in conjunction with other validation controls to display the **ErrorMessage**s from all the validation controls on the page together.

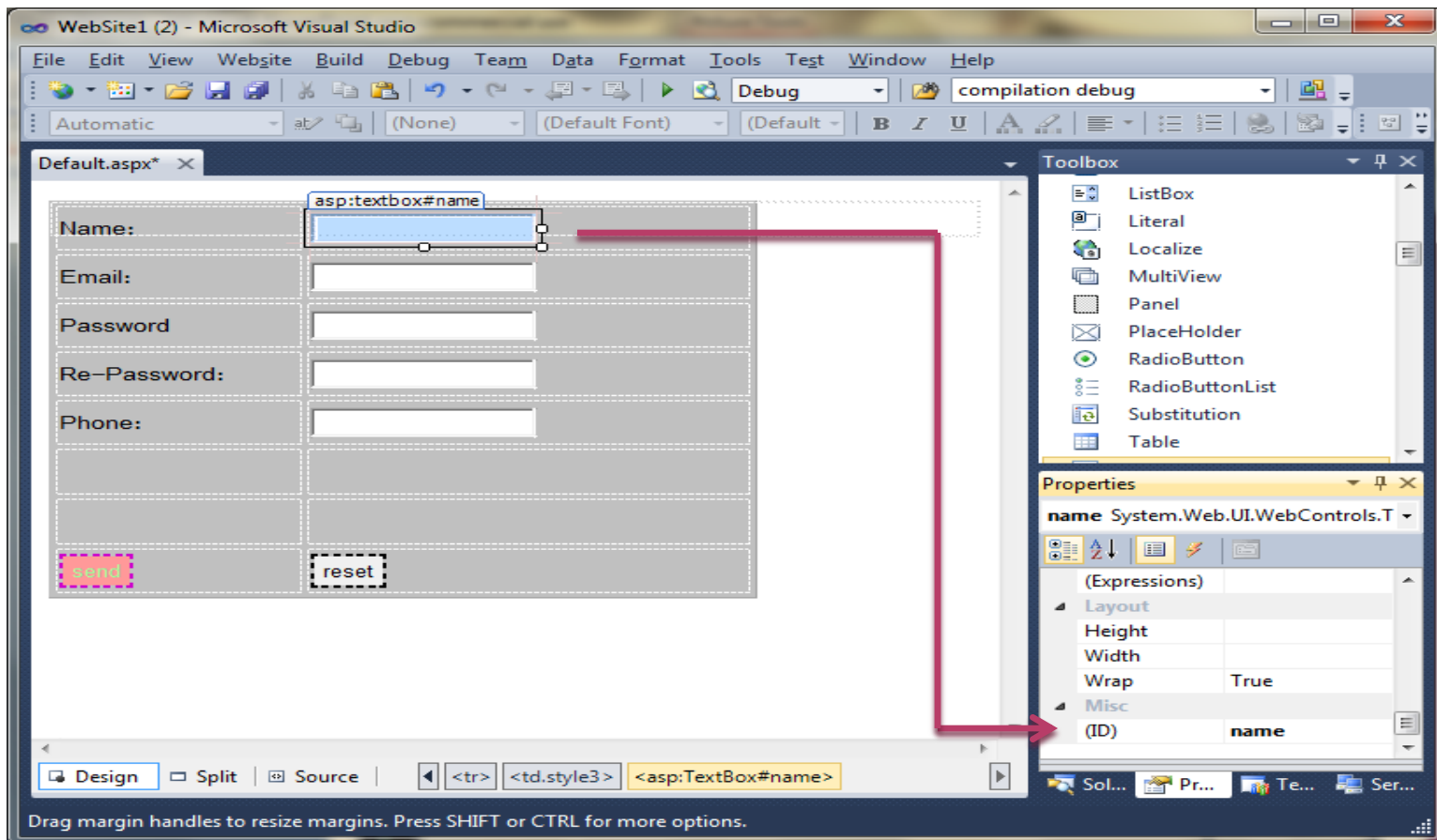
STEPS

○ Build the form

The screenshot displays the Microsoft Visual Studio interface for a web application. The main window shows a design view of a form titled "Default.aspx". The form contains several input fields: "Name:", "Email:", "Password:", "Re-Password:", and "Phone:". At the bottom of the form, there are two buttons labeled "send" and "reset". The right-hand side of the interface features a "Toolbox" with various web controls, including "TextBox", which is currently selected. Below the toolbox is the "Properties" window, showing the "DOCUMENT" properties for the selected element. The bottom status bar indicates the current design mode is "Design" and shows the underlying HTML structure: `<html> <body> <form#form1> <div> <table.style1> <tr> <td.style2>`. The system tray at the bottom left shows the time as 9:32 PM on 2/17/2013.

STEPS

- Check that all the textBox have the same name as the label in which it links to.



STEPS

The screenshot shows Microsoft Visual Studio with a web form in Design view. The form contains the following elements:

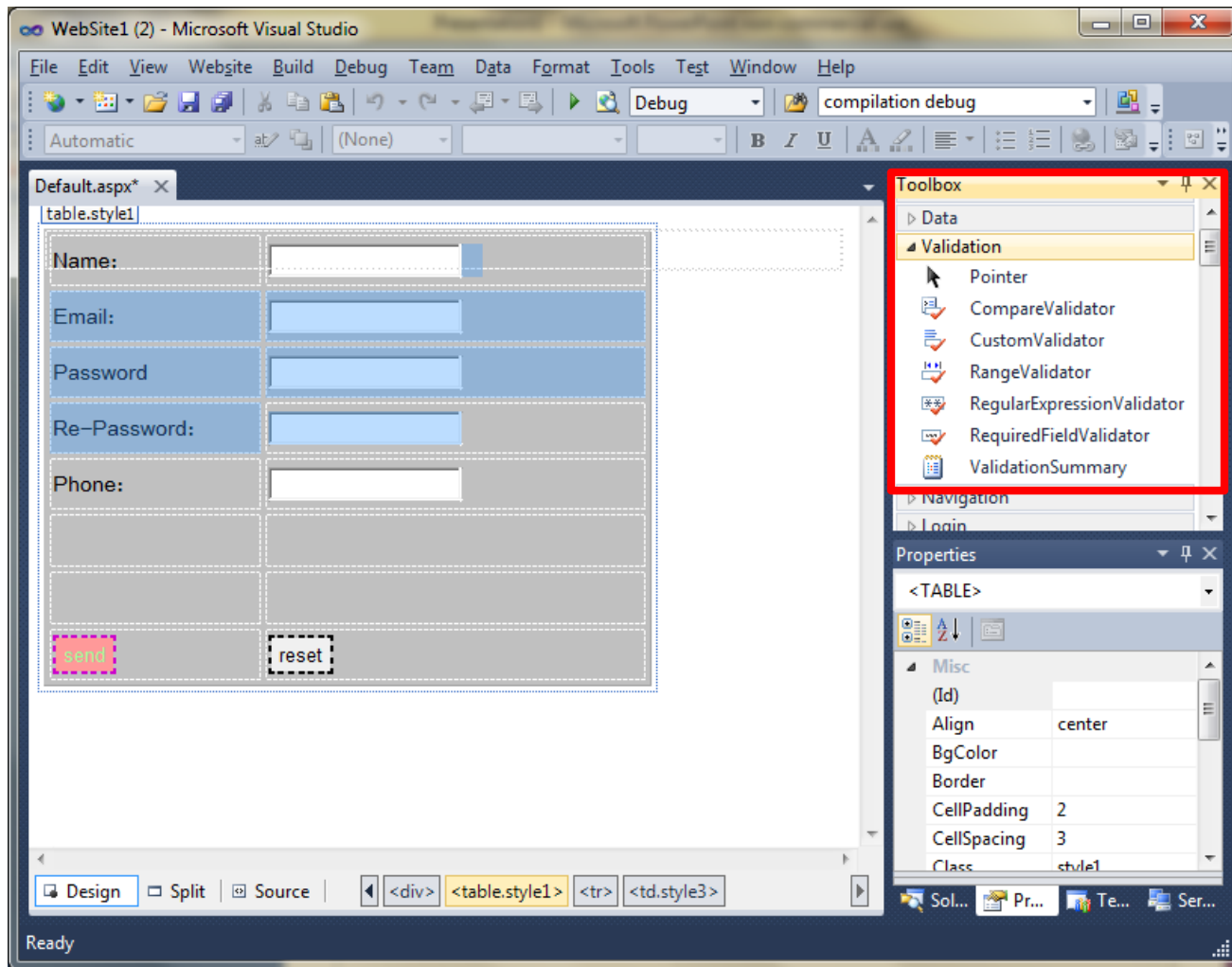
- Name:
- Email: (ID: `asp:textbox#Email`)
- Password:
- Re-Password:
- Phone:
- Buttons: and

The Properties window on the right shows the selected **Email** control with the following properties:

Property	Value
Layout	
Height	
Width	
Wrap	True
Misc	
(ID)	Email

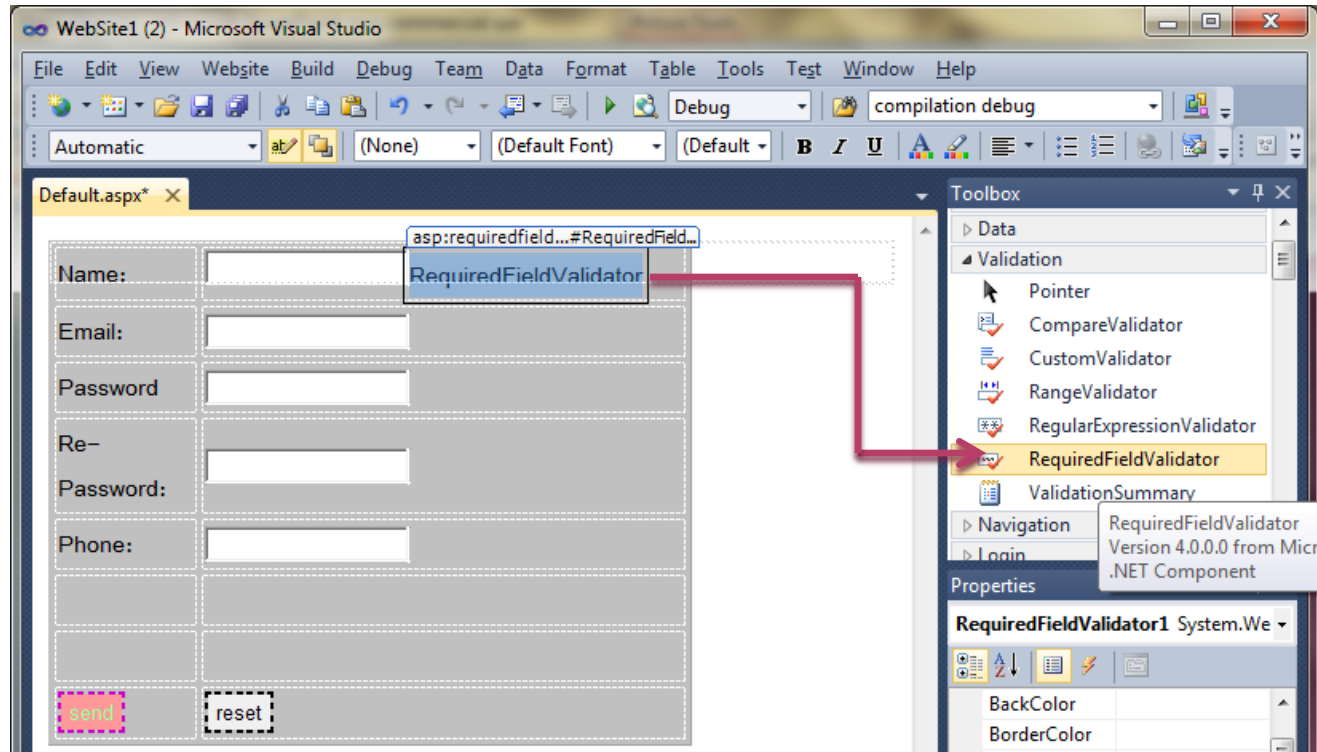
The status bar at the bottom shows the current code: `<asp:TextBox#Email>`.

TOOLBAR > VALIDATION



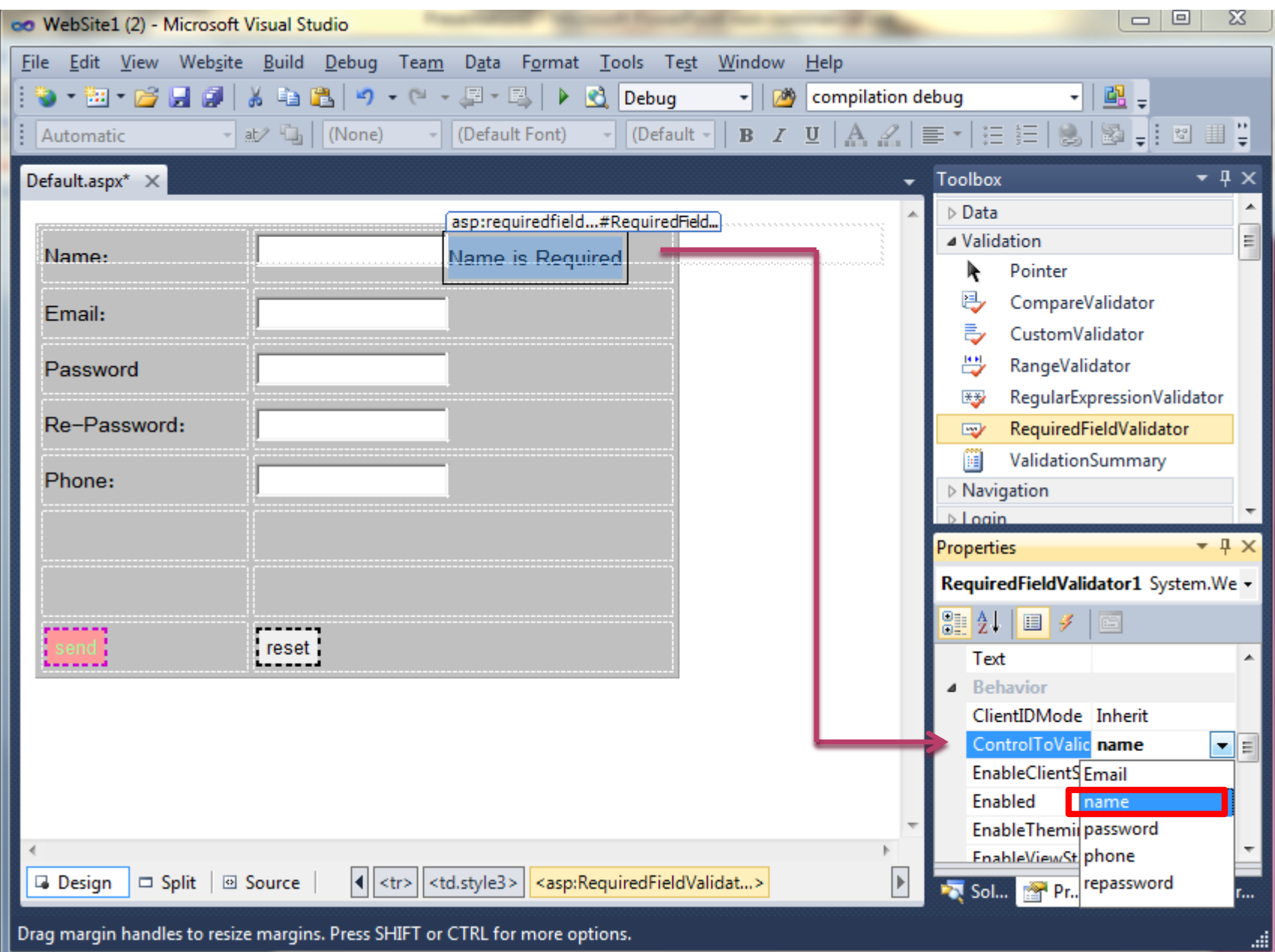
STEPS

- Drag the appropriate (wanted) validation control next to the field want to validate.

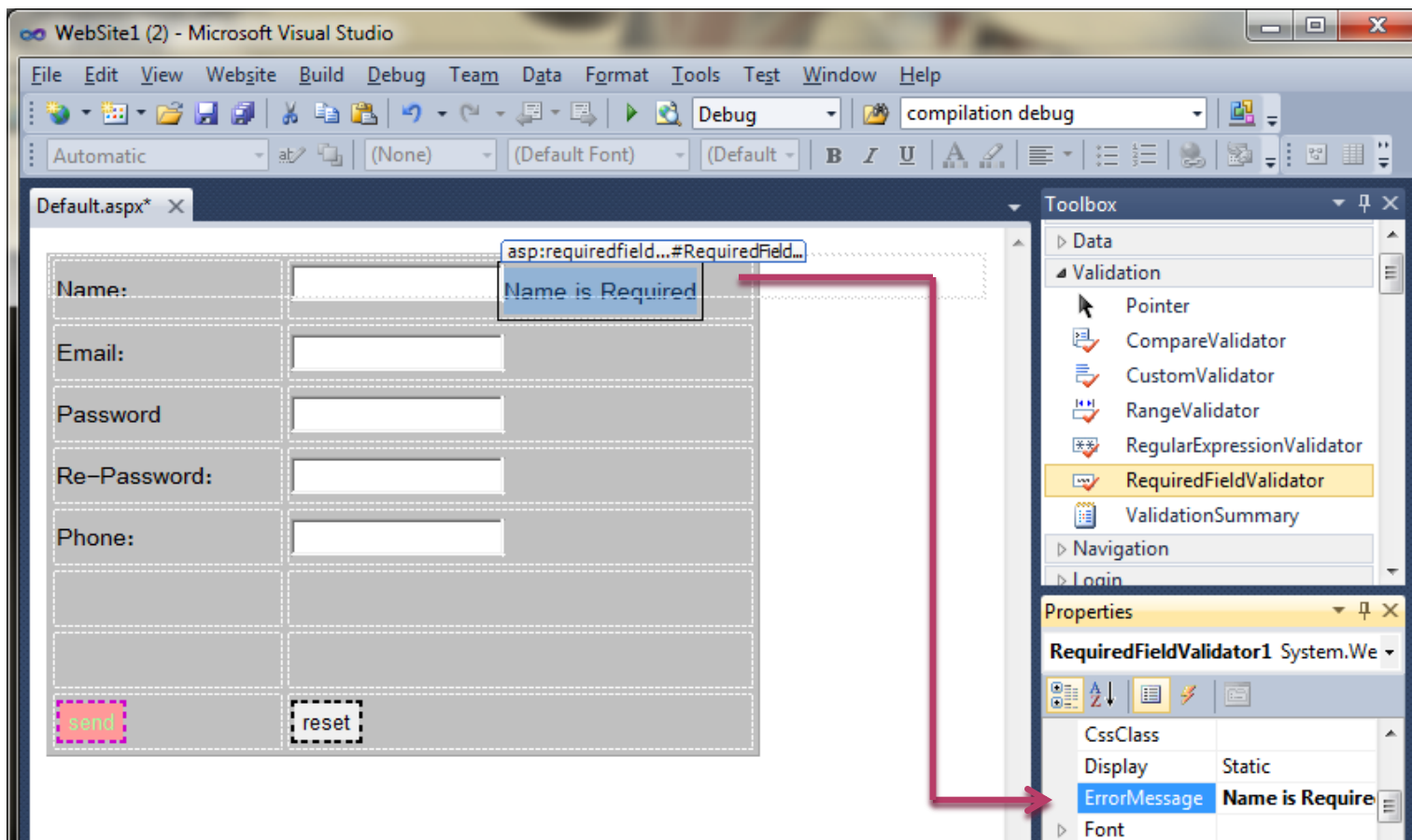


STEPS

- Check the validation Control to edit its properties from Properties window.
- It is **VERY IMPORTANT** to change **ControlToValidate** property to the ID of the field wanted to validate (ex: Name)
- You can change the Error Message that will be displayed to the user by editing ErrorMessage Property **** The ErrorMessage will be displayed in the ValidationSummary While TEXT property displays the message next to the field being tested or validated.**



- You can change the Error Message to be displayed to the user by editing ErrorMessage Property



TESTING:

- Test the validation by running the code.



The screenshot shows a web browser window with the address bar displaying `http://localhost:59939/WebSite1/Default.aspx`. The browser tab is labeled "localhost". The main content area contains a registration form with the following fields and labels:

- Name: R
- Email:
- Password:
- Re-Password:
- Phone:

At the bottom of the form, there are two buttons: "send" and "reset". A red arrow points to the "R" label next to the Name input field.

HINT.

- You can do the same for the rest of the Validation Controls. BUT; be careful to change the appropriate and required properties for each **Validation Control**.

PROPERTIES REQUIRED TO DO A SIMPLE VALIDATION.

<i>Type of validation</i>	<i>Control to use</i>	<i>Properties Needed to change</i>
Required entry	RequiredFieldValidator	ControlToValidate, Text,
Comparison to a value	CompareValidator	ControlToValidate, Text, ControlToCompare
Range checking	RangeValidator	ControlToValidate, Text, MaximumValue, MinimumValue
Pattern matching	RegularExpressionValidator	ControlToValidate, Text, ValidationExpression
User-defined	CustomValidator	ControlToValidate, Text. ** AS NEEDED**

DISPLAY PROPERTY

Display behavior	Description
None	The validation message is never displayed inline.
Static	Space for the validation message is allocated in the page layout.
Dynamic	Space for the validation message is dynamically added to the page if validation fails.

REFERENCE:

- [http://msdn.microsoft.com/en-us/library/debza5t0\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/debza5t0(v=vs.100).aspx)
- [http://msdn.microsoft.com/en-us/library/bwd43d0x\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/bwd43d0x(v=vs.100).aspx)