

This Free E-Book is brought to you by [Natural-Aging.com](http://Natural-Aging.com).



## Understanding ASP.NET Validation Controls

By Balaji

### Understanding ASP.NET Validation Controls by Balaji

Understanding ASP.NET Validation Controls

After you create a web form, you should make sure that mandatory fields of the form elements such as login name and password are not left blank; data inserted is correct and is within the specified range. Validation is the method of scrutinizing that the user has entered the correct values in input fields. In HTML you can perform validation either by checking the values at client-side or after submitting the form at the server-side. But these methods in HTML take lots of time to create and maintain the code. Moreover, if the user has disabled JavaScript then he or she may not receive the message regarding the error.

Therefore, in ASP.NET you can use ASP.NET Validation Controls while creating the form and specify what ASP.NET Validation Controls you want to use and to which server control you want bind this. ASP.NET Validation Controls are derived from a common base class and share a common set of properties and methods. You just have to drag and drop the ASP.NET Validation Control in the web form and write one line of code to describe its functionality.

This reduces the developer time from writing JavaScript for each type of validation. Moreover, through ASP.NET Validation Controls if any invalid data is entered the browser itself detects the error on the client side and displays the error without requesting the server. This is another advantage because it reduces the server load.

The five types of ASP.NET Validation Controls in ASP.NET Framework include:

- RequiredField Validation control: prompts message if any input field is left blank. This validation control can also be used to prompt message to the user if he or she has left any input field with its default value.
- Range Validation control: Prompts message to the user, if the data entered in the input field is not within the range of the values specified by the Maximum and Minimum properties of the validation

control.

- **Comparison Validation control:** Allows the user to compare two values and check for comparisons such as equality, greater-than, less-than etc. In addition you can check whether the data entered in the input field is of the data type as specified by you.
- **RegularExpression Validation control:** Allows you to validate if the format of a certain input field is correct or not. You can check the validation of the commonly performed formats such as social security numbers, e-mail addresses, telephone numbers, and postal code.
- **Custom Validation control:** Allows you to define your own condition for validating the data in the input fields. Two validation functions can be performed using Custom Validation control: first on the server-side and second on the client-side. These functions contain logic defined by you to validate the input fields. These functions returns the True value and False value, if the condition you specified is correct or not respectively.

If the control's value does not validate to what you have specified in the validation control tag, the web form displays an error message. Moreover, you can customize the format of the error message by using the control properties such as BackColor, BorderStyle, and BorderWidth. In addition, you can use Cascading Style Sheet (CSS) to format an error message.

To access online version of the above article, please go to <http://www.dotnet-guide.com/validation.html>

Visit <http://www.dotnet-guide.com> for a complete introduction to .NET framework. Learn about ASP.NET, VB.NET, C# and other related technologies.

## **Understanding Custom Server Controls in ASP.NET**

**By Balaji**

### **Understanding Custom Server Controls in ASP.NET by Balaji**

Understanding Custom Server Controls in ASP.NET

ASP.NET offers many server controls for the developers to create web applications. However, at some point of development time when the developer does not get the control they want, they can create a new server control. This new server control can be called ASP.NET Custom Server Controls or user control. The basic difference between a ASP.NET Custom Server Controls and a user control is that unlike a user control that does not appear in the Toolbox, you can view a ASP.NET Custom Server Controls in the Toolbox. ASP.NET Custom Server Controls have their own events such as Enter, Onclick, and Onmouseover.

ASP.NET Custom Server Controls are very handy tool to use in your web forms. User controls prove

inefficient in advanced scenarios. ASP.NET Custom Server Controls are the compiled code that makes them user friendly. You should write the ASP.NET Custom Server Controls in code, thereby view it in the visual designer with full properties and design-time features. You can create a ASP.NET Custom Server Controls by inheriting one of the base control classes. These classes have all the functionality that is needed for a server control. Therefore, you just have to modify some of the programming aspects to suit your requirements.

Just as you have created a ASP.NET Custom Server Controls you can customize it too to create a unique identity. You can even consume a single ASP.NET Custom Server Controls for multiple web forms. For ASP.NET Custom Server Controls only one copy of the control is needed in the Global Assembly Cache (GAC). Moreover, ASP.NET Custom Server Controls are very helpful if you want a dynamic layout for your web application.

To access online version of the above article, go to <http://www.dotnet-guide.com/customcontrols.html>

Visit <http://www.dotnet-guide.com> for a complete introduction to .NET framework. Learn about ASP.NET, VB.NET, C# and other related technologies.



**This Free E-Book has been brought to you by [Natural-Aging.com](http://Natural-Aging.com).**



**[100% Effective Natural Hormone Treatment](#)**

**Menopause, Andropause And Other Hormone Imbalances  
Impair Healthy Healing In People Over The Age Of 30!**