

# Using TextBoxWatermark With Validation Controls

Christian Wenz

## Overview

The **TextBoxWatermark** control in the AJAX Control Toolkit extends a text box so that a text is displayed within the box. When a user clicks into the box, it is emptied. If the user leaves the box without entering text, the prefilled text reappears. This may collide with ASP.NET Validation Controls on the same page, but these issues may be overcome.

## Steps

The basic setup of the sample is the following: a **TextBox** control is watermarked using a **TextBoxWatermarkExtender** control. A button triggers a postback and will later be used to trigger the validation controls on the page. Also, a **ScriptManager** control is required to initialize ASP.NET AJAX:

```
<form id="form1" runat="server">
  <asp:ScriptManager ID="asm" runat="server" />
  <div>
    <asp:TextBox ID="Name" runat="server" />
    <br /><asp:Button ID="btn" runat="server" Text="Submit"
    OnClick="btn_Click" /><br />
    <asp:Label ID="lbl" runat="server" />
  </div>
  <ajaxToolkit:TextBoxWatermarkExtender ID="tbwe" runat="server"
    TargetControlID="Name" WatermarkText="&lt;Your Name&gt;" />
</form>
```

Now add a **RequiredFieldValidator** control that checks whether there is text in the field when the form is submitted. The **InitialValue** property of the validator must be set to the same value that is used in the **TextBoxWatermarkExtender** control: When the form is submitted, the value of an unchanged textbox is the watermark value within it:

```
<asp:RequiredFieldValidator ID="rfv1" ControlToValidate="Name"
  Text="*" InitialValue="&lt;Your Name&gt;" Display="Dynamic"
  runat="server" />
```

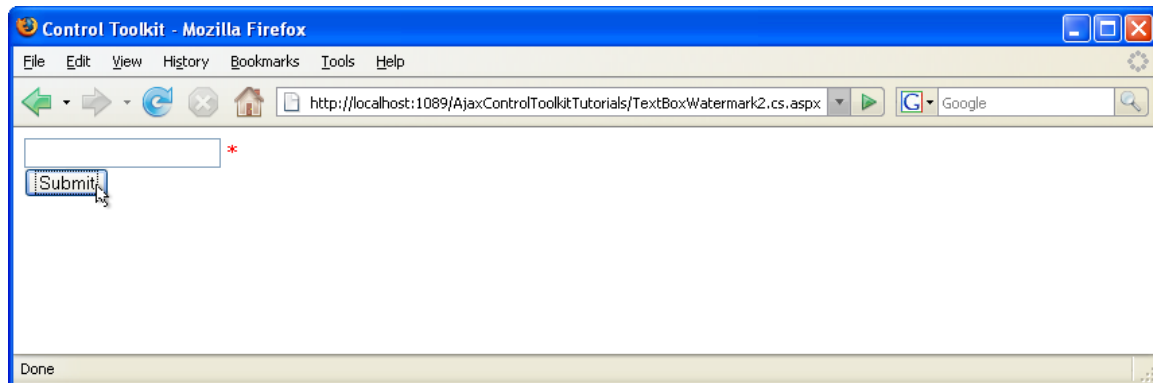
However there is one problem with this approach: If the client disables JavaScript, the text field is not prefilled with the watermark text, therefore the **RequiredFieldValidator** does not trigger an error message. Therefore, a second **RequiredFieldValidator** control is required which checks for an empty text box (omitting the **InitialValue** attribute).

```
<asp:RequiredFieldValidator ID="rfv2" ControlToValidate="Name"
  Text="*" Display="Dynamic" runat="server" />
```

Since both validators use **Display="Dynamic"**, the end user cannot distinguish from the visual appearance which of the two validators was fired; instead, it looks like there was only one of them.

Finally, add some server-side code to output the text in the field if no validator issued an error message:

```
<script runat="server">
  Protected Sub btn_Click(ByVal sender As Object, ByVal e As
    System.EventArgs)
    lbl.Text = HttpUtility.HtmlEncode(Name.Text)
  End Sub
</script>
```



**The validator complains that there is no text in the field**