

# Creating a Rating Control

Christian Wenz

## Overview

Many websites, from e-commerce to community sites, offer their users to rate articles or items. This usually requires some coding effort, but we do have the Control Toolkit to our disposal.

## Steps

First of all, you need (at least) two kinds of images: one for a filled out rating item, and one for an empty rating item. A "rating item" is usually a star or a smiley. For this scenario, you find three files, smiley.png and empty.png and smiley-done.png as part of the source code downloads for this tutorial.

Then, create a new ASP.NET file and start with adding a **ScriptManager** control to it:

```
<asp:ScriptManager ID="asm" runat="server" />
```

Then, add the **Rating** control from the ASP.NET AJAX Control Toolkit. The following attributes need to be set for this example:

- **CurrentRating**—the initial rating to be used
- **MaxRating**—the maximum rating
- **EmptyStarCssClass**—the CSS class to use when a rating item ("star") is empty
- **FilledStarCssClass**—the CSS class to use when a rating item ("star") is filled out
- **StarCssClass**—the CSS class to use for a visible star
- **WaitingStarCssClass**—the CSS class to use while a star rating is sent back to the server

And here is the markup which creates a rating control with five items (smileys) of which none is filled out initially:

```
<ajaxToolkit:Rating ID="r1" runat="server"  
  CurrentRating="0" MaxRating="5"  
  EmptyStarCssClass="emptypng" FilledStarCssClass="smileypng"  
  StarCssClass="smileypng" WaitingStarCssClass="donesmileypng" />
```

The three referenced CSS classes now need to show the appropriate image files, which is easy to do using CSS:

```
<style type="text/css">  
  .emptypng { background-image: url(empty.png); width: 32px;  
    height: 32px; }  
  .smileypng { background-image: url(smiley.png); width: 32px;  
    height: 32px; }  
  .donesmileypng { background-image: url(smiley-done.png);  
    width: 32px; height: 32px; }  
</style>
```

Make sure that you provide the width and height of the three images, otherwise the display may look a bit messed up.

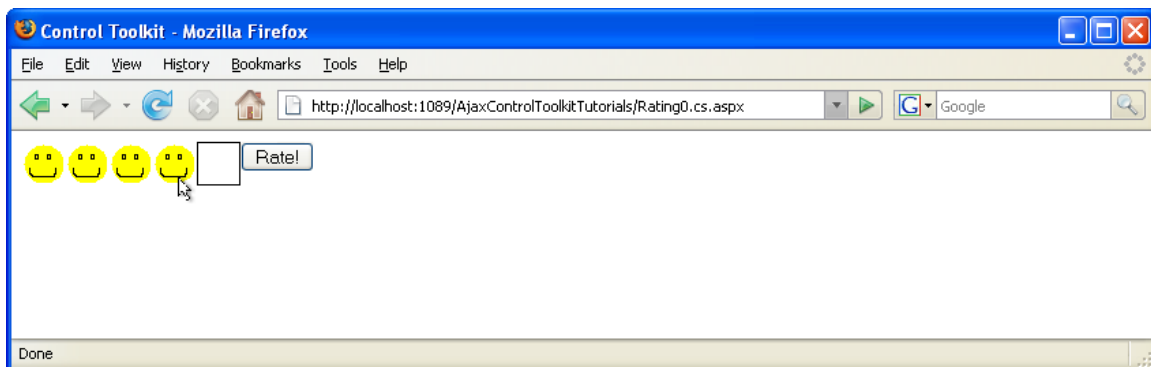
Finally, the result of the rating should be displayed to the user (or, at least saved in a database). So add a label for the output of a text message and a submit button to post back the rating form to the server:

```
<asp:Label ID="Label1" runat="server" />
<input type="submit" id="Submit1" runat="server" value="Rate!" />
```

In the server-side code, access the Rating control via its **ID** and then access its **CurrentRating** property which is the number of the selected rating items, in our example a value between 0 and 5.

```
<script runat="server">
  Sub Page_Load()
    If Page.IsPostBack Then
      Label1.Text = "Your rating: " & r1.CurrentRating
    End If
  End Sub
</script>
```

Save the page and load it into your browser. When you hover over the (initially empty) rating items, a JavaScript effect occurs: The rating changes. When you click on the set of stars, the current rating is retained. Finally, when you submit the form, the server-side code outputs the selected rating.



## Creating a rating system with minimal code