

ASP.NET Image Gallery, Uploader & Thumbnailer – Part 1

Working with images is a requirement of many applications, from business software to simple personal web pages. There are hundreds of different implementations of this feature, some good, others not so good. When using images in ASP.NET applications you have the added benefit of making things a little easier on yourself by using a more object oriented approach so that you can create components that you can reuse in other applications.

Depending on the requirements of your images module it may be necessary to look towards third party components. There is literally hundreds of ASP.NET third party controls which deal with imaging, image uploading and image manipulation. Before you implement a costly third party component it's important to understand what you can and cannot do with the existing .NET framework.

The purpose of this article is to show 3 basic imaging techniques you can implement in your applications which all make use of the existing .NET framework classes. These three techniques are: uploading, thumbnailing, and dynamic display (gallery).

For this article we'll build a simple demonstration ASP.NET application where you can upload an image into a specific folder on the web server, have that image automatically thumbnailled on the fly, and then have the image thumbnails rendered into an HTML dynamically. These are the basic building blocks of most applications utilizing images.

Note: Visual Studio.NET is required for this tutorial. Both VB.NET and C# code examples will be provided

Preparing the Application

Begin by creating a new ASP.NET application named "ImgThumbnail". Rename the default WebForm1.aspx to UploadImage.aspx for a more meaningful name. If you're using C# you'll also need to manually add a default StyleSheet to the project named "Styles.css".

Our sample application will store images into 2 subfolders: People, and Cars. These folders are hardcoded for the purpose of demonstration, and could map to nearly anything in your database or web server. The important thing to note is that these folders should not normally be hardcoded.

First add a new folder to the project named "Images". Then add two subfolders to the Images folder: One named "People" and the other named "Cars".

Next let's replace the default contents of the Styles.css stylesheet file with our own look and feel CSS rules:

```
BODY
{
    BACKGROUND-COLOR: white;
    FONT-FAMILY: Verdana, Helvetica, sans-serif;
    FONT-SIZE: 10px;
    FONT-WEIGHT: normal;
}

.DROPDOWN
{
    FONT-FAMILY: Verdana, Helvetica, sans-serif;
    FONT-SIZE: 10px;
```

Thanks for your interest in the "ASP.NET Image Gallery, Uploader & Thumbnailer – Part 1" article by Kevin Koch. To purchase the full article [click here](#)