

# File Upload with Deletion of Files during Edit or Delete

## Displaying your image:

### 1. Details View

- Replace the `@Html.DisplayFor()` with an `img` tag
- Make sure the `src` location is the place where you are storing your files.

```
<dd>
    @*@Html.DisplayFor(model => model.Logo)*@
    
</dd>
```

### 2. Create View

- Before doing any work in the view, go to a search engine and locate an image to be used if the user doesn't provide one with the record. This will be our default image for any record without an image. This image should be saved where the rest of your images will go (per previous step).
- In the `BeginForm()`, update it to reflect The **CREATE action**, **ControllerName**, **FormMethod** of `Post`, and an **Encoding Type** of `multipart/form-data`

```
@using (Html.BeginForm("Create", "Recipients", FormMethod.Post,
    new { enctype="multipart/form-data" }))
{
    @Html.AntiForgeryToken()
```

- Replace the `EditorFor()` and `ValidationMessageFor()` with some prompt text and an **input type=file** as shown below. The **name attribute** here **MUST MATCH** the **value passed into the Post Action** later in these steps. Spelling matters, casing does not.

No Images Taller than 350px: `<input type="file" style="display:inline" name="logo" />`

### 3. EDIT View:

- In the `BeginForm()`, update it to reflect The **EDIT action**, **ControllerName**, **FormMethod** of `Post`, and an **Encoding Type** of `multipart/form-data`
- Below the `HiddenFor()` for the ID, **add a `HiddenFor()`** for the property of the object that represents your **image**.

```
@using (Html.BeginForm("Edit", "Recipients", FormMethod.Post,
    new { @enctype="multipart/form-data" }))
{
    @Html.AntiForgeryToken()

    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
    @Html.HiddenFor(model => model.ID)
    @Html.HiddenFor(model => model.Logo)
```

- Replace the `EditorFor()` and `ValidationMessageFor()` with some prompt text and an **input type=file** as shown below. The **name attribute** here **MUST MATCH** the **value passed into the Post Action** later in these steps. Spelling matters, casing does not. (optionally, you can display the current image as shown below).

Existing Image: `<br /><br />`  
Change Image (No Images Taller than 350px): `<input type="file" style="display:inline" name="logo" />`

#### 4. Create Post

- Add the parameter for HttpPostedFileBase. This MUST match the name attribute of the input type=file in the view
- In the if(ModelState.IsValid) – code the region for File upload – replace noImage.png with your default image

```
// POST: Recipients/Create
// To protect from overposting attacks, please enable the specific
// properties you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "ID,Name,Description,URL,Logo")]
    Charity charity,[ HttpPostedFileBase logo)
{
    if (ModelState.IsValid)
    {
        #region File Upload
        //use a default image if none is provided
        string imgName = "noImage.png";
        if (logo != null)//Your HttpPostedFileBase Object that should be
            //added to the action != null
        {
            //get image and assign to variable
            imgName = logo.FileName;

            //declare and assign ext value
            string ext = imgName.Substring(imgName.LastIndexOf('.'));

            //declare a list of valid extensions
            string[] goodExts = { ".jpeg", ".jpg", ".gif", ".png" };

            //check the ext variable (toLowerCase()) against the valid list
            if (goodExts.Contains(ext.ToLower()) && (logo.ContentLength <= 4194304))//4194304 - 4MB max by ASP.NET
            {
                //if it is in the list rename using a guid
                imgName = Guid.NewGuid() + ext;

                //save to the webserver
                logo.SaveAs(Server.MapPath("~/Content/Images/" + imgName));
            }
            else
            {
                imgName = "noImage.png";
            }
        }

        //no matter what add the imageName to the object
        charity.Logo = imgName;
        #endregion

        db.Charities.Add(charity);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(charity);
}
```

## 5. Edit Post

- Add the parameter for HttpPostedFileBase. This MUST match the name attribute of the input type=file in the view
- In the if(ModelState.IsValid) – code the region for File upload

```
// POST: Recipients/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "ID,Name,Description,URL,Logo")] Charity charity,
    HttpPostedFileBase logo)
{
    if (ModelState.IsValid)
    {
        #region File Upload
        if (logo != null) //Your HttpPostedFileBase Object that should be added to the action != null
        {
            //get image and assign to variable
            string imgName = logo.FileName;

            //declare and assign ext value
            string ext = imgName.Substring(imgName.LastIndexOf('.'));

            //declare a list of valid extensions
            string[] goodExts = { ".jpeg", ".jpg", ".gif", ".png" };

            //check the ext variable (toLowerCase()) against the valid list
            if (goodExts.Contains(ext.ToLower()) && (logo.ContentLength <= 4194304)) //4194304 - 4MB max by ASP.NET
            {
                //if it is in the list rename using a guid
                imgName = Guid.NewGuid() + ext;

                //save to the webserver
                logo.SaveAs(Server.MapPath("~/Content/Images/" + imgName));

                //Make sure you are not deleting your default image
                if (charity.Logo != null && charity.Logo != "noImage.png")
                {
                    //remove the original file (ViewBag.Hidden) from the edit view
                    System.IO.File.Delete(Server.MapPath("~/Content/Images/" + Session["currentImage"].ToString()));
                }

                //only save if the image meets criteria imageName to the object
                charity.Logo = imgName;
            }
        }

        //If the image is bad (not in our list OR No file was included) the
        //HiddenFor() in the view will care for retaining the value.

        #endregion

        db.Entry(charity).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(charity);
}
```

## 6. Delete Post

- a. Add the code for removing the existing image with the current record.

```
// POST: Recipients/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Charity charity = db.Charities.Find(id);

    if (charity.Logo != null && charity.Logo != "noImage.png")
    {
        //remove the original file from the edit view
        System.IO.File.Delete(Server.MapPath("~/Content/Images/" + Session["currentImage"].ToString()));
    }

    db.Charities.Remove(charity);
    db.SaveChanges();
    return RedirectToAction("Index");
}
```