Taking it to the Limit with Pure ASP upload 3.0: Image Update Options

DMXzone Pure ASP Upload add-on pack will allow you to delete the image on update or when you delete a record. However, there's no way to completely remove the image from an existing record once one has been uploaded. Due to the way Dreamweaver and the extension layer works, the only option is to replace it with another image. In addition to all the functionality Pure ASP Upload and Smart Image Processor offered I also wanted my CMS to have options to remove an image from the page/product/article whatever the case may be without replacing it. And with product catalogues and applications where there are child records which have images you will want to set up the deletes of the child records to prevent slowing your application with widows and orphans. And you'll want to delete images associated with child records as well.

In this tutorial we'll first look at how to delete one image and set the database field null by toggling the skip empty fields feature of Pure ASP Upload 3. Then, we will learn to delete an image and its thumb while also setting the database field to null.

If you implement the methods presented in this tutorial, not only will you offer your users more options, your applications will run faster and consume fewer resources.

Assumptions

This tutorial assumes you know how to create a record set to populate your form with existing data. If you have never configured a connection string, created a record set, populated a form with data, applied an insert or update behaviour and successfully run the page then this tutorial is not for you. Of course you also need to know how to apply the Pure ASP Upload behaviour. The first thing we do is set up our form. So let's get started!

Form Modifications

Since we're dealing with images it seems a good idea to display the existing image on the update page so the user can make a decision as to whether or not to replace or remove the image. To do this we will create a conditional region. The conditional region will display two elements: If the database image field has a value other than null, the first will be image itself. The second element is the **Delete Image?** checkbox. The conditional region will do two things. If there is an image, it will be displayed along with the Delete Image? checkbox. If there is no image, the message **No existing image** will be displayed. If the record set image field has an image specified and no image is found, then the dreaded red X will appear. On the front of end of the application, I prefer to use the **file scripting object** to determine if the specified image exists, before displaying it, thus avoiding the red X when the image cannot be found. You will be learning to use the file scripting object into today's tutorial.

Tip: The File Scripting Object--often abbreviated so--allows us to access the server file system.

Insert two new rows in your form. And the first cell of the first row type Existing Image. In the second row open ASP script block and type **If** to open the conditional region and from the bindings panel drag the record set image field onto the page after the opening **If**. Then enter the not equal symbol empty quote marks and the word Then. Now close to script block.

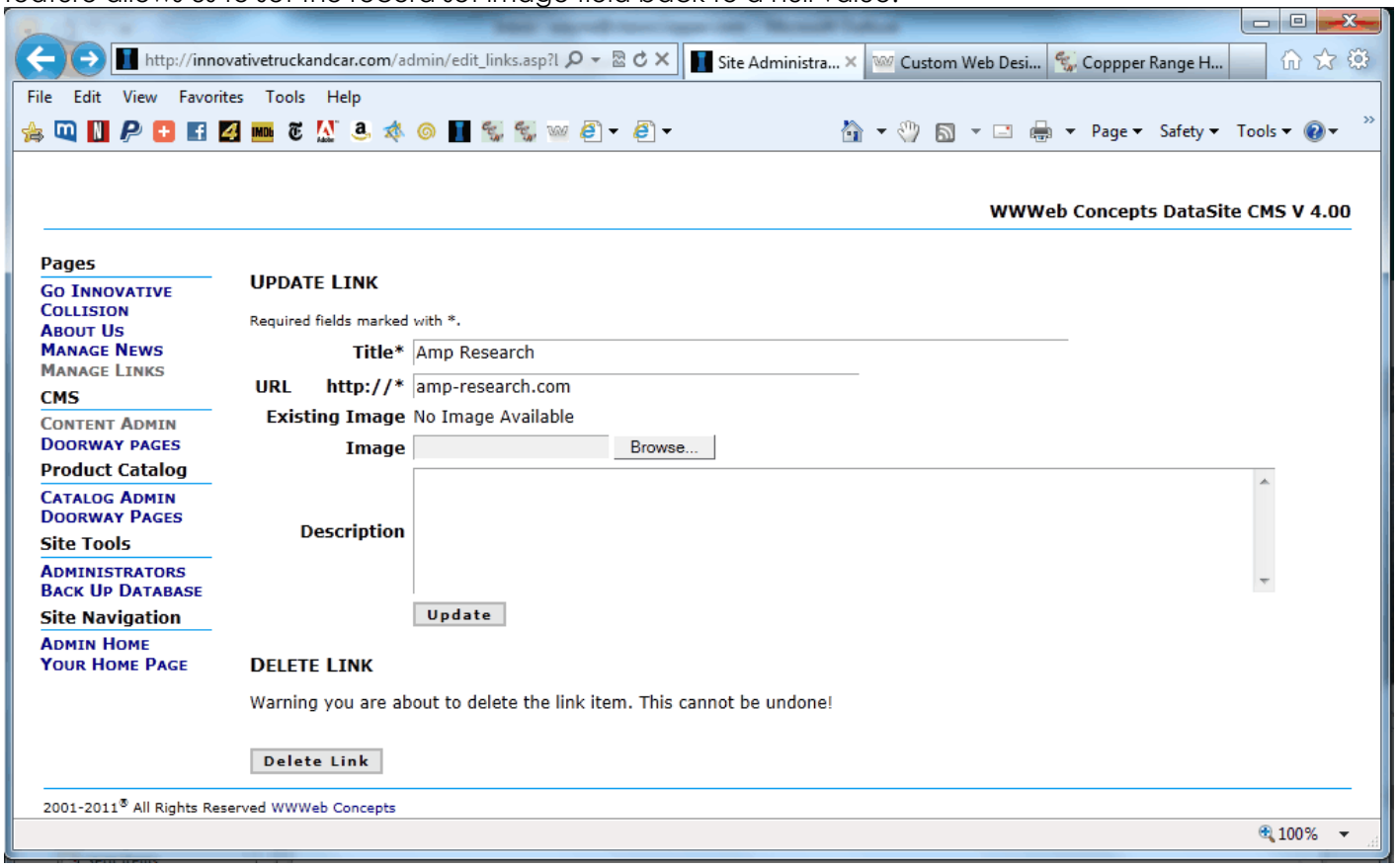
Copyright © 2011 DMXzone.com All Rights Reserved
To get more go to DMXzone.com

In the second cell, create an image tag to display your image. In the first cell of the second row, type **Delete Image?** In the second cell, insert a form checkbox, name it **Delete_Image**, set checked value is **Y**. Enter a carriage return, open an ASP script block. Type else and close the ASP script block. Type **No Image Available**. Open an ASP script block type **End If** to close the conditional region.

```
<tr>
  <td colspan="2"> </td>
  <td>
   <% If (rs_links.Fields.Item("Link_Image").Value)<> "" Then%>
   <img src="../site_images/<%=(rs_links.Fields.Item("Link_Image").Value)%>"
alt="Existing Image" /></td>
</tr>
<tr>
  <td colspan="2">delete Image?</td>
  <td><input name="Delete_Image" type="checkbox" id="Delete_Image" value="Y" />
   <% Else%>No Image Available
   <% End If %>
  </td>
</tr>
```

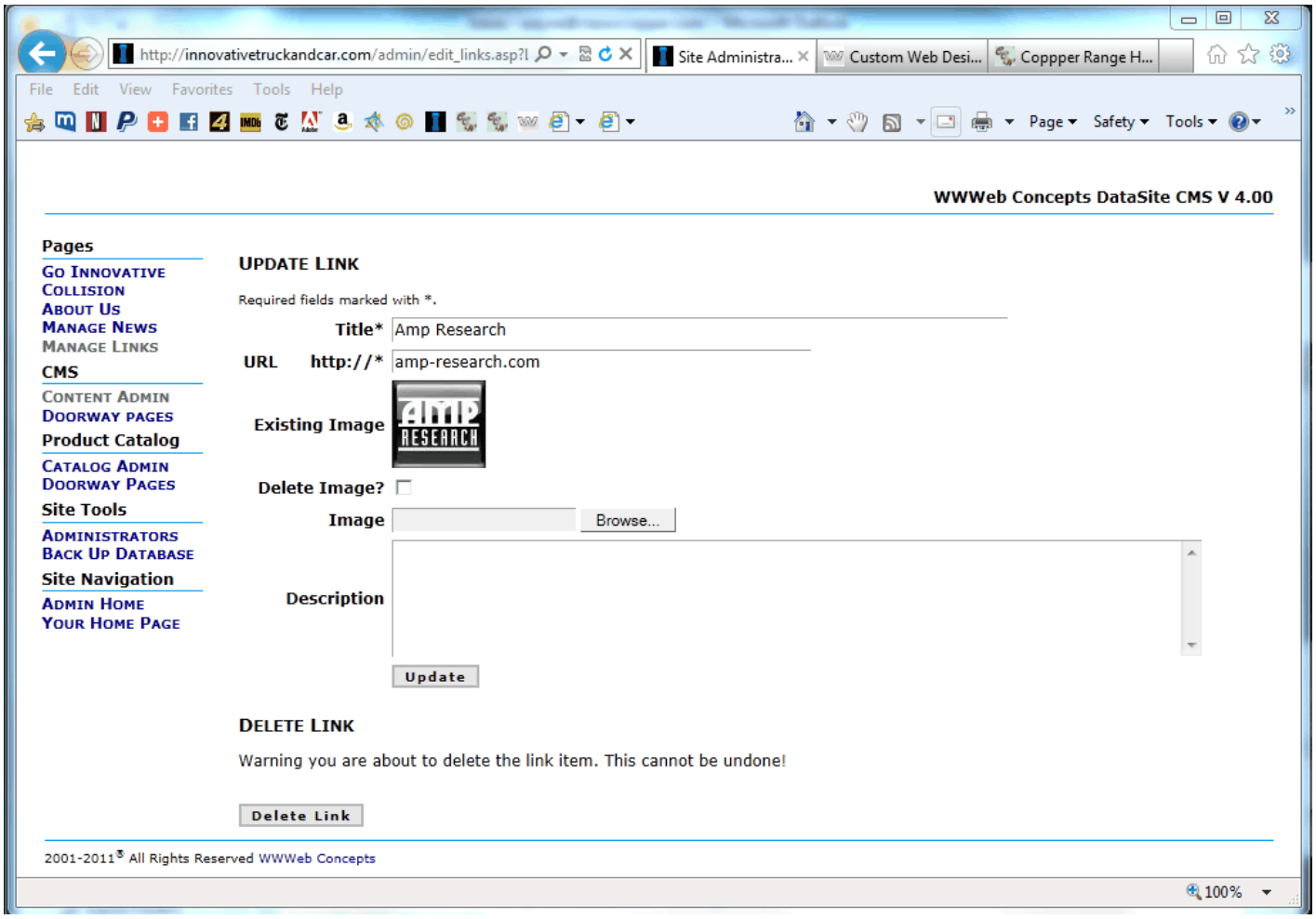
Form code block

The **Delete Image?** checkbox does two things if checked. It turns off the skip empty fields feature of Pure ASP Upload and it turns on the code that deletes the image file from the server. Turning off the skip empty fields feature allows us to set the record set image field back to a null value.



Page with no image. 1

Copyright © 2011 DMXzone.com All Rights Reserved
To get more go to DMXzone.com



Page with image 1

Toggling the Skip Empty Fields Feature

Before we can toggle the empty fields feature, we need to apply the Dreamweaver update server behavior and Pure ASP Upload. Let's go ahead and do that now. But don't save the page yet.

Switch to code view and look at the Dreamweaver update server behavior code you'll notice that it's using the **generic Request()** Collection and the **Request.Form()** collection. Now save the page and as you do watch the code changes. You just saw the Pure ASP Upload modify the Dreamweaver server update server behavior for uploads. You'll notice that the generic Request() and Request.Form collections have been replaced with **UploadFormRequest()**. What you've just seen usually goes on behind-the-scenes. If you were paying close attention you may have noticed that the image field parameter code has also changed.

Tip: The generic Request() and the Request.Form() collections cannot be called after calling a binary upload.

The DMXzone Pure ASP upload extension has implemented the **MM_IIF** for the skip empty fields feature. The MM_IIF enough returns one of two values true or false. A false value means the requested field is null or empty. A true value indicates that the requested field is not empty. If the MM_IIF returns false, the hidden field **upload_org_image_field** created by the Pure ASP Upload extension which contains the original image field value is substituted for the upload form field, thus skipping the empty field. This is the parameter we're going to use to toggle the skip empty fields feature. We're going to do this by creating two parameters: one

Copyright © 2011 DMXzone.com All Rights Reserved
To get more go to DMXzone.com

which skips the empty field using the MM_IIF and one that does not skip the empty field. Then we are placing them in an **if… then… else** statement and using the Delete Image? checkbox to toggle the feature and set the database image field value to null or skip the empty field when we wish.

Tip: You'll find the MM_IIF implementation in the code block immediately above the Dreamweaver update server behaviour.

Now that we know what we are going to do, let's get to it. The modified Dreamweaver update server behavior appears in the code block below. In this case the name of my image field is named Link_Image. You will notice I have added a blank space and a comment before and after our If… Then… Else… statement. Notice I've used the UploadFormRequest collection created by Pure ASP Upload to request the value of the Delete Image? checkbox. If the value is anything other than null, the code to reset the image field value to null is activated. Notice that if the user checks the Delete Image? checkbox and uploads a new image, the page still works properly and the image name value is updated in the database.

```
<%
If (CStr(UploadFormRequest("MM_update")) = "UpdateForm") Then
  If (Not MM_abortEdit) Then
    ' execute the update
    Dim MM_editCmd

    Set MM_editCmd = Server.CreateObject ("ADODB.Command")
    MM_editCmd.ActiveConnection = MM_datasource_STRING
    MM_editCmd.CommandText = "UPDATE Links SET Link_Title = ?, Link_URL = ?, Link_Image
= ?, Link_Description = ? WHERE Link_ID = ?"
    MM_editCmd.Prepared = true
    MM_editCmd.Parameters.Append MM_editCmd.CreateParameter("param1", 203, 1,
536870910, UploadFormRequest("Link_Title")) ' adLongVarWChar
    MM_editCmd.Parameters.Append MM_editCmd.CreateParameter("param2", 203, 1,
536870910, UploadFormRequest("Link_URL")) ' adLongVarWChar

   'Toggle skip empty field behavior to remove existing image
   If CStr(UploadFormRequest("Delete_Image")) <> "" Then
      MM_editCmd.Parameters.Append MM_editCmd.CreateParameter("param3", 203, 1,
536870910, UploadFormRequest("Link_Image")) ' adLongVarWChar
   Else
      MM_editCmd.Parameters.Append MM_editCmd.CreateParameter("param3", 203, 1,
536870910, MM_IIF(UploadFormRequest("Link_Image"), UploadFormRequest("Link_Image"),
UploadFormRequest("upload_org_Link_Image"))) ' adLongVarWChar
   End If
  'End toggle

      MM_editCmd.Parameters.Append MM_editCmd.CreateParameter("param4", 203, 1,
536870910, UploadFormRequest("Link_Description")) ' adLongVarWChar
      MM_editCmd.Parameters.Append MM_editCmd.CreateParameter("param5", 5, 1, -1,
MM_IIF(UploadFormRequest("MM_recordId"), UploadFormRequest("MM_recordId"), null)) '
adDouble
```

The line of code after the **Then** uses the Dreamweaver generated parameter code modified for upload. It does not use the MM_IIF implementation and does not skip the empty field. The line of code after the **Else** uses the parameter as modified by the pure ASP upload extension with the MM implementation and skips a beat fields, instead using the value from the hidden field upload_org_Link_Image, which as you recall holds the value of the original image. Now all we have to do is close the If… Then… Else I statement with the final **End If**.

Copyright © 2011 DMXzone.com All Rights Reserved
To get more go to DMXzone.com

Tip: The parameters both have the same name: param3 in the example. This is because only one parameter is used in the SQL statement depending upon the condition of the Delete Image? checkbox.

Now that we have successfully toggled the skip empty fields feature. It's time to set up the image delete.

Deleting the Image

Before we can delete the image we check two conditions: Is the user removing the existing image or is the user uploading a new image. In either case, we delete the existing image. But before we delete the image we need to create our file scripting object to determine if the image exists on the server's file system.

First we get the two functions from Marcellino Bommezijn's great tutorial on deleting images when updating or deleting a record. The first function, **newFileSystemObject()**, creates the file scripting object, and the second function **fileExists** uses the newly created file scripting object to determine if the file exists. I usually place the functions in an include file with other miscellaneous functions. However you can also place them in the page, provided it's before you call them. Immediately above the MM_IIF Implementation is a good place.

```
<%
Function newFileSystemObject()
set newFileSystemObject=Server.CreateObject("Scripting.FileSystemObject")
End Function

Function fileExists(aFileSpec)
fileExists=newFileSystemObject.fileExists(aFileSpec)
End Function
%>
```

Now let's take a look at the delete code. We use the UploadFormRequest collection to ask for two values: the delete image checkbox value and image upload field value. If either has a value other than null then the image delete code executes.

Next we create a new file scripting object to actually delete the image, set the folder path, and the image file name. We'll get the file name from the hidden form field, upload_org_image_field, created by Pure ASP Upload which holds the original image value. We then call the fileExists function to make sure we have the file to delete. If we try to execute a delete where there's no file the page will throw an error. If the file exists we use the **File.DeleteFile()** method to remove the image from the server's file system. We close our If… Then statement and set the file back to nothing.

```
    ' Delete the file before we update the record
   If CStr(UploadFormRequest("Delete_Image")) <> "" Or
CStr(UploadFormRequest("Link_Image") <> "") Then
      'create file scripting object
      Set File = CreateObject("Scripting.FileSystemObject")
      ImageFolder = Server.MapPath("..\site_images\")
      ImagePath = Imagefolder & "\" & (UploadFormRequest("upload_org_Link_Image"))
      'if file exists delete the file
      If fileExists(ImagePath) Then
            File.DeleteFile(ImagePath)
      End If
      Set File = Nothing
      End If 'end delete file
```

Copyright © 2011 DMXzone.com All Rights Reserved
To get more go to DMXzone.com

Finally the Dreamweaver generated update behavior appends the query string to redirect to the desired page after the update.

```
    MM_editCmd.Execute
    MM_editCmd.ActiveConnection.Close

    ' append the query string to the redirect URL
    Dim MM_editRedirectUrl
    MM_editRedirectUrl = "add_links.asp"
    If (UploadQueryString <> "") Then
      If (InStr(1, MM_editRedirectUrl, "?", vbTextCompare) = 0) Then
        MM_editRedirectUrl = MM_editRedirectUrl & "?" & UploadQueryString
      Else
        MM_editRedirectUrl = MM_editRedirectUrl & "&" & UploadQueryString
      End If
    End If
    Response.Redirect(MM_editRedirectUrl)
  End If
End If
%>
```

Thumb Image

Now that we've deleted the image associated with the record, we may also need to delete a thumbnail image--if you're using DMXzone Smart Image Processor to create thumbnails of your uploaded images. Let's quickly look at how to modify our code block to delete the thumbnail image as well. To do this we must set the path to the thumb image well.

In the modified code block below you'll notice I've added a second path, **ThumbImagePath.** Then I call the fileExists function to make sure the thumb image exists. Notice I've modified the file spec passed for the thumb image. I've also modified the file spec passed in the File.DeleteFile method for the thumb image. Now close the If… Then statement and that's all there is to it.

```
    ' Delete the file before we update the record
   If CStr(UploadFormRequest("Delete_Image")) <> "" Or
CStr(UploadFormRequest("Link_Image") <> "") Then
      'create file scripting object
      Set File = CreateObject("Scripting.FileSystemObject")
      ImageFolder = Server.MapPath("..\site_images\")
      ImagePath = ImageFolder & "\" & (UploadFormRequest("upload_org_Link_Image"))
ThumbImagePath = ImageFolder & "\thumb " & (UploadFormRequest("upload org Link Image"))
      'if file exists delete the file
      If fileExists(ImagePath) Then
            File.DeleteFile(ImagePath)
      End If
      If fileExists(ThumbImagePath) Then
            File.DeleteFile(ThumbImagePath)
      End If

      Set File = Nothing
      End If 'end delete file
```

Copyright © 2011 DMXzone.com All Rights Reserved
To get more go to DMXzone.com

Bonus: Displaying Images Using the File Scripting Object

Earlier in this tutorial I mentioned using the file scripting object to prevent the dreaded red X when displaying and image. This is the method I use. Notice that I close the ASP script block before the image tag and start a new script for the closing End If. If you do it this way Dreamweaver will display the image placeholder in design view. If you put it all in one script block and use the Response.Write() method to display the image you will not see the image placeholder in design view.

```
<%Set objFileSystem = Server.CreateObject("Scripting.FileSystemObject")
  objFile = Request.ServerVariables("SCRIPT_NAME")
  ImageFolder = Server.MapPath("../site_images\")
  ImagePath = ImageFolder & "\" & (rs_links.Fields.Item("Link_Image").Value)
  If objFileSystem.FileExists(ImagePath) then%>
<a href="http://<%=(rs_links.Fields.Item("Link_URL").Value)%>" target="_blank">
<img src="../site_images/<%=(rs_links.Fields.Item("Link_Image").Value) %>" hspace="5"
vspace="5" border="0" alt="Link Image" /></a>
<%End If%>
```

Conclusion

That's all there is to it. In this tutorial you learned how to toggle the skip empty fields feature of pure ASP upload, and how to delete the image using the file scripting object to access the server's file system. In the next part of this series, "Taking it to the Limit with Pure ASP Upload 3.0." Next time we will learn how to handle multiple images and sequential images using the For… Each loop. You also learned to prevent the red X of missing images on your site front end using the file scripting object.

All of this code and more will be available in the WWWeb Concepts DataSite™ V4.00 CMS scheduled for release August 15, 20011.

Copyright © 2011 DMXzone.com All Rights Reserved
To get more go to DMXzone.com