Pure ASP.NET Upload Tutorials

Simple File Upload

Simple File Upload

In this tutorial we will explain how to do a simple file upload using the Pure ASP.NET Upload extension.

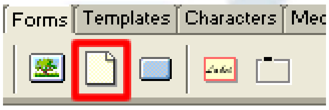
How to do it

To make a simple file upload you need to:

1. **Create a new page**

   You should open a new page in Dreamweaver MX. Of course an existing one will do well also.

2. **Add a file upload field and a form on it**



3. **Add a submit button**

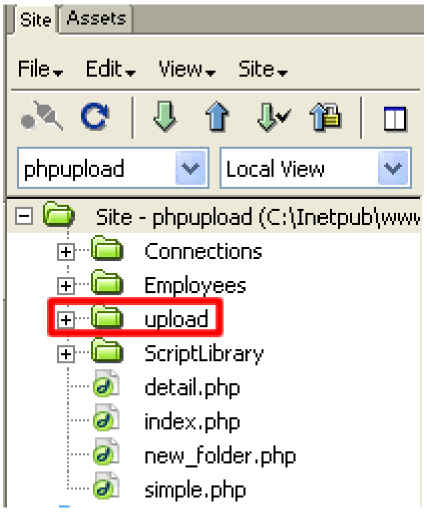
   To be able to submit the form you will need a submit button.

4. **Save the file in your site**

   You will need to save the file in your site because Pure ASP.NET Upload has to know your site location. It will add some more files to it - you will be informed about that so you can upload those also to your server.

5. **Create an upload folder, and give it the appropriate security**

   Usually you will create an upload folder somewhere inside your site.



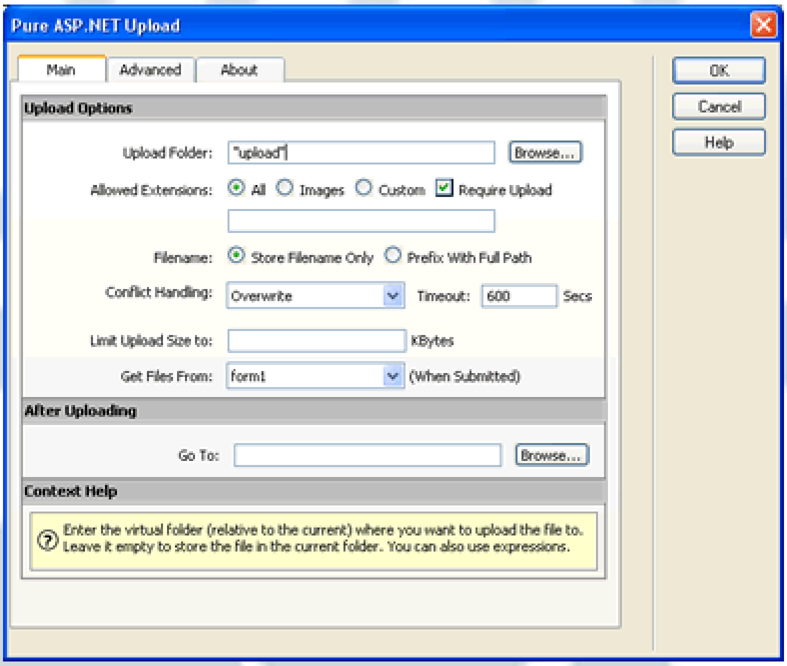
For setting the appropriate security read Setting the folder security on your server tutorial

6. **Insert the Pure ASP.NET Upload server behavior on your page**

Choose the Pure ASP.NET Upload Server Behavior from the Server Behavior palette / DMXzone / Pure ASP.NET Upload category.

7. **Choose the created directory**

For this simple tutorial you don't need to fill in anything in the extension dialog - only select the upload folder and then press OK



Upload Options

Upload Folder: "upload"

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

8. **Save and done**

Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to upload any file.

Optional file upload

Optional file upload

In this tutorial we will explain how to make an optional file upload.

How to do it

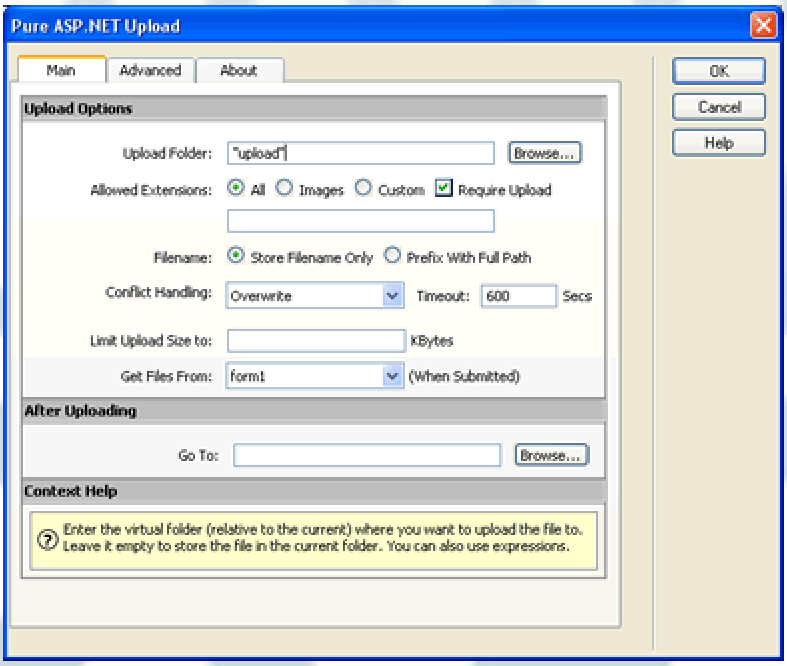
To make an optional file upload, you need to:

1. **Start with basic upload page**

   Start with an upload page where you have Pure ASP.NET Upload on it. If you haven't done it yet, go through the Simple file upload tutorial.

2. **Open the Pure ASP.NET Upload dialog**

   Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone - Pure ASP.NET Upload.



Upload Options

Upload Folder: "upload"

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

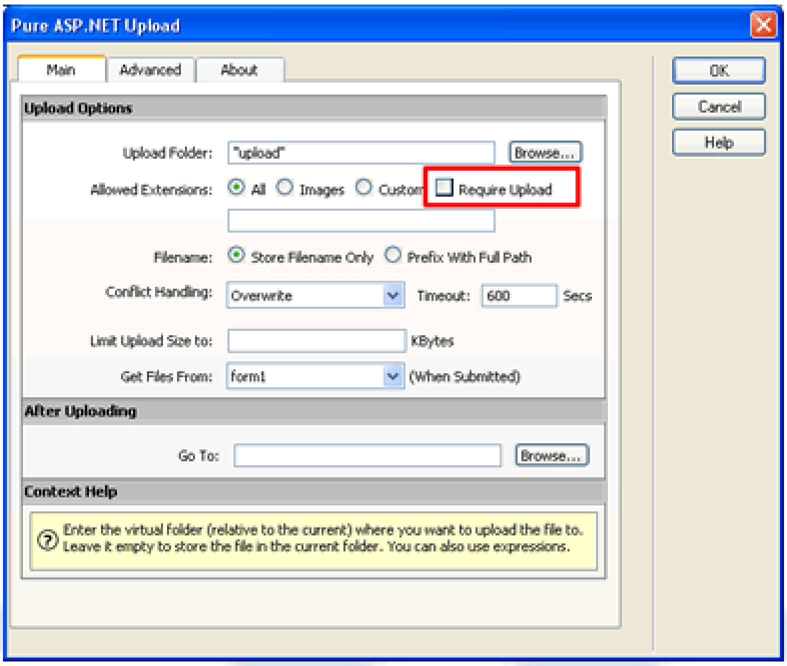
Go To:

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

3. **Make the upload optional**

   When you need to have an optional upload you only have to clear the **Require Upload** checkbox.



Upload Folder: "upload"

Allowed Extensions: ● All ○ Images ○ Custom □ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

Go To:

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

Then the users won't get a warning when they try to submit the page without specifing a file to upload.

4. **Save and done**

Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to skip the file upload now.

Uploading large files

Uploading large files

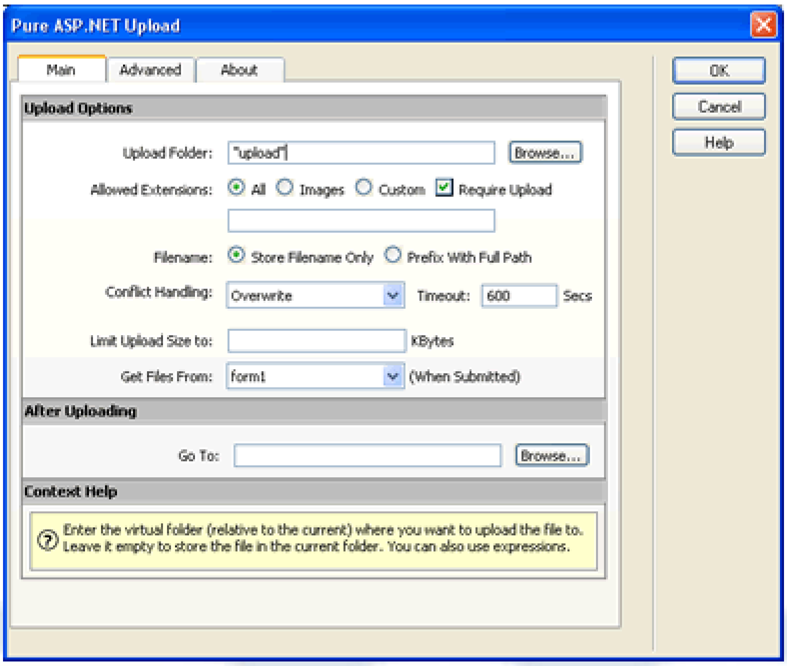
In this tutorial we will explain how to upload large files.

How to do it

To upload large files you need to:

1. **Open the Pure ASP.NET Upload dialog**

Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone - Pure ASP.NET Upload.



Upload Options

Upload Folder: "upload"

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

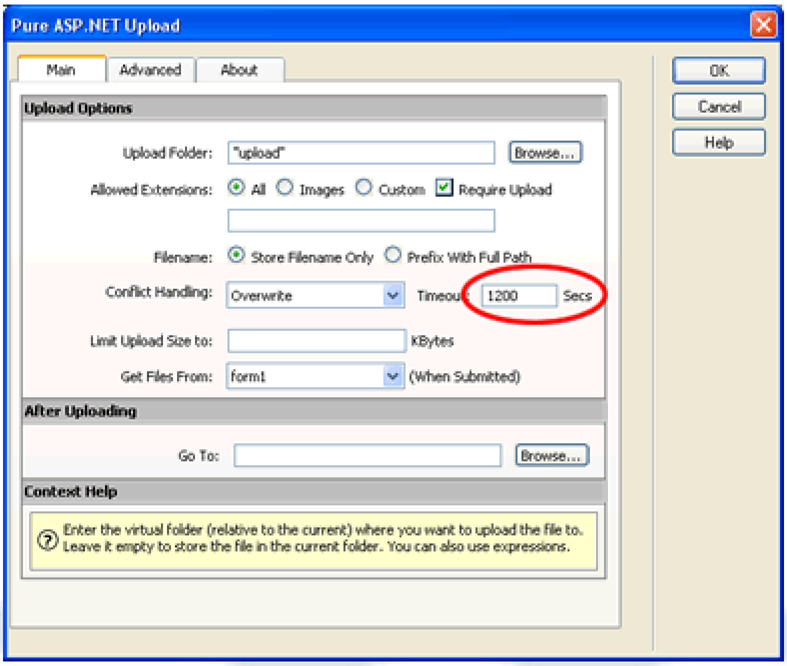
Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

2. **Change the timeout**

When you need to upload large files, you need to increase the script timeout. You can easily calculate this time:

A 28.8kbps modem can download at a speed of approximately 2,5KB/s. So when you need to upload a file of 1MB (equals to 1.024KB) it will take 1.024 / 2.5 = 410 seconds. Always allow extra time so set the **timeout** to at least 500 seconds.

Unfortunately, ASP.NET has another mechanism to avoid deadlocks, which should be taken into account. In the machine.config file (usually located in C:\Windows\Microsoft.NET\framework\v1.0xxx.x\config) there is a setting called "responseDeadlockInterval" of the "processModel" node and this must be set to a larger value than the default "00:03:00". It can also be set to = "Infinite".



Pure ASP.NET Upload

Upload Options

Upload Folder: "upload" Browse...

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To: Browse...

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

When the timeout is not sufficient the user will see a timeout error - so always make the timeout larger than you think is necessary.

3. **Save and done**

Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to upload large files now.

Folder security on your server

Setting the folder security on your server

In this tutorial we will explain the following topics.

- Introduction
- Setting the security on windows XP so **every visitor** can upload
- Setting the security on windows XP so **only selected users** can upload

---

Introduction

In this tutorial we will explain how to set the security on Windows XP so users are able to upload to a specific folder using the Pure ASP.NET Upload extension.

Setting the security so every visitor can upload

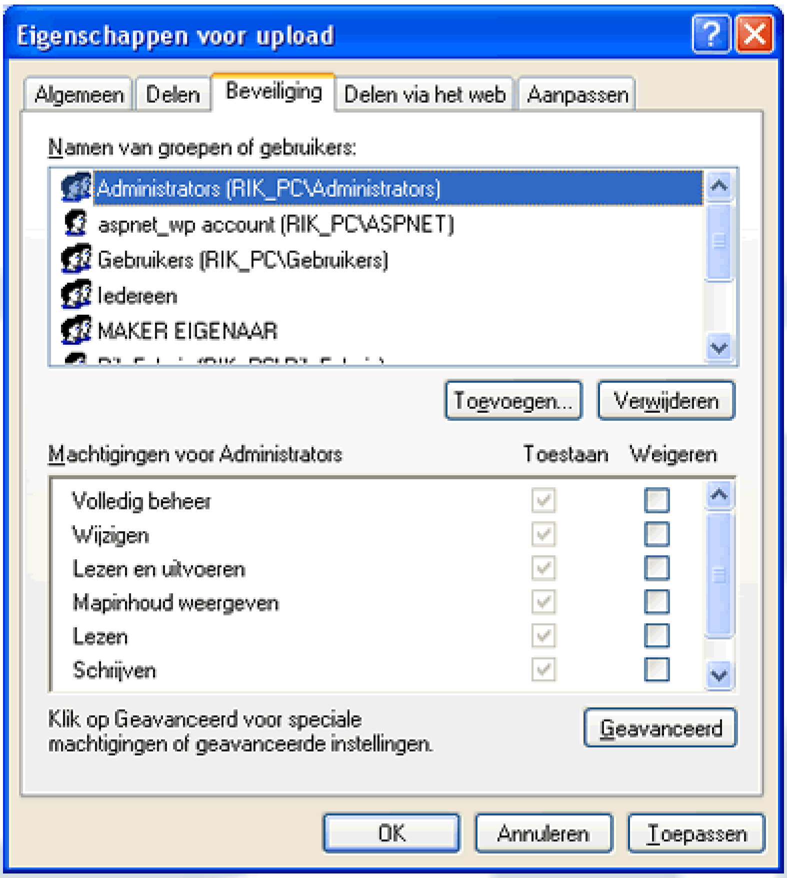
To set the right security on Windows XP so every anonymous visitor can upload you need to:

1. **Find your upload folder on the server.**

   Go to your server and browse to your site with the Windows explorer. Select the upload folder you created.

2. **Change the security settings**

   Right click on the upload folder, select properties, then choose the **security** tab.



Namen van groepen of gebruikers:

- Administrators (RIK_PC\Administrators)
- aspnet_wp account (RIK_PC\ASPNET)
- Gebruikers (RIK_PC\Gebruikers)
- Iedereen
- MAKER EIGENAAR

Machtigingen voor Administrators	Toestaan	Weigeren
Volledig beheer	☑	□
Wijzigen	☑	□
Lezen en uitvoeren	☑	□
Mapinhoud weergeven	☑	□
Lezen	☑	□
Schrijven	☑	□

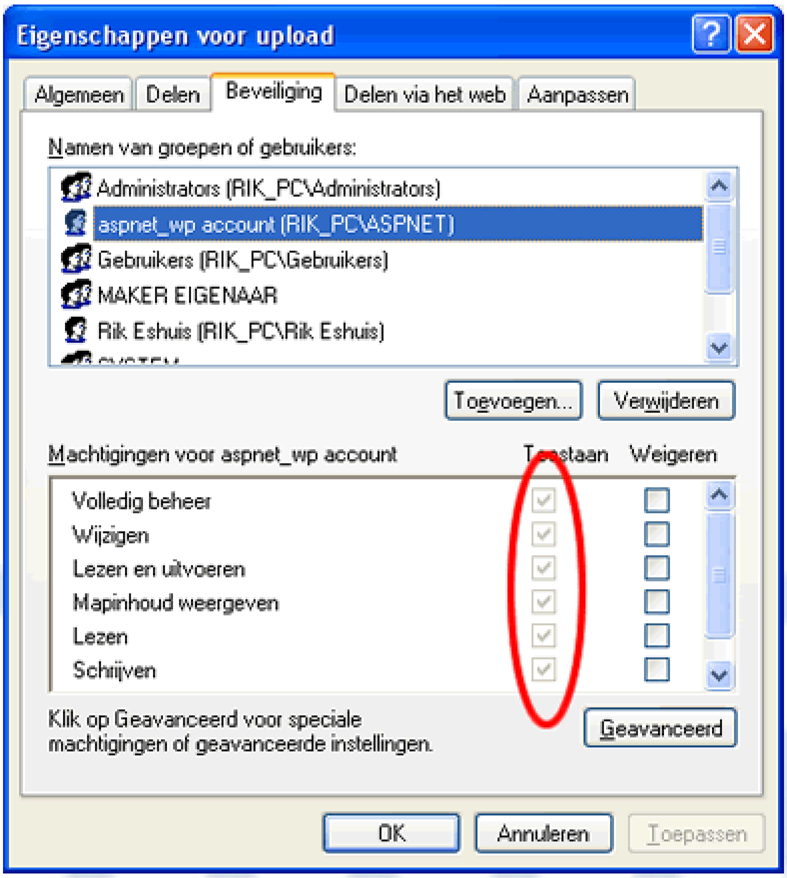
Klik op Geavanceerd voor speciale machtigingen of geavanceerde instellingen.

3. **Add the anonymous user to the list of users**

   We need to give every web visitor permission to upload. Because the users do not login, they all share the same account: **IUSR_<machinename>** where **machinename** is your server name.

   So we need to give this account full write permissions. To do so click on the **"Add..."** button then select the IUSR_<machinename> user and click OK.

   When you are back at the security setting dialog you need to give this user also write permissions - so check the appropriate checkboxes:



Namen van groepen of gebruikers:

- Administrators (RIK_PC\Administrators)
- aspnet_wp account (RIK_PC\ASPNET)
- Gebruikers (RIK_PC\Gebruikers)
- MAKER EIGENAAR
- Rik Eshuis (RIK_PC\Rik Eshuis)

Machtigingen voor aspnet_wp account	Toestaan	Weigeren
Volledig beheer	☑	□
Wijzigen	☑	□
Lezen en uitvoeren	☑	□
Mapinhoud weergeven	☑	□
Lezen	☑	□
Schrijven	☑	□

Klik op Geavanceerd voor speciale machtigingen of geavanceerde instellingen.

4. **Save and you are done**

   Save the security settings and you are ready to go!

Setting the security so only selected users can upload

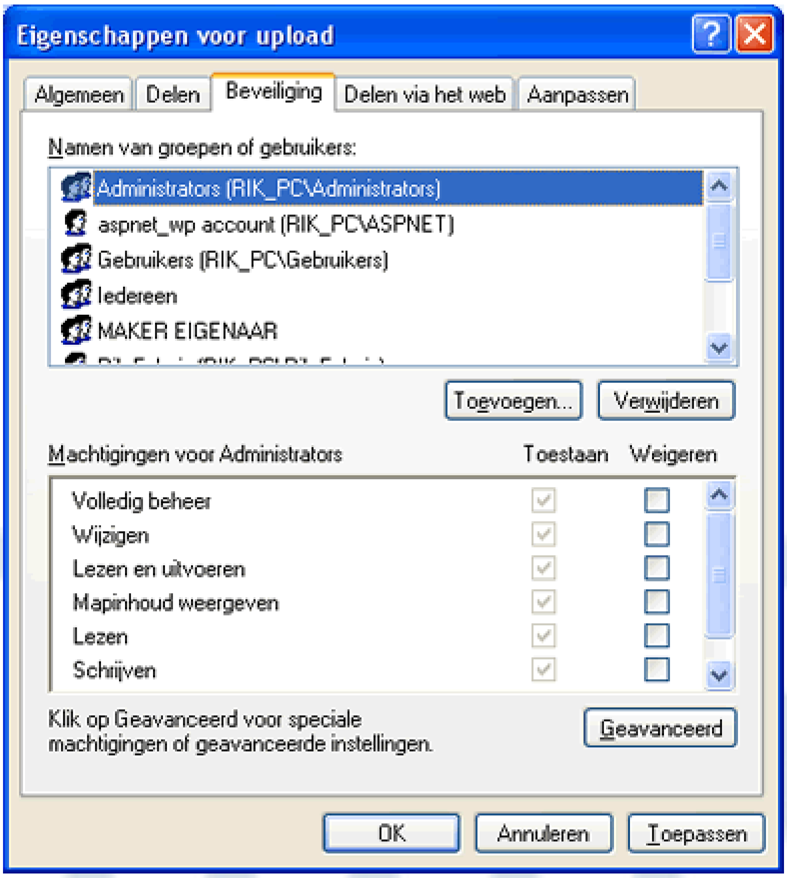
To set the right security on Windows 2000 so only selected users can upload you need to:

1. **Find your upload folder on the server.**

   Go to your server and browse with the Windows explorer your site. Select the upload folder you created.

2. **Change the security settings**

Right click on the upload folder, select properties, then choose the **security** tab.



Namen van groepen of gebruikers:

- Administrators (RIK_PC\Administrators)
- aspnet_wp account (RIK_PC\ASPNET)
- Gebruikers (RIK_PC\Gebruikers)
- Iedereen
- MAKER EIGENAAR

Machtigingen voor Administrators	Toestaan	Weigeren
Volledig beheer	☑	□
Wijzigen	☑	□
Lezen en uitvoeren	☑	□
Mapinhoud weergeven	☑	□
Lezen	☑	□
Schrijven	☑	□

Klik op Geavanceerd voor speciale machtigingen of geavanceerde instellingen.

3. **Remove "Everyone"**

Because we will give only a specific number of users access we need to remove the group **Everyone**. To do so just select the group **Everyone** and click on "**Remove"**.

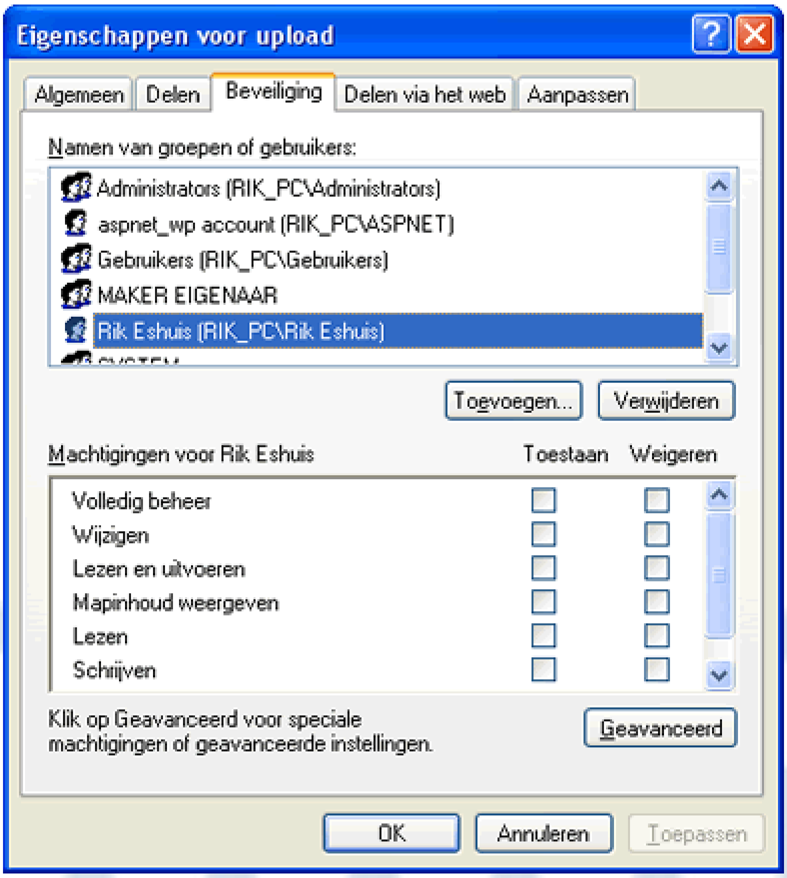
Please note that you might need to clear the "**Allow inheritable permissions**" checkbox, so your specific permissions aren't overwritten. If you do so choose to "**Copy**" existing permissions when asked.

4. **Add the specific users to the list of users**

We need to give every user that needs to upload the right permissions.

To do so click on the "**Add...**" button then select the user login and click OK. Do so for all users that need to be able to upload.

When you are back at the security setting dialog you need to also give the users write permissions - so check the appropriate checkboxes:



Eigenschappen voor upload

Algemeen | Delen | Beveiliging | Delen via het web | Aanpassen

Namen van groepen of gebruikers:

- Administrators (RIK_PC\Administrators)
- aspnet_wp account (RIK_PC\ASPNET)
- Gebruikers (RIK_PC\Gebruikers)
- MAKER EIGENAAR
- Rik Eshuis (RIK_PC\Rik Eshuis)
- SYSTEM

Toevoegen... | Verwijderen

Machtigingen voor Rik Eshuis	Toestaan	Weigeren
Volledig beheer	□	□
Wijzigen	□	□
Lezen en uitvoeren	□	□
Mapinhoud weergeven	□	□
Lezen	□	□
Schrijven	□	□

Klik op Geavanceerd voor speciale machtigingen of geavanceerde instellingen.

Geavanceerd

OK | Annuleren | Toepassen

5. **Save and you are done**

   Save the security settings and you are ready to go!

Upload to variable folder

Upload to variable folder

In this tutorial we will explain how to do a file upload to a folder that is chosen dynamically. This means that your folder name can vary. An example of this is when you want to have the user name as a folder name.

How to do it

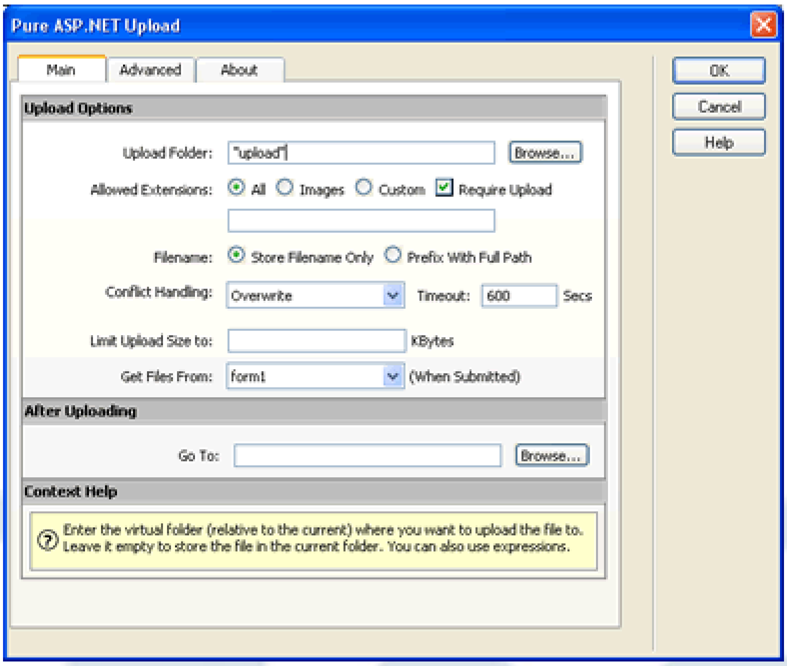
To make the file upload to a variable folder, you need to:

1. **Start with basic upload page**

   Start with an upload page with the Pure ASP.NET Upload in place. If you haven't done it yet, go through the Simple file upload tutorial.

2. **Open the Pure ASP.NET Upload dialog**

Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone - Pure ASP.NET Upload.



Pure ASP.NET Upload

Upload Options

Upload Folder: "upload"

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

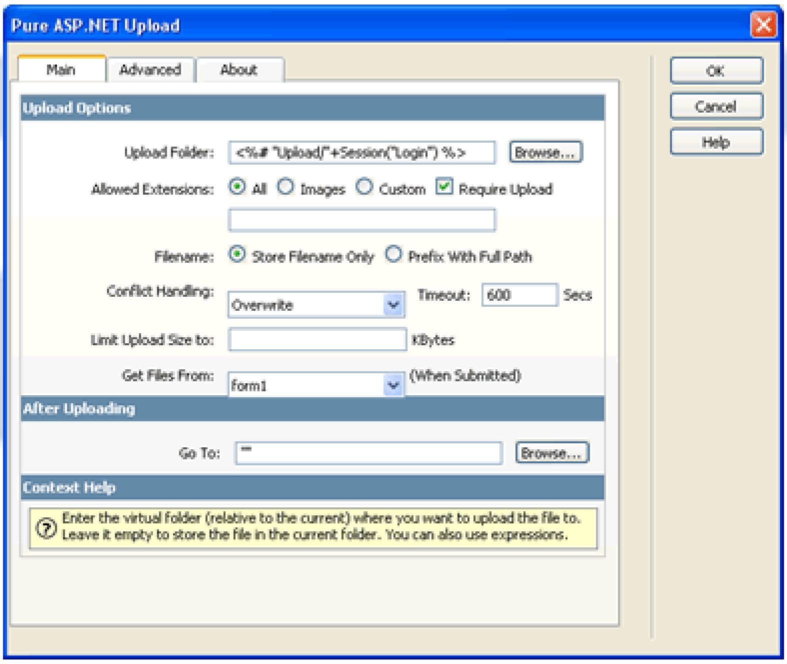
Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

3. **Choose the variable folder**

When you need to use a variable folder, you can enter any expression you want in the **Upload Folder** field. For example if I have a **Session** variable that has the user login in it, I could write:

```
<%# "Upload/"+Session("Login") %>; for VB#
<%# "Upload/"+Session["Login"] %>; for C#
```

this will create a new directory for every user that tries to upload! Non existent folders will be **auto-created** so this is really powerful! You only need to setup the security on the main **Upload** folder, for more info how to do this read the Setting the folder security on your server tutorial.



Pure ASP.NET Upload

Upload Options

Upload Folder: <%# "Upload/"+Session("Login") %>

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To: ""

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

You might also check the **Prefix With Full Path** option. This option will store the full path in your database (if you are saving the filename in a database). So it will be easy to retrieve a list of all files later on.

4. **Save and done**

   Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to upload any file to a dynamic folder.

Upload to physical folder

Upload to physical folder - upload your database file

In this tutorial we will explain how to do a file upload to a folder that is not relative to your site. This means that your folder can be located in a very different directory or disk on the server. An example of this is when you want to upload to your database.

How to do it

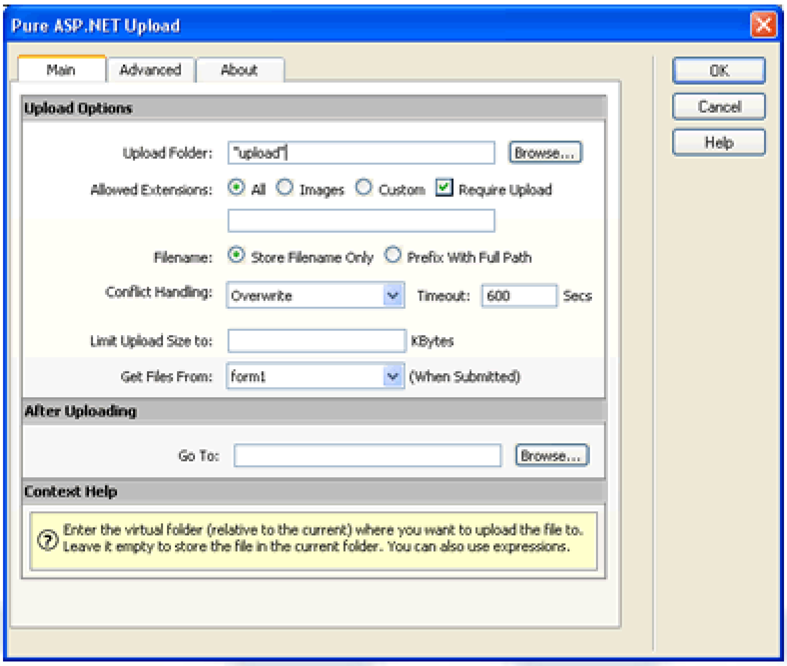
To upload to a physical folder you need to:

1. **Start with basic upload page**

   Start with an upload page where you have Pure ASP.NET Upload on it. If you haven't done it yet, go through the Simple file upload tutorial.

2. **Open the Pure ASP.NET Upload dialog**

   Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone -> Pure ASP.NET Upload.



Upload Options

Upload Folder: "upload"

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

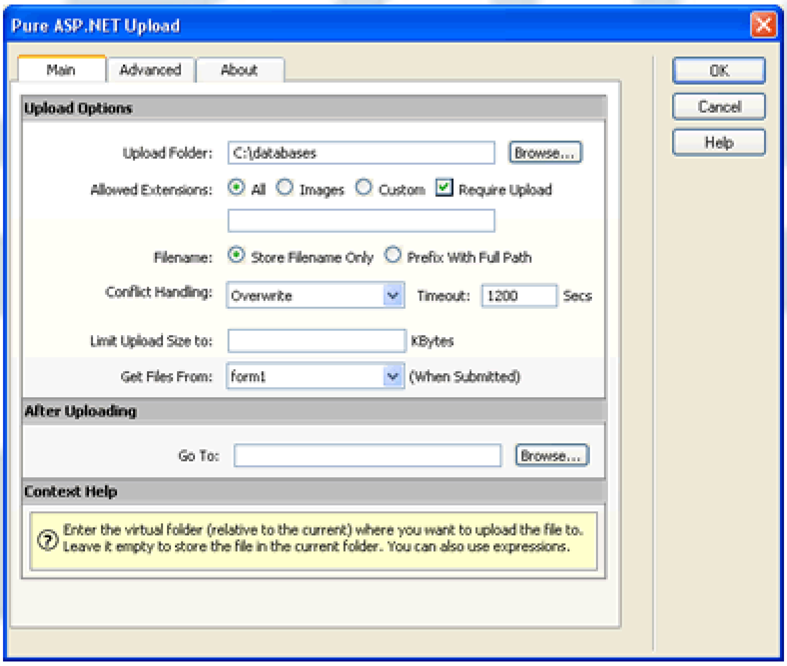
Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

3. **Choosing the variable folder**

When you need to use a variable folder, you can enter the full path - including the drive in the **Upload Folder** field. For example:

```
"c:\databases"
```

this will upload the files to this path on the server. You only need to setup the right security on this folder, for more info how to do this read the Setting the folder security on your server tutorial.



Upload Options

Upload Folder: C:\databases

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

You might also check the **Prefix With Full Path** option. This option will store the full path in your database (if you are saving the filename in a database). So it will be easy to retrieve a list of all files later on.

4. **Save and done**

   Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to upload any file to a physical folder.

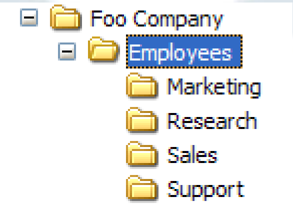
Upload to Organized Folder

Upload to Organized Folder

Many users have asked me how to use this new feature of Pure ASP.NET Upload for organizing the uploaded files in different folders. In this tutorial I will explain how this is accomplished.

In sites where many files are going to be uploaded you often need a better way to organize them then to just upload them all in one and the same folder.

Let's take an intranet site where you will be uploading photo's of all employees as an example. You want those photos to be organized in different folders depending on the employee's department. The folder structure you want might look like this:



How to do it

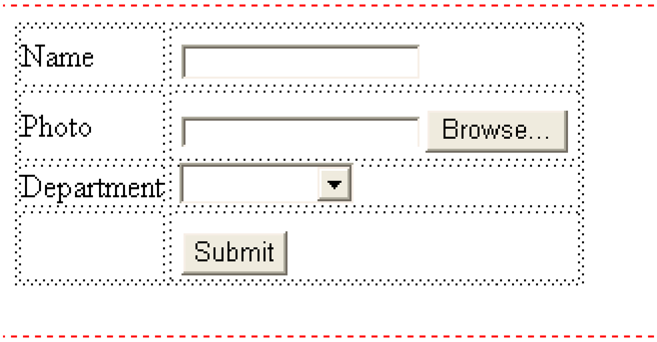
To make the file upload to an organized folder, you need to:

1. **Start with a basic upload page**

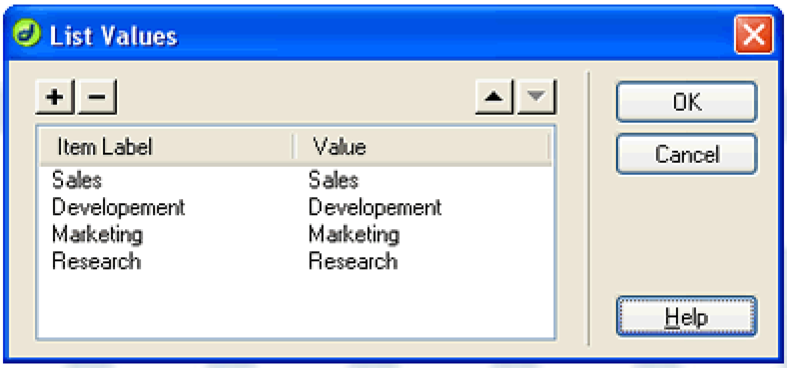
   Start with an upload page with the Pure ASP.NET Upload extension in place. If you haven't done this yet, go through the Simple file upload tutorial.

2. **Extend the upload form with Department selection field**

   You can now edit the basic form and add some more form fields to it:



As you can see I've added a Department field. This is a drop down list that includes all the departments we have. As values I've selected the same strings as the department names. I will use the name of the department to make the upload folder name.



Item Label	Value
Sales	Sales
Developement	Developement
Marketing	Marketing
Research	Research

3. **Open the Pure ASP.NET Upload dialog**

Select the Pure ASP.NET Upload dialog by opening the Server Behaviors panel. Select DMXzone - Pure ASP.NET Upload.

4. **Enter the variable folder**

When you need to use a variable folder, you can enter any expression you want in the **Upload Folder** field. In my example another form field contains the Departments.

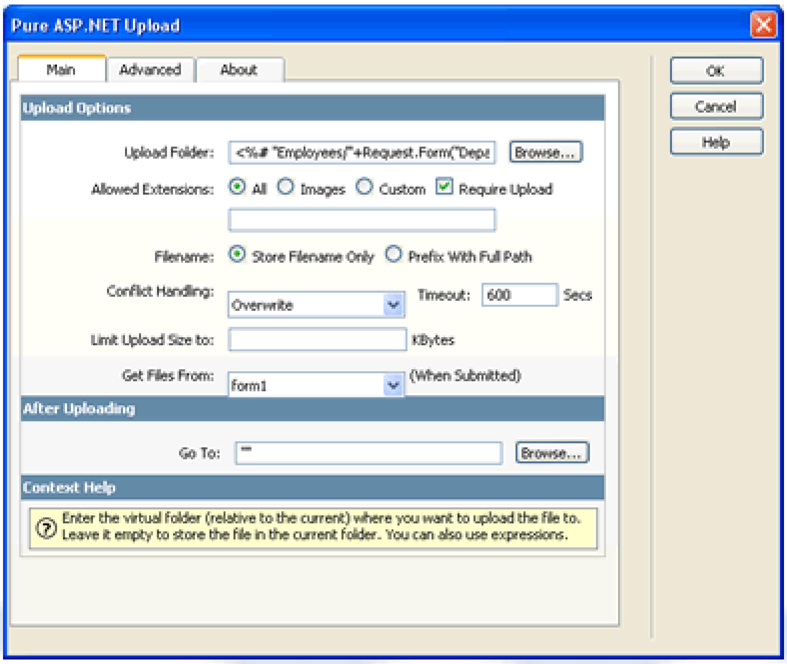
So I need the following upload folder expression for VB#:

```
<%# "Employees/"+Request.Form("Department") %>
```

So I need the following upload folder expression for VB#:

```
<%# "Employees/"+Request.Form["Department"] %>
```

this will create a new directory for every non existed department and will place the photos of each employee that belongs to that department there! Non existent folders will be **auto-created** so this is really powerful! You only need to setup the security on the main **Upload** folder, for more info how to do this read the Setting the folder security on your server tutorial.



Upload Folder: <%# "Employees/"+Request.Form("Depa

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite

Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

Go To: ""

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

You might also check the **Prefix With Full Path** option. This option will store the full path in your database (if you are saving the filename in a database). So it will be easy to retrieve a list of all files later on.

5. **Save and done**

   Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to upload any file to a dynamic folder.

Restricting upload file types

Restricting upload file types

In this tutorial we will explain how to only allow specific files to be uploaded. You can restrict unwanted file types from being uploaded (depending on their extension).

How to do it

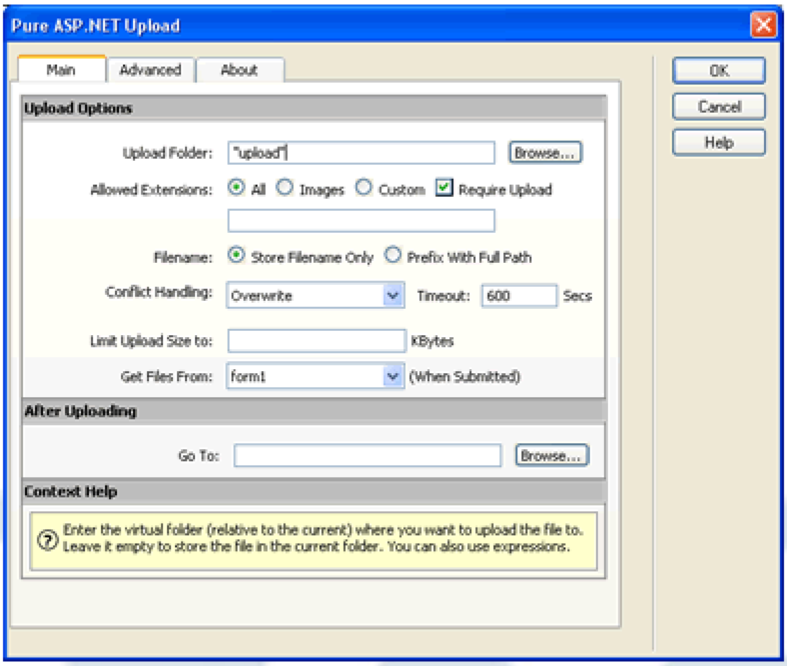
To make the file upload with restricted file types, you need to:

1. **Start with basic upload page**

   Start with an upload page where you have Pure ASP.NET Upload on it. If you haven't done it yet, go through the Simple file upload tutorial.

2. **Open the Pure ASP.NET Upload dialog**

Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone - Pure ASP.NET Upload.



Pure ASP.NET Upload

Upload Options

Upload Folder: "upload"

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

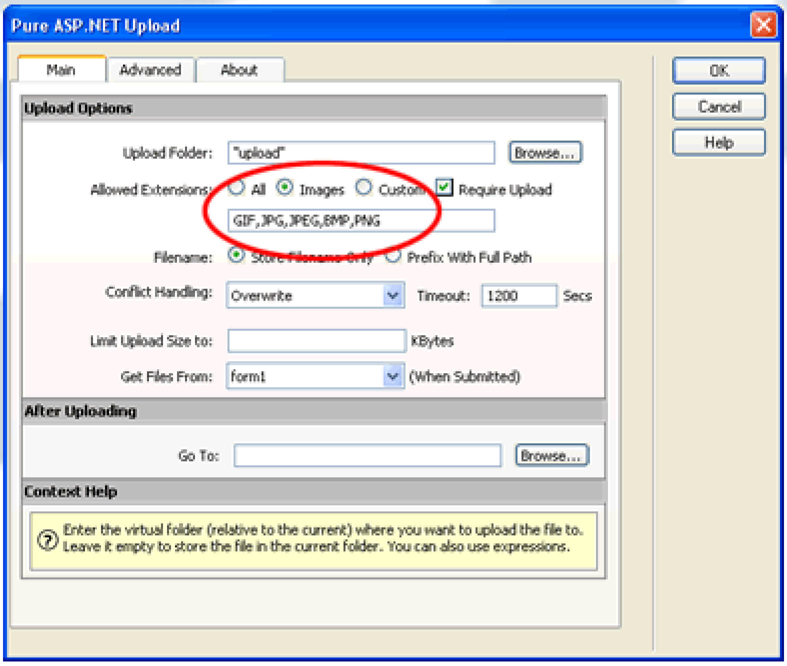
Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

3. **Select the Allowed Extensions**

When you need to restrict the uploaded files to specific number of extensions, you need to select the **Allowed Extensions** field.

First choose the extensions type. **All** means no restriction, **Images Only** will fill in the most common image extensions and **Custom** will allow you to enter your own comma separated list of allowed extensions.

In this example we've chosen to only allow our users to upload **Images Only**.



Pure ASP.NET Upload

Upload Options

Upload Folder: "upload"

Allowed Extensions: ○ All ● Images ○ Custom ☑ Require Upload

GIF,JPG,JPEG,BMP,PNG

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

So now you choose for example to upload **Images Only**.

4. **Save and done**

   Save your page and upload it to your server, and you are done! Go to your website and test it - you should only be able to upload images now.

Restricting file size

Restricting file size

In this tutorial we will explain how to restrict the maximum size of the uploaded files.

There are two types of file size checking:

- check on the client browser, **before file is uploaded** - this is done when the files are images
- on the server, **after file is uploaded** - for the other file types.

How to do it

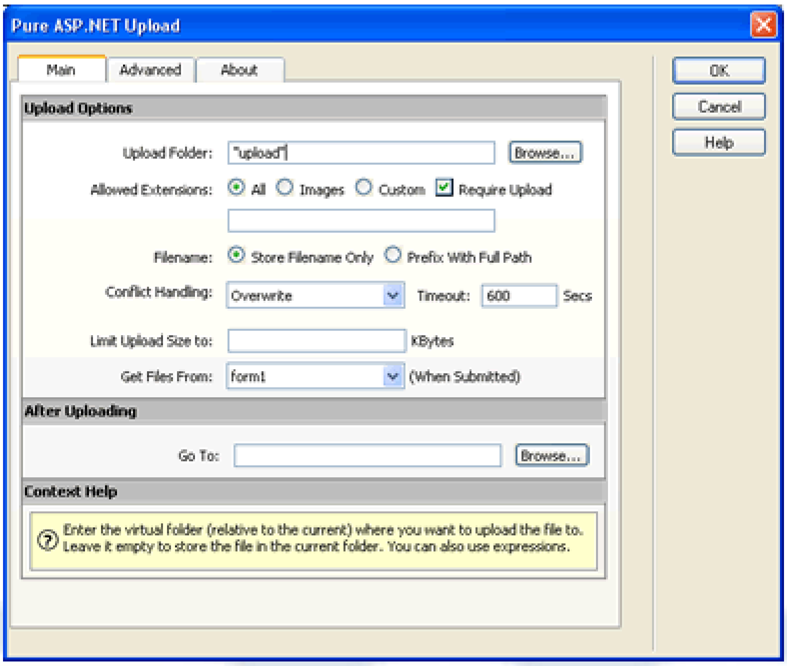
To make the file upload with restricted file size, you need to:

1. **Start with basic upload page**

   Start with an upload page where you have Pure ASP.NET Upload on it. If you haven't done it yet, go through the Simple file upload tutorial.

2. **Open the Pure ASP.NET Upload dialog**

   Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone - Pure ASP.NET Upload.



Upload Options

Upload Folder: "upload" Browse...

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 600 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To: Browse...

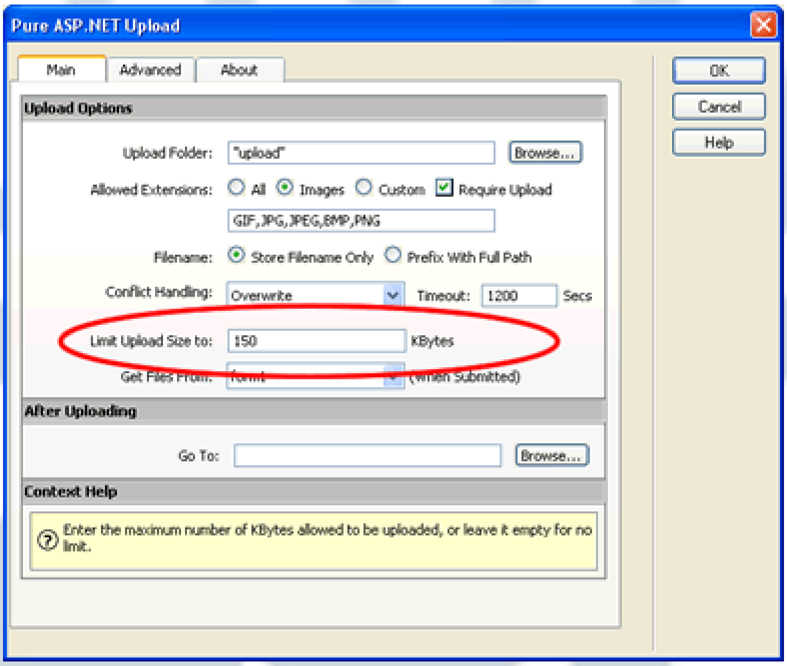
Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

3. **Select the Allowed Extensions**

When you need to restrict the uploaded files to a specific size, you need to select the **Limit Upload Size To** field and enter the maximum size in KBytes.

Please note that 1024 Kbytes = 1MB
(KBytes = Kilo Bytes, MB = Mega Bytes)



Upload Options

Upload Folder: "upload" Browse...

Allowed Extensions: ○ All ● Images ○ Custom ☑ Require Upload

GIF,JPG,JPEG,BMP,PNG

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: 150 KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To: Browse...

Context Help

Enter the maximum number of KBytes allowed to be uploaded, or leave it empty for no limit.

In this example, we've chosen to limit the file upload size to 150 KBytes.

4. **Save and done**

Save your page and upload it to your server, and you are done! You shouldn't be able to upload images that are larger than 150 KBytes.

Restricting image width and height

Restricting image width and height

In this tutorial we will explain how to set minimum and maximum width and height for the uploaded images.

There are two types of image size checking:

- check on the client browser, **before file is uploaded** - this is done when the files are images
- on the server, **after file is uploaded** - for the other file types.

How to do it

To make the file upload with restricted image size, you need to:

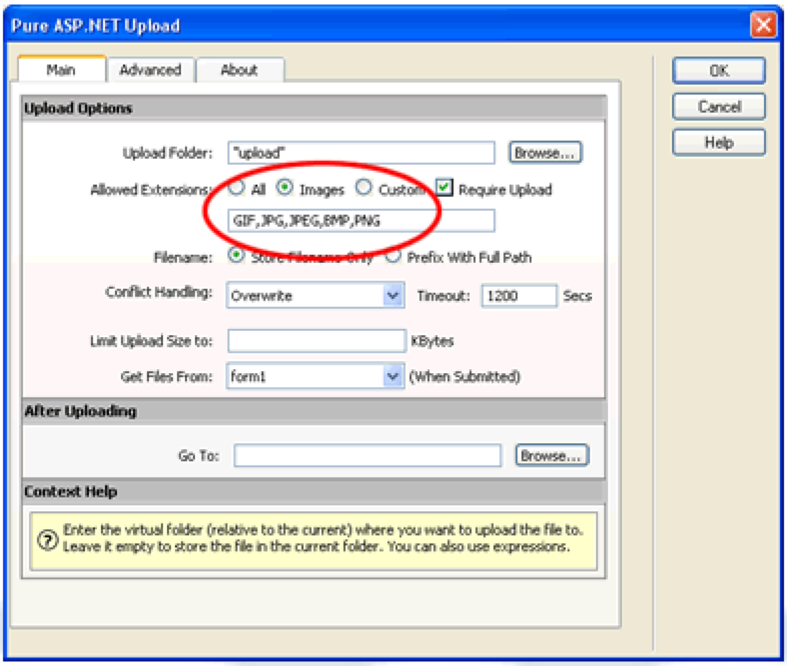
1. **Start with basic upload page**

   Start with an upload page where you have Pure ASP.NET Upload on it. If you haven't done it yet, go through the Simple file upload tutorial.

2. **Open the Pure ASP.NET Upload dialog**

   Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone - Pure ASP.NET Upload.

   Make sure you've selected **Images Only** for your allowed extensions.



Upload Options

Upload Folder: "upload"

Allowed Extensions: ○ All ● Images ○ Custom ☑ Require Upload

GIF,JPG,JPEG,BMP,PNG

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

3. **Select the Allowed Image Size**

When you need to restrict the size of the uploaded images, you need to fill in the minimum and/or maximum width and height on the Advanced tab. If you leave one of those values empty it means it is unlimited. The sizes are in pixels.

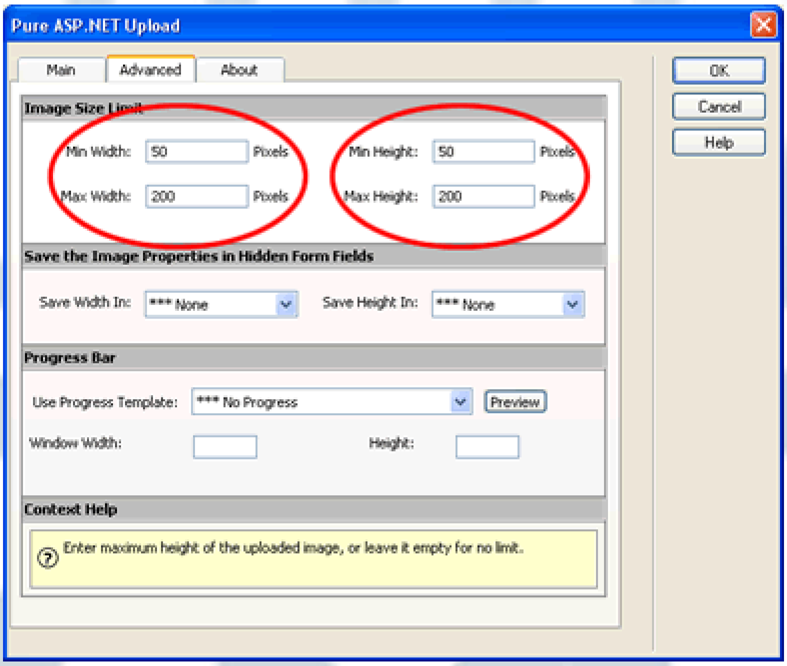


Image Size Limit

Min Width: 50 Pixels Min Height: 50 Pixels

Max Width: 200 Pixels Max Height: 200 Pixels

Save the Image Properties in Hidden Form Fields

Save Width In: *** None Save Height In: *** None

Progress Bar

Use Progress Template: *** No Progress

Window Width: Height:

Context Help

Enter maximum height of the uploaded image, or leave it empty for no limit.

In our example, we've chosen to limit the image size to min 50x50 and max 200x200 pixels.

4. **Save and done**

Save your page then, upload it to your server, and you are done! Go to your website and test it - you should be able to upload images only that are not bigger then 200x200 pixels and minimum 50x50 pixels.

Upload and insert record form

Upload and insert record form

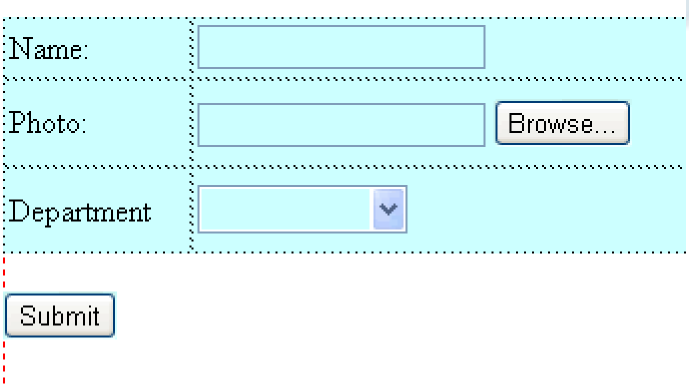
In this tutorial we will explain how to upload files and other fields and save those to a database. The files will be saved on the server and their names and the other form fields will be saved in the database.

How to do it

To make file upload and insert a record, you need to:

1. **Make your form**

   You can start with any page where you have a form on it with one or more entry fields. You need to add a file upload field if you haven't got one yet. If you haven't done it yet, go through the Simple file upload tutorial. So your form will look like:

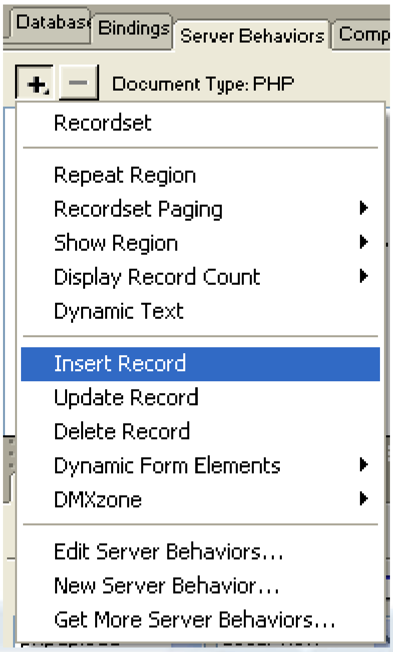


2. **Save the page in your site**

   After you've completed your form, you should save the page in the site you are currently working on.

3. **Add Insert Record Server Behavior**

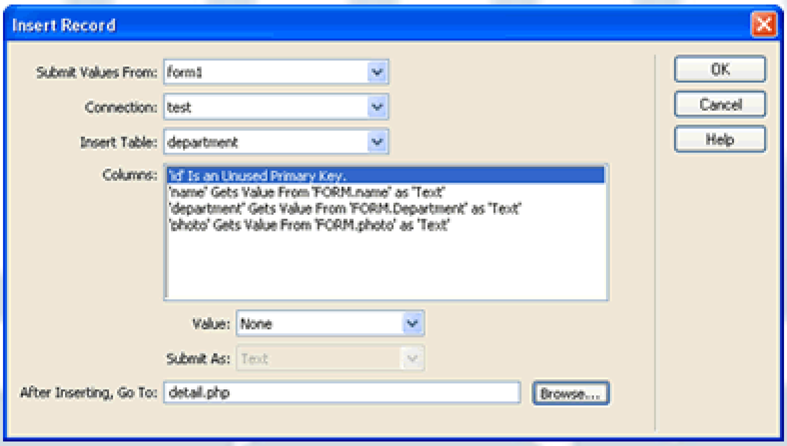
   Choose the **Insert Record** Server Behavior from the Server behaviors palette.



Select your database connection and table where you want to save the data in.

Assign all the entry fields from the form to the corresponding fields in your database.

Enter a redirect page at **After Inserting, Go To** this will be the follow up page.



Insert Record

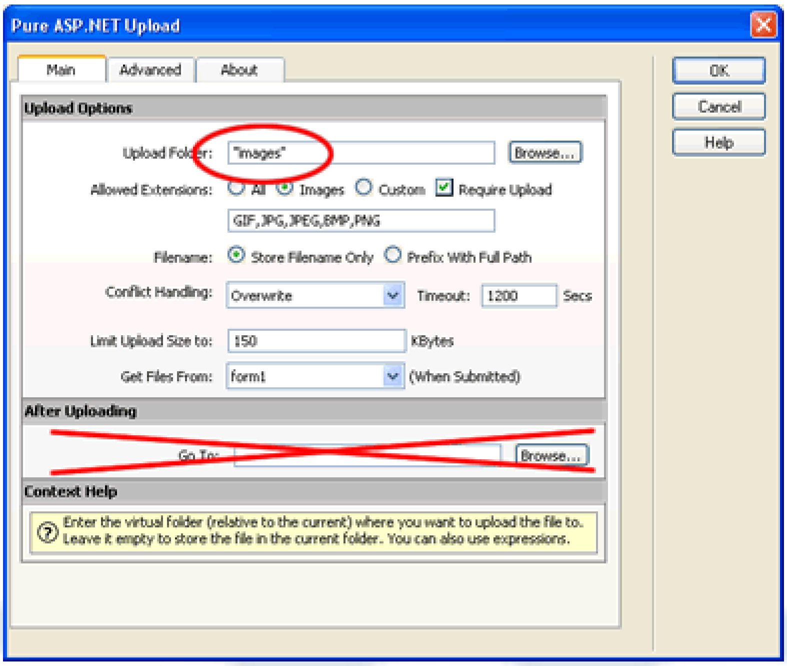
Submit Values From:	form1
Connection:	test
Insert Table:	department
Columns:	'id' Is an Unused Primary Key. 'name' Gets Value From 'FORM.name' as 'Text' 'department' Gets Value From 'FORM.Department' as 'Text' 'photo' Gets Value From 'FORM.photo' as 'Text'
Value:	None
Submit As:	Text
After Inserting, Go To:	detail.php

Click OK to insert the server behavior.

4. **Add the Pure ASP.NET Upload extension**

Select the Pure ASP.NET Upload dialog by choosing it from the Server Behaviors palette, DMXzone / Pure ASP.NET Upload.

Choose upload folder, Images only but **don't enter** a redirect url in the **Go To** field. This is because you already have a redirect in the **Insert Record** server behavior.



Upload Options

Upload Folder: "images" Browse...

Allowed Extensions: ○ All ● Images ○ Custom ☑ Require Upload

GIF,JPG,JPEG,BMP,PNG

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: 150 KBytes

Get Files From: form1 (When Submitted)

After Uploading

~~Go To: Browse...~~

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

To read more about uploading images read Allowing only images upload tutorial.

To read more about limiting the image size read Restricting image width and height tutorial

5. **Save and done**

Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to upload images together with other form fields and all fields should be saved as a new record in your database.

Upload and update record form

Upload and update record form

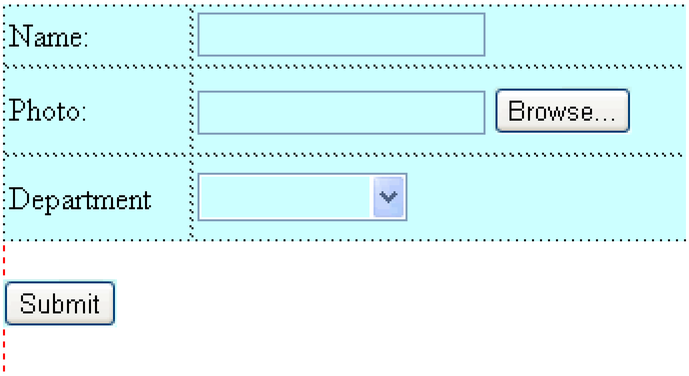
In this tutorial we will explain how to upload files and other fields and update those in the database. The files will be saved on the server and their names and the other form fields will be updated in the database.

How to do it

To make file upload and update a record, you need to:

1. **Make your form**

You can start with an any page where you have a form on it with one or more entry fields. You need to add a file upload field if you haven't got one yet. If you haven't done it yet, go through the Simple file upload tutorial. So your form will look like:



Name:

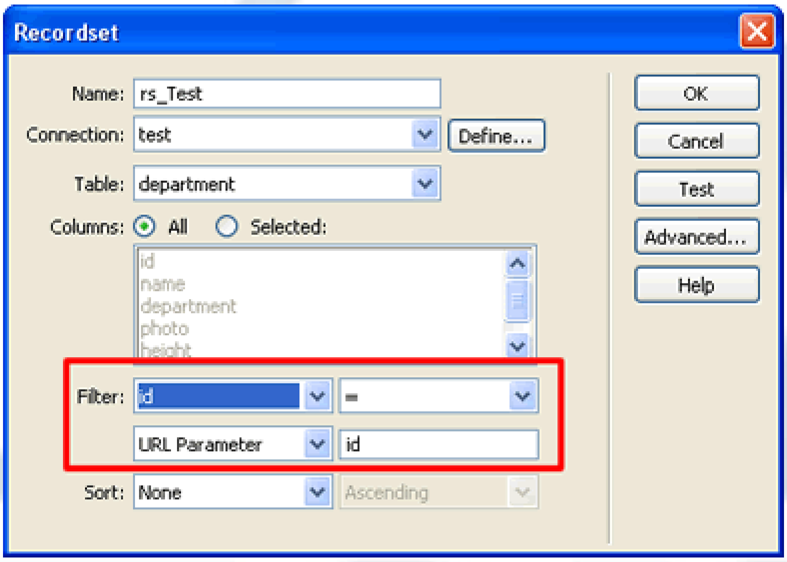
Photo:

Department

TIP: You might also reuse the form you created for Upload and insert record form tutorial, but delete all server behaviors first.

2. **Add a Recordset**

You will need a recordset that will contain the current record. This record will be updated. Usually you filter this recordset with a parameter passed in the url.



Recordset

Name: rs_Test

Connection: test

Table: department

Columns: ● All ○ Selected:

id
name
department
photo
height

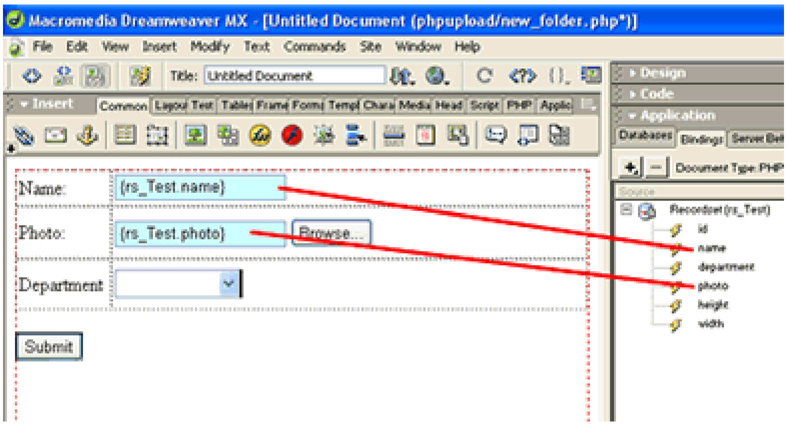
Filter: id =

URL Parameter id

Sort: None Ascending

3. **Bind your database fields to the form**

After you have inserted a recordset, you should bind the recordset fields to the form fields. You can do that by dragging and dropping the fields on top of the form fields.


Macromedia Dreamweaver MX - [Untitled Document (phpupload/new_folder.php*)]
File Edit View Insert Modify Text Commands Site Window Help
Title: Untitled Document
Insert
Design
Code
Application
Databases Bindings Server Behaviors
Document Type: PHP
Source
Recordset (rs_Test)
id
name
department
photo
height
width

Name: {rs_Test.name}

Photo: {rs_Test.photo} Browse...

Department

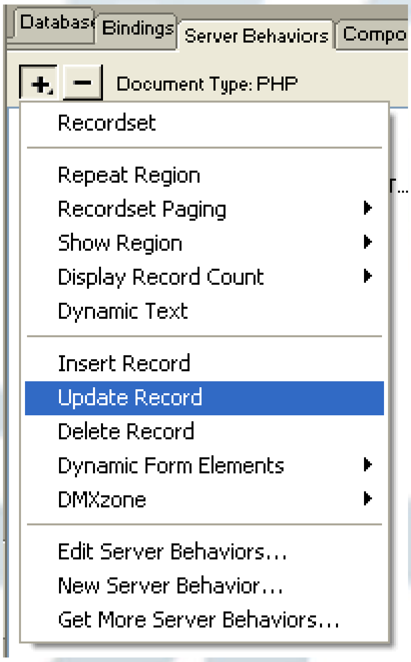
Submit

4. **Save the page in your site**

After you've completed your form, you should save the page in the site you are currently working on.

5. **Add Update Record Server Behavior**

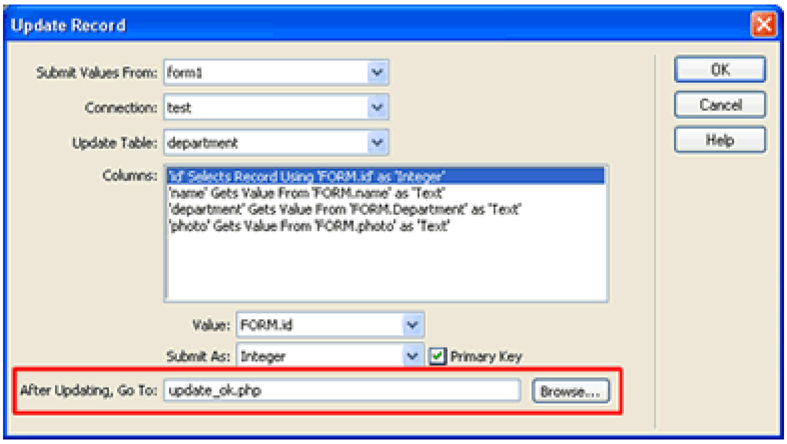
Choose the **Update Record** Server Behavior from the Server behaviors palette.



Select your database connection and table where you want to save the data in.

Assign all the entry fields from the form to the corresponding fields in your database.

Enter a redirect page at **After Updating, Go To** this will be the follow up page.



Update Record

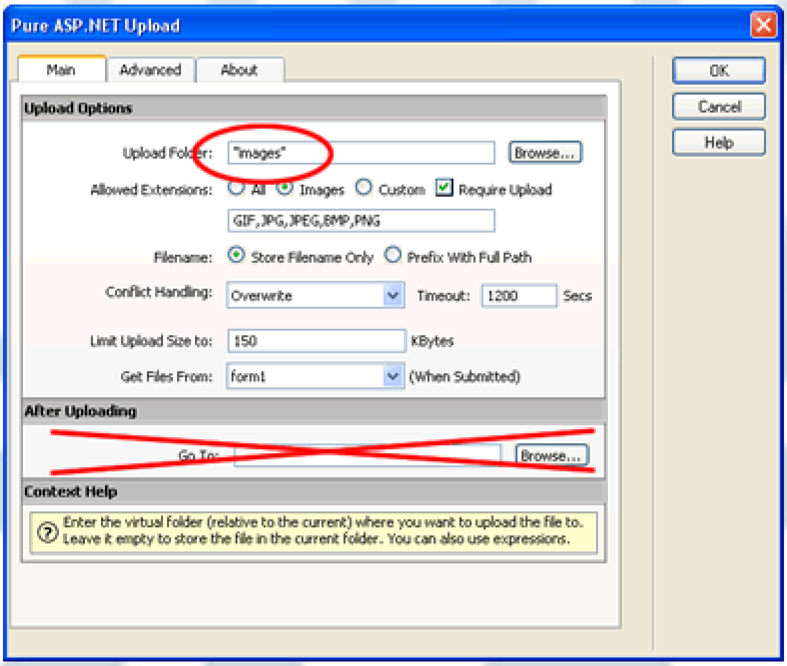
Submit Values From: form1
Connection: test
Update Table: department
Columns:
'id' Selects Record Using 'FORM.id' as 'Integer'
'name' Gets Value From 'FORM.name' as 'Text'
'department' Gets Value From 'FORM.Department' as 'Text'
'photo' Gets Value From 'FORM.photo' as 'Text'
Value: FORM.id
Submit As: Integer ☑ Primary Key
After Updating, Go To: update_ok.php

Click OK to insert the server behavior.

6. **Add the Pure ASP.NET Upload extension**

Select the Pure ASP.NET Upload dialog by choosing it from the Server Behaviors palette, DMXzone / Pure ASP.NET Upload.

Choose upload folder, Images only but **don't enter** a redirect url in the **Go To** field. This is because you already have a redirect in the **Upload Record** server behavior.



Pure ASP.NET Upload

Upload Options
Upload Folder: "images"
Allowed Extensions: ○ All ● Images ○ Custom ☑ Require Upload
GIF,JPG,JPEG,BMP,PNG
Filename: ● Store Filename Only ○ Prefix With Full Path
Conflict Handling: Overwrite Timeout: 1200 Secs
Limit Upload Size to: 150 KBytes
Get Files From: form1 (When Submitted)
After Uploading
Go To:
Context Help
Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

To read more about uploading images read Allowing only images upload tutorial.

To read more about limiting the image size read Restricting image width and height tutorial

7. **Save and done**

Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to upload images together with other form fields and all fields should be updated in your database. Don't forget to pass an existing **Id** as a parameter.

List of records with images

Making a list of records with images

In this tutorial we will explain how to make a nice list of records with images. A detail page after the upload page.

How to do it

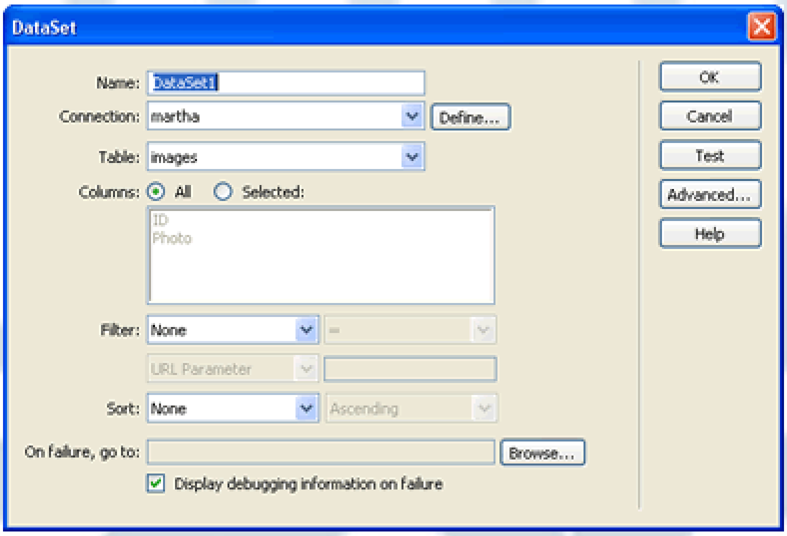
To make a list of images you have to:

1. **Make the new list page**

   You can start with a new blank page.

2. **Add a Dataset**

   You will need a dataset to get a list of all records.



DataSet

Name: DataSet1
Connection: martha
Table: images
Columns: ● All ○ Selected:
ID
Photo
Filter: None --
URL Parameter
Sort: None Ascending
On failure, go to:
☑ Display debugging information on failure

3. **Insert File Name**

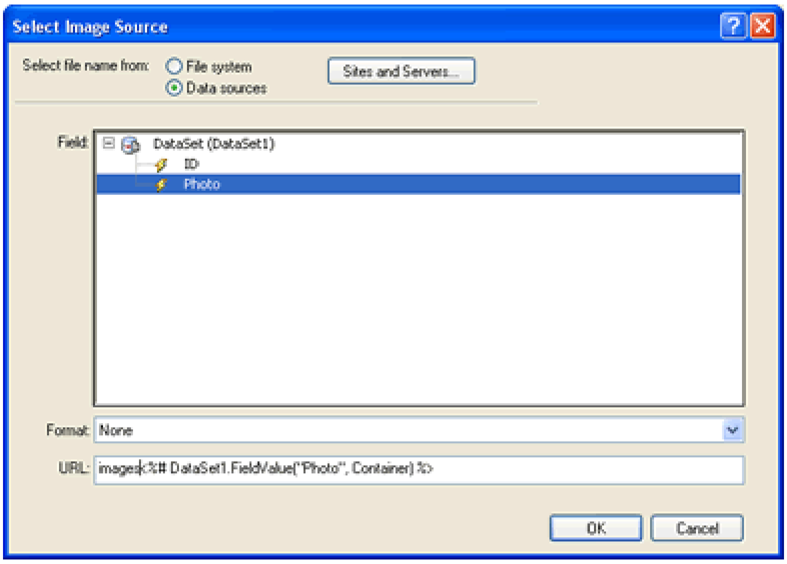
   Put your cursor on the page. Open the Bindings Panel. Select Name and click insert.

   You will see **{Dataset1.name}** on your page.

4. **Add the image**

   Next we will insert the image.

   Then click on **Insert - Image**. Instead of selecting an existing image from the **File system**, you should choose **Data Sources !**



Select Image Source

Select file name from: ○ File system ● Data sources

Field: DataSet (DataSet1)
ID
Photo

Format: None

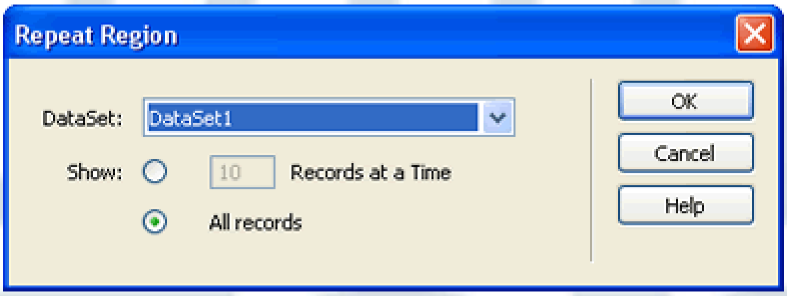
URL: images<%# DataSet1.FieldValue("Photo", Container) %>

Then select the **name** field from the Dataset1 dataset. Because we uploaded all pictures to the **images** folder and store only the filename in the database, you need to put **images/** before the database field.

5. **Add a Repeat Region**

Now that we have inserted the image, we need to make a repeat region from this record. So you can see a list of all records.

First select the whole first row, then choose **Repeat Region** from the Server Behaviors palette.



Repeat Region

DataSet: DataSet1

Show: ○ 10 Records at a Time
● All records

Now we have a full list of all existing images in the database.

6. **Save and done**

Save your page and upload it to your server, and you are done! Go to your website and test it - you should be able to list records!

Saving the image size in the database

Saving the image size in the database

In this tutorial we will explain how to save the image size in the database.

How to do it

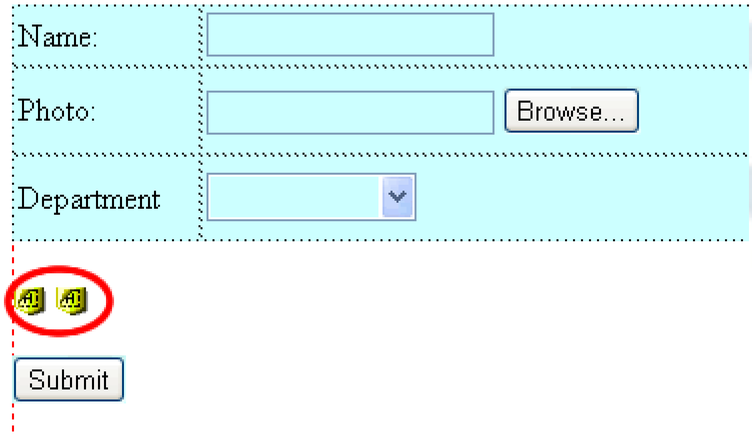
To save the uploaded image size in the database, you need to:

1. **Start with insert or update record upload page**

   Start with an insert or update record page where you have Pure ASP.NET Upload on it. We have created those pages in the Upload and insert record form and Upload and update record form tutorials.

2. **Add two hidden fields to your form**

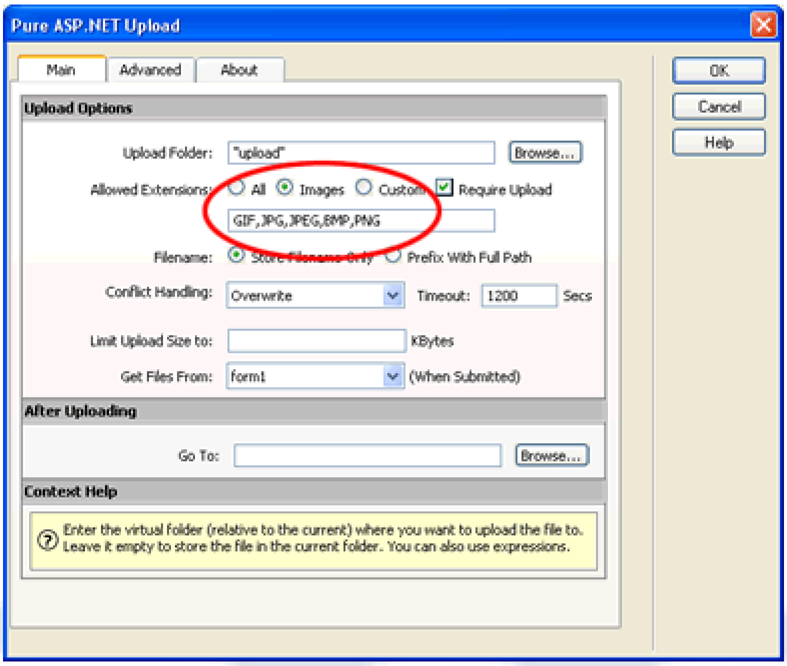
   To save the image size in temporary fields first, you need two hidden fields. Insert those and name them **imageWidth** and **imageHeight** for example.



3. **Open the Pure ASP.NET Upload dialog**

   Select the Pure ASP.NET Upload dialog by opening the Server Behaviors Panel. Select DMXzone - Pure ASP.NET Upload.

   Make sure you've selected **Images Only** for your allowed extensions.



Upload Options

Upload Folder: "upload"

Allowed Extensions: ○ All ● Images ○ Custom ☑ Require Upload

GIF,JPG,JPEG,BMP,PNG

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

Go To:

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

4. **Choose the save fields**

Go to the **Advanced** tab of the Pure ASP.NET Upload options. There choose the hidden fields you just created in the **Save the Image Properties in Hidden Form Fields** section.

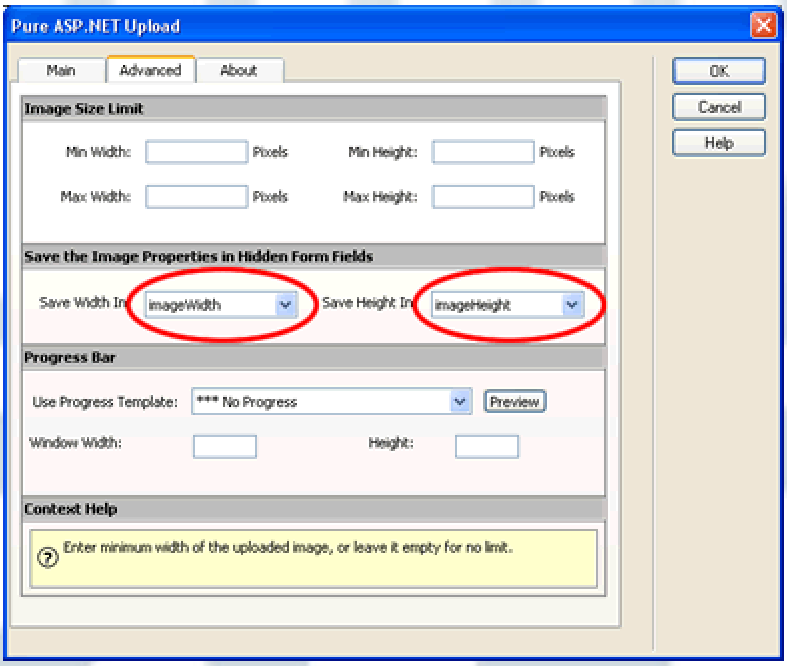


Image Size Limit

Min Width: Pixels Min Height: Pixels

Max Width: Pixels Max Height: Pixels

Save the Image Properties in Hidden Form Fields

Save Width In: imageWidth Save Height In: imageHeight

Progress Bar

Use Progress Template: *** No Progress

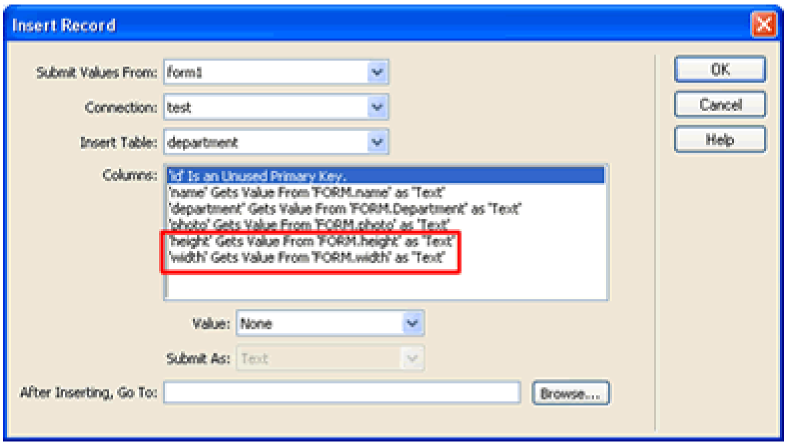
Window Width: Height:

Context Help

Enter minimum width of the uploaded image, or leave it empty for no limit.

5. **Reapply the Update record**

After you apply the Pure ASP.NET Upload extension, it will save the image size in the hidden fields. Now we need to apply the **Insert Record** server behavior (or Update Record if you are using an updating a page) to also save the hidden fields.



Submit Values From: form1

Connection: test

Insert Table: department

Columns:
'id' Is an Unused Primary Key.
'name' Gets Value From 'FORM.name' as 'Text'
'department' Gets Value From 'FORM.Department' as 'Text'
'photo' Gets Value From 'FORM.photo' as 'Text'
'height' Gets Value From 'FORM.height' as 'Text'
'width' Gets Value From 'FORM.width' as 'Text'

Value: None

Submit As: Text

After Inserting, Go To:

You should assign the hidden fields to the fields in your database.

6. **Save and done**

   Save your page then, upload it to your server, and you are done! Go to your website and test it - you should be able to upload images and their size will be saved in the database!

   NOTE: This method will work only if you upload a single image! It won't work if you try to upload multiple images at the same time - ie if you have multiple file upload fields in your form.

Upload with Progress Bar

Upload with Progress Bar

In this tutorial we will explain how to make a file upload with a progress bar.

How to do it

To make the file upload with progress bar, you need to:

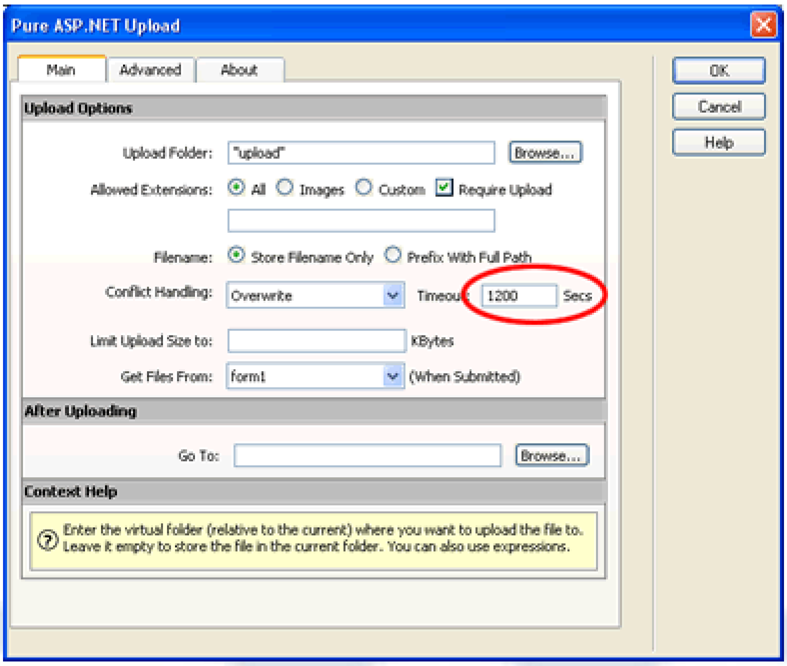
1. **Start with basic upload page**

   Start with an upload page where you have Pure ASP.NET Upload on it. If you haven't done it yet, go through the Simple file upload tutorial.

2. **Open the Pure ASP.NET Upload dialog**

   Select the Pure ASP.NET Upload dialog by choosing it from the Server Behaviors palette, DMXzone / Pure ASP.NET Upload.

   Make sure you've entered a long **timeout**. For more info see the Uploading large files tutorial.



Pure ASP.NET Upload

Upload Options

Upload Folder: "upload" Browse...

Allowed Extensions: ● All ○ Images ○ Custom ☑ Require Upload

Filename: ● Store Filename Only ○ Prefix With Full Path

Conflict Handling: Overwrite Timeout: 1200 Secs

Limit Upload Size to: KBytes

Get Files From: form1 (When Submitted)

After Uploading

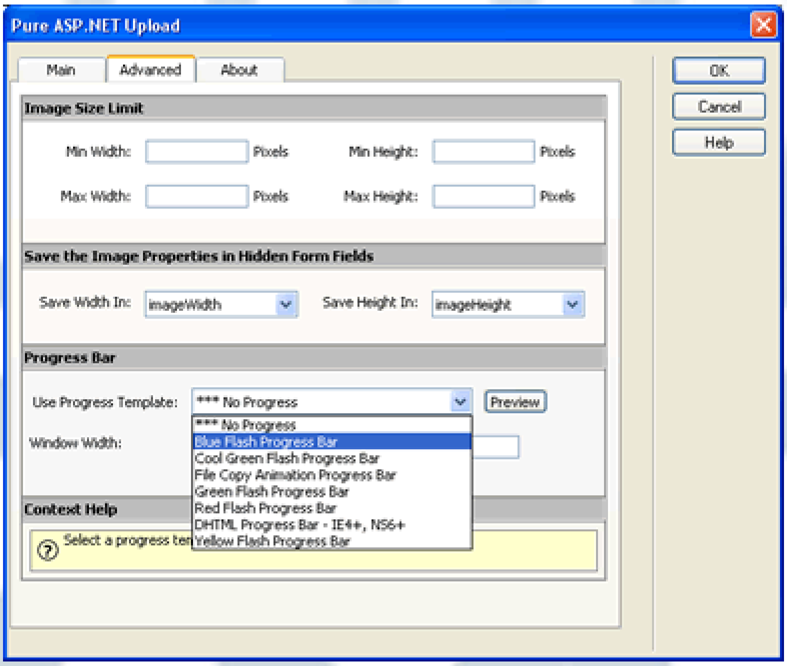
Go To: Browse...

Context Help

Enter the virtual folder (relative to the current) where you want to upload the file to. Leave it empty to store the file in the current folder. You can also use expressions.

3. **Select the desired progress bar**

When you need to restrict the size of the uploaded images, you need to fill in the minimum and/or maximum width and height on the Advanced tab. If you leave one of those values empty it means it is unlimited. The sizes are in pixels.



Pure ASP.NET Upload

Main
Advanced
About
OK
Cancel
Help

Image Size Limit

Min Width: Pixels Min Height: Pixels

Max Width: Pixels Max Height: Pixels

Save the Image Properties in Hidden Form Fields

Save Width In: imageWidth Save Height In: imageHeight

Progress Bar

Use Progress Template: *** No Progress Preview

Window Width:

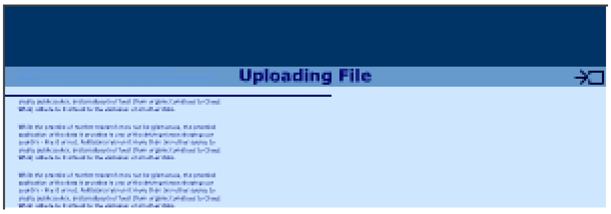
*** No Progress
Blue Flash Progress Bar
Cool Green Flash Progress Bar
File Copy Animation Progress Bar
Green Flash Progress Bar
Red Flash Progress Bar
DHTML Progress Bar - IE4+, NS6+
Yellow Flash Progress Bar

Context Help

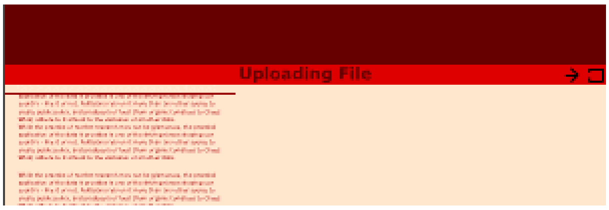
Select a progress ten

Currenly Pure ASP.NET Upload is distributed with the following progress bars:

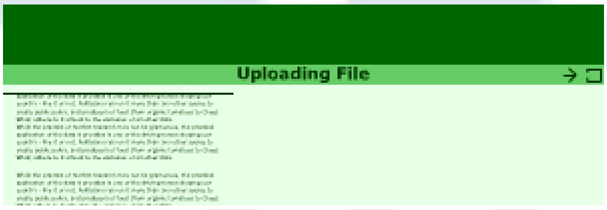
Blue Flash Progress Bar



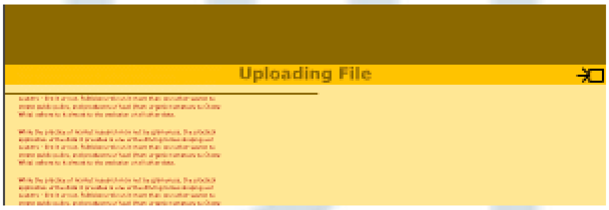
Red Flash Progress Bar



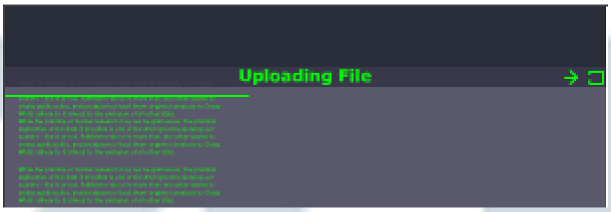
Green Flash Progress Bar



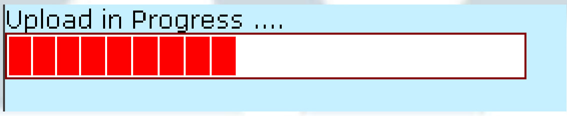
Yellow Flash Progress Bar



Cool Green Flash Progress Bar



DHTML Progress Bar


Upload in Progress ....

File Copy Progress Bar

Upload in Progress ....

You can also write your own progress bars and plug them in Pure ASP.NET Upload.

After you select a progress bar you will see that extra files are copied to your site. You have to upload those to your remote site as well.

4. **Save and done**

   Save your page then, upload it to your server, and you are done! Go to your website and test it - you should be able to upload files and a nice progress bar will appear.

Making your own Progress Bars

Making your own Progress Bars

In this tutorial we will explain how to make your own progress bars and use them with Pure ASP.NET Upload.

How to do it

To make your own progress bar, you need to:

1. **Build the progress file**

   First you should make an html file that is to be shown as a progress bar. It is up to you what animations or visual progress you will put on it. You can use images, dhtml, flash - whatever you want.

2. **Required javascript functions**

   In your html file you do need to have some very simple javascript functions. Those will be used by Pure ASP.NET Upload to read more info about your progress bar.

   You will need the following three functions:

   getProgressWindowSize

   > returns the size of the popup window where the progress bar is displayed. Enter a string with **width** and **height**.
   >
   > **Sample:**

```
function getProgressWindowSize() {
  return "300,100";
}
```

getProgressWindowTitle

returns the title of the progress bar. To be used in the Pure ASP.NET Upload choice list.

Sample:

```
function getProgressWindowTitle() {
  return "Blue Flash Progress Bar";
}
```

getProgressWindowFiles

returns a comma separated list of all files needed for the upload to work. This includes the current html file.

Sample:

```
function getProgressWindowFiles() {
  return "blueFlashProgress.htm,blue_progress.swf";
}
```

3. **Save in the special folder**

All files of your progress bar should be saved in the following folder:

\Dreamweaver UltraDev 4\Configuration\Shared\DMXzone\Pure ASP.NET Upload\Progress Bars

4. **Done**

Your progress bar is ready now. When you open the Pure ASP.NET Upload dialog, you will see your progress bar in the drop down list with of available progress bars.