Table of contents

About Pure ASP Upload 3

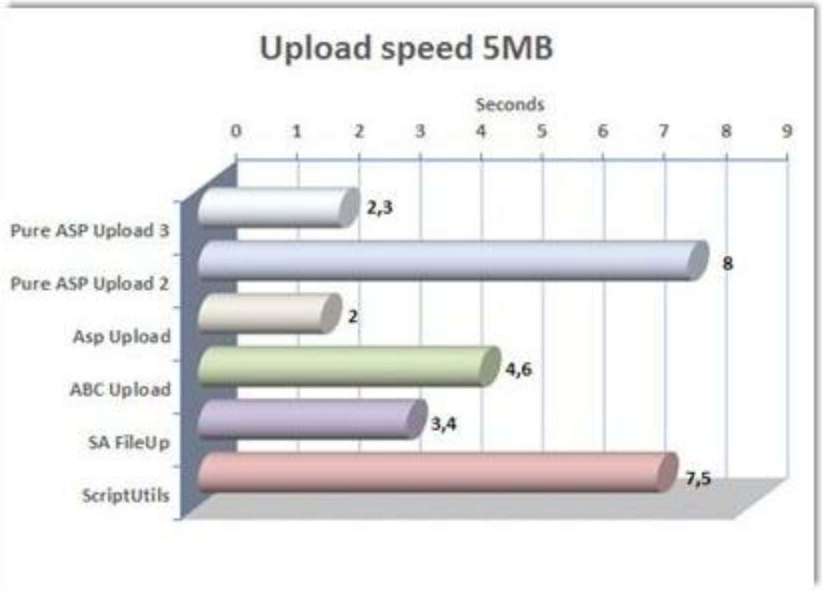
Pure ASP Upload 3 is the most **advanced upload solution** ever! Combining all the latest technologies in the web development world like **AJAX, HTML5** and **CSS3**. All available at your fingertips and in your trusted Dreamweaver environment. With just a few clicks, you become an expert.

Pure ASP Upload 3 empowers strong validated client and server side restricted uploads, making them more secure than ever. Full support for mobile file upload - now with IOS 6 and Android 4 you can upload any image from your mobile phone directly with a great looking HTML5 progress bar.

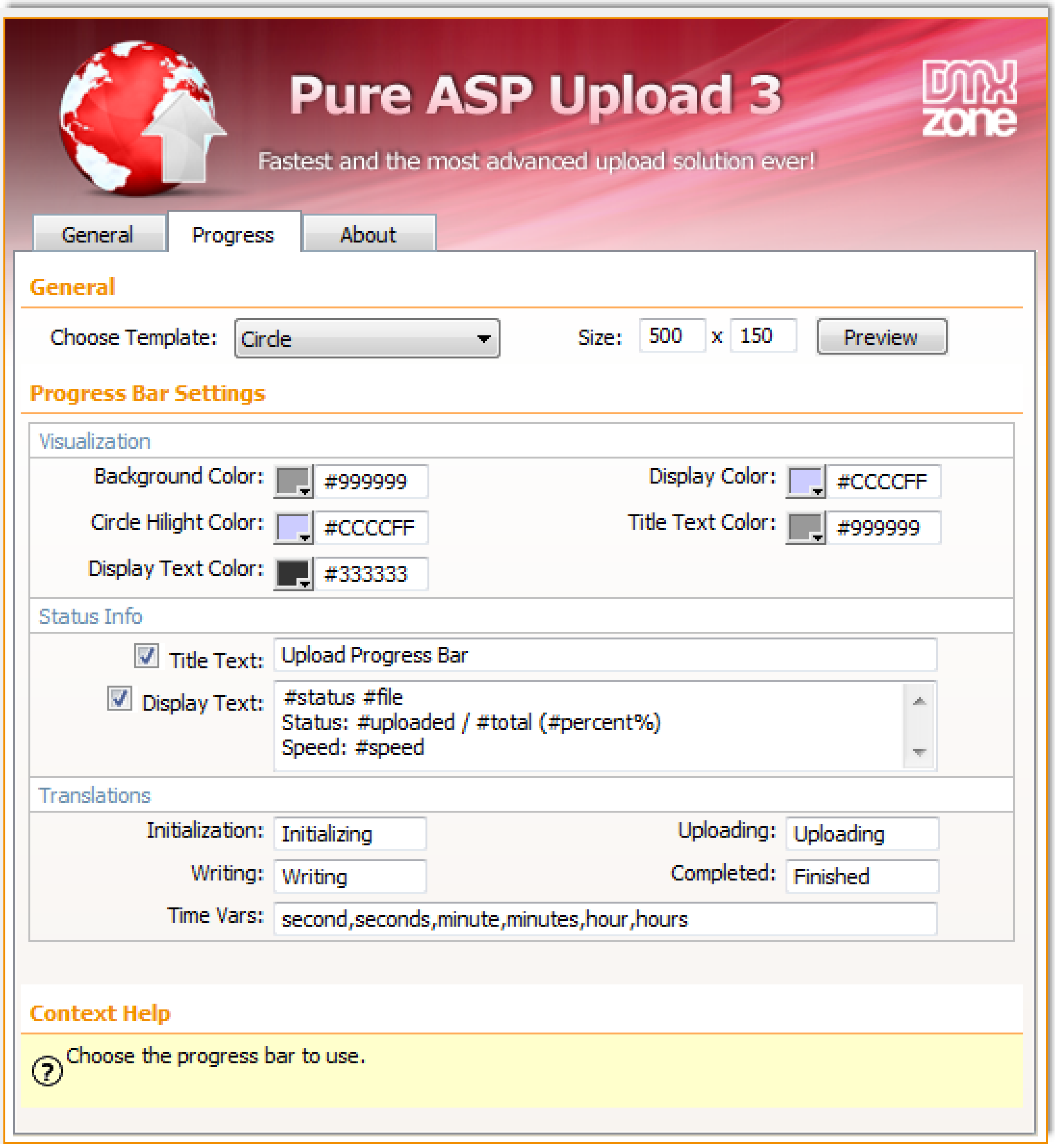
It also supports the **DMXzone Ajax Form** - turn your form to a dynamic ajax form so that it is submitted and processed directly without any page refreshes and show the result directly inline! Combine **Pure ASP Upload 3** with **DMXzone Ajax Form** and you will get full Ajax File Upload! With real time accurate progress tracking in all the latest browsers!

Features in Detail

- **Increased performance** - Pure ASP Upload 3 offers the best performance ever and is up to twenty times faster than the previous version, using three times less memory. It is even faster than many of the best upload server components on the market and still it's just pure script, so you don't need to hassle your server administrator with extra components that need to be installed. Get ready for the best upload solution.



- **Customizable progress bars** - The build-in Dreamweaver menu enables you to select multiple predefined Progress bars and customize them.



General

Choose Template: Circle   Size: 500 x 150

Progress Bar Settings

Visualization

Background Color: #999999   Display Color: #CCCCFF

Circle Hilight Color: #CCCCFF   Title Text Color: #999999

Display Text Color: #333333

Status Info

☑ Title Text: Upload Progress Bar

☑ Display Text:

```
#status #file
Status: #uploaded / #total (#percent%)
Speed: #speed
```

Translations

Initialization: Initializing   Uploading: Uploading

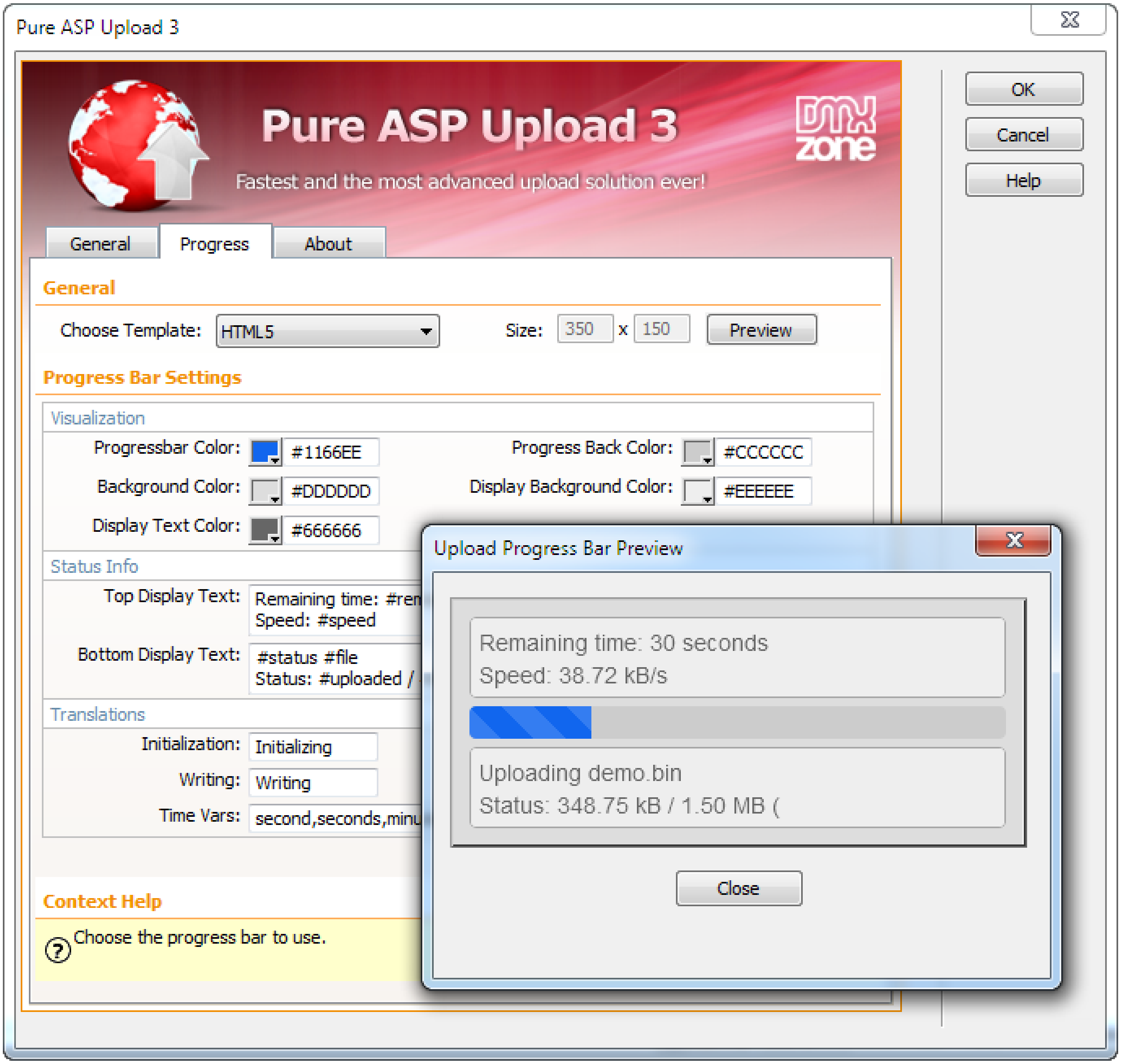
Writing: Writing   Completed: Finished

Time Vars: second,seconds,minute,minutes,hour,hours

Context Help

Choose the progress bar to use.

- **Progress bar preview -** You can preview your progress bar design directly in Dreamweaver by clicking the "preview" button.



General

Choose Template: HTML5   Size: 350 x 150

Progress Bar Settings

Visualization

Progressbar Color: #1166EE   Progress Back Color: #CCCCCC

Background Color: #DDDDDD   Display Background Color: #EEEEEE

Display Text Color: #666666

Status Info

Top Display Text: Remaining time: #ren
Speed: #speed

Bottom Display Text: #status #file
Status: #uploaded /

Translations

Initialization: Initializing

Writing: Writing

Time Vars: second,seconds,minu

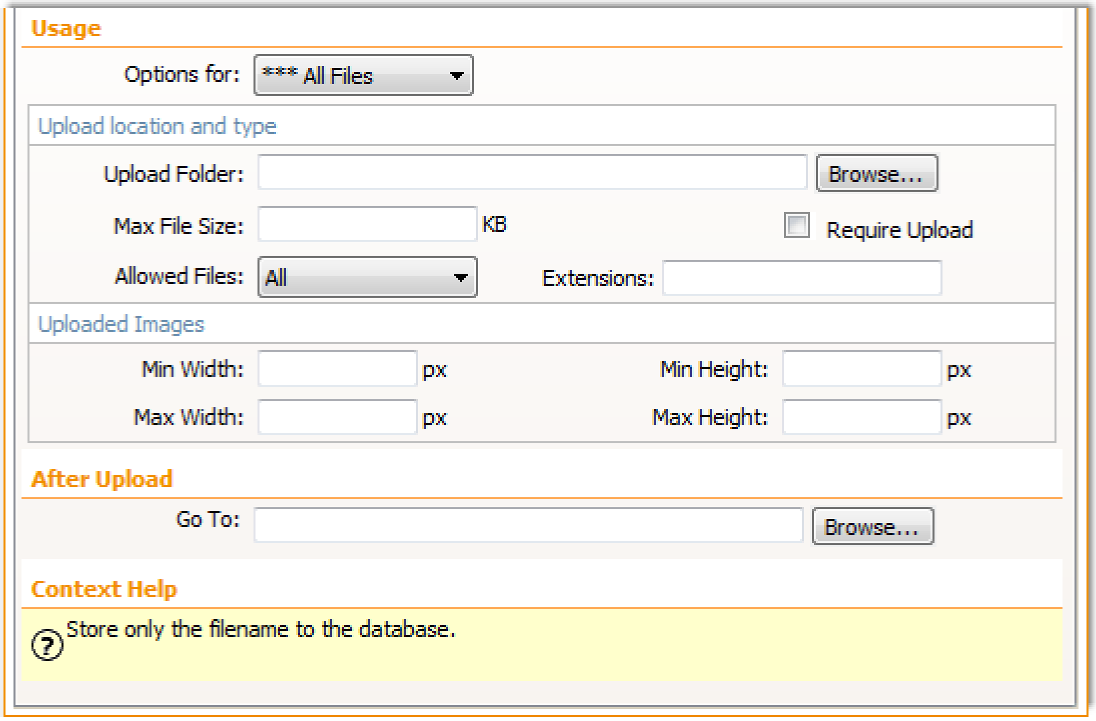
Context Help

Choose the progress bar to use.

Remaining time: 30 seconds
Speed: 38.72 kB/s

Uploading demo.bin
Status: 348.75 kB / 1.50 MB (

- **Different options for each file upload field** – You can specify different options for each file upload field. This way you can have each upload field stored in a different location and specify which file types are allowed per field.



Usage

Options for: *** All Files

Upload location and type

Upload Folder:

Max File Size: KB

□ Require Upload

Allowed Files: All

Extensions:

Uploaded Images

Min Width: px

Min Height: px

Max Width: px

Max Height: px

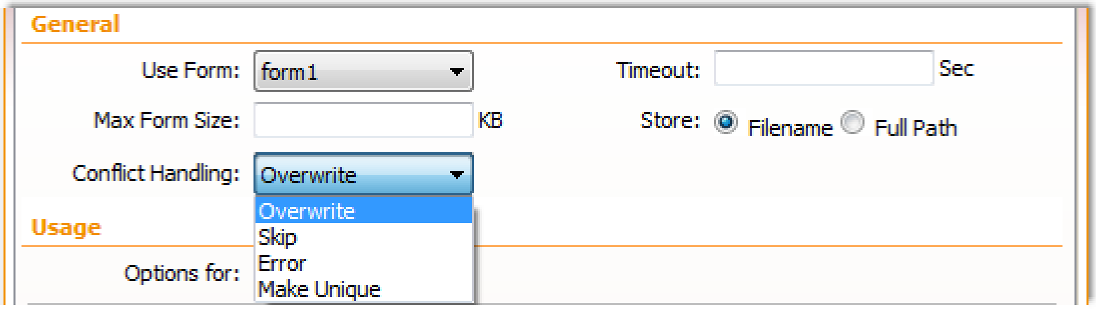
After Upload

Go To:

Context Help

Store only the filename to the database.

- **File conflict handling** – Determine how a file should be handled if its name conflicts with the name of an existing file in the directory.



General

Use Form: form1

Timeout: Sec

Max Form Size: KB

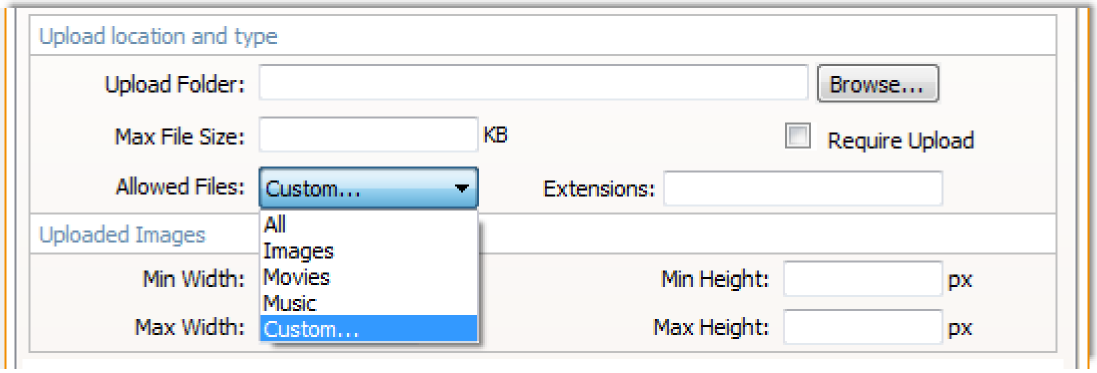
Store: ● Filename ○ Full Path

Conflict Handling: Overwrite

Usage

Options for:

- **Specify the uploaded file types** – From the dropdown menu you can select the file types that can be uploaded or enter their extensions in a comma separated list. Leave empty for no file restrictions.



Upload location and type

Upload Folder:

Max File Size: KB

□ Require Upload

Allowed Files: Custom...

Extensions:

Uploaded Images

Min Width:

Max Width:

Min Height: px

Max Height: px

- **Upload engine for Advanced HTML Editor 3** - You can use Pure ASP Upload 3 extension as an uploading solution in the **Advanced HTML Editor 3**.

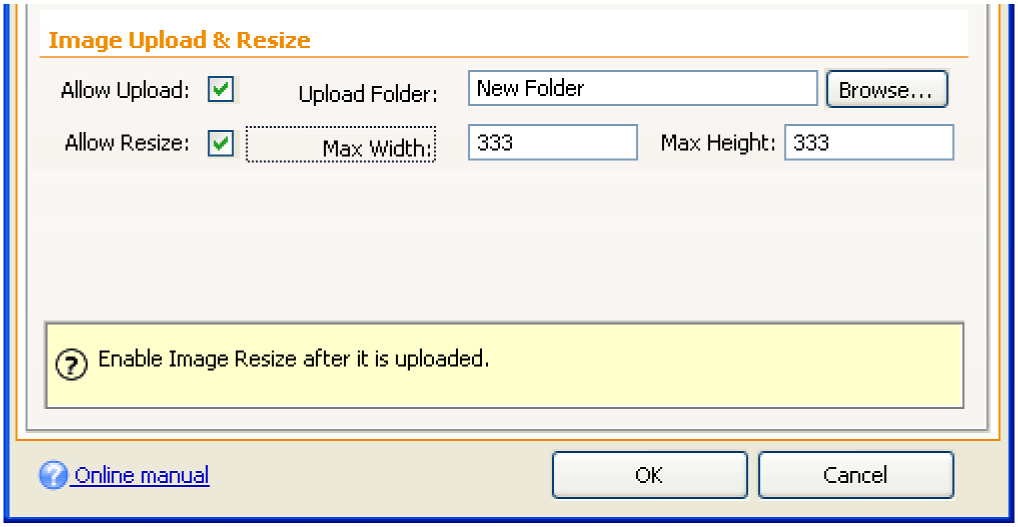


Image Upload & Resize

Allow Upload: ☑

Upload Folder: New Folder

Allow Resize: ☑

Max Width: 333

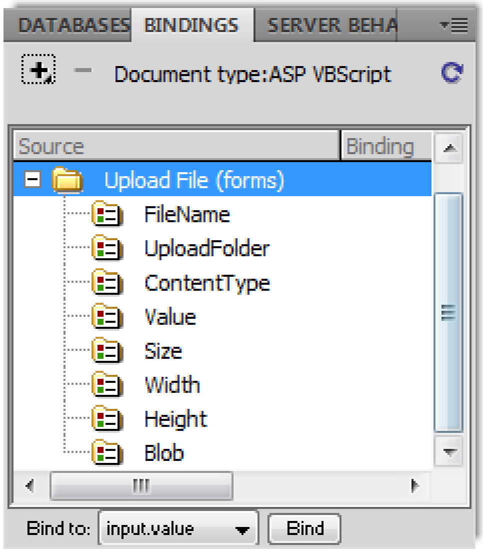
Max Height: 333

Enable Image Resize after it is uploaded.

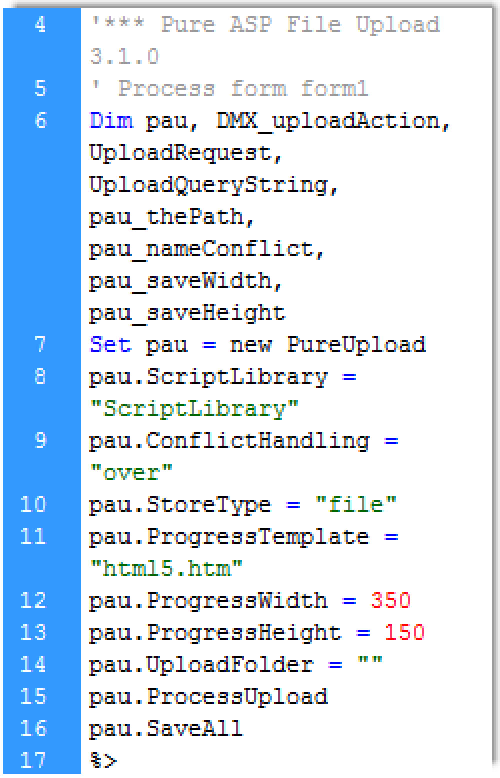
- **Advanced client and server side upload restrictions** - File size, image dimensions and type restrictions are checked client-side as well as server-side! This is the most secure way to allow uploads. Note that client side file size and image dimensions might not be allowed to be checked by all browsers due to security settings, but they will be checked server side anyway.

- **Multiple language support** - Pure ASP Upload 3 contains advanced browser detection that determines the language of your visitor and translates the messages of the extension into their language.

- **Extended Data bindings** - extended data bindings are available so that you can easily access the properties of the uploaded files and have total freedom to use them any way you intent to. Pure ASP Upload 3 allows you to set the properties of your form fields in the bindings menu.

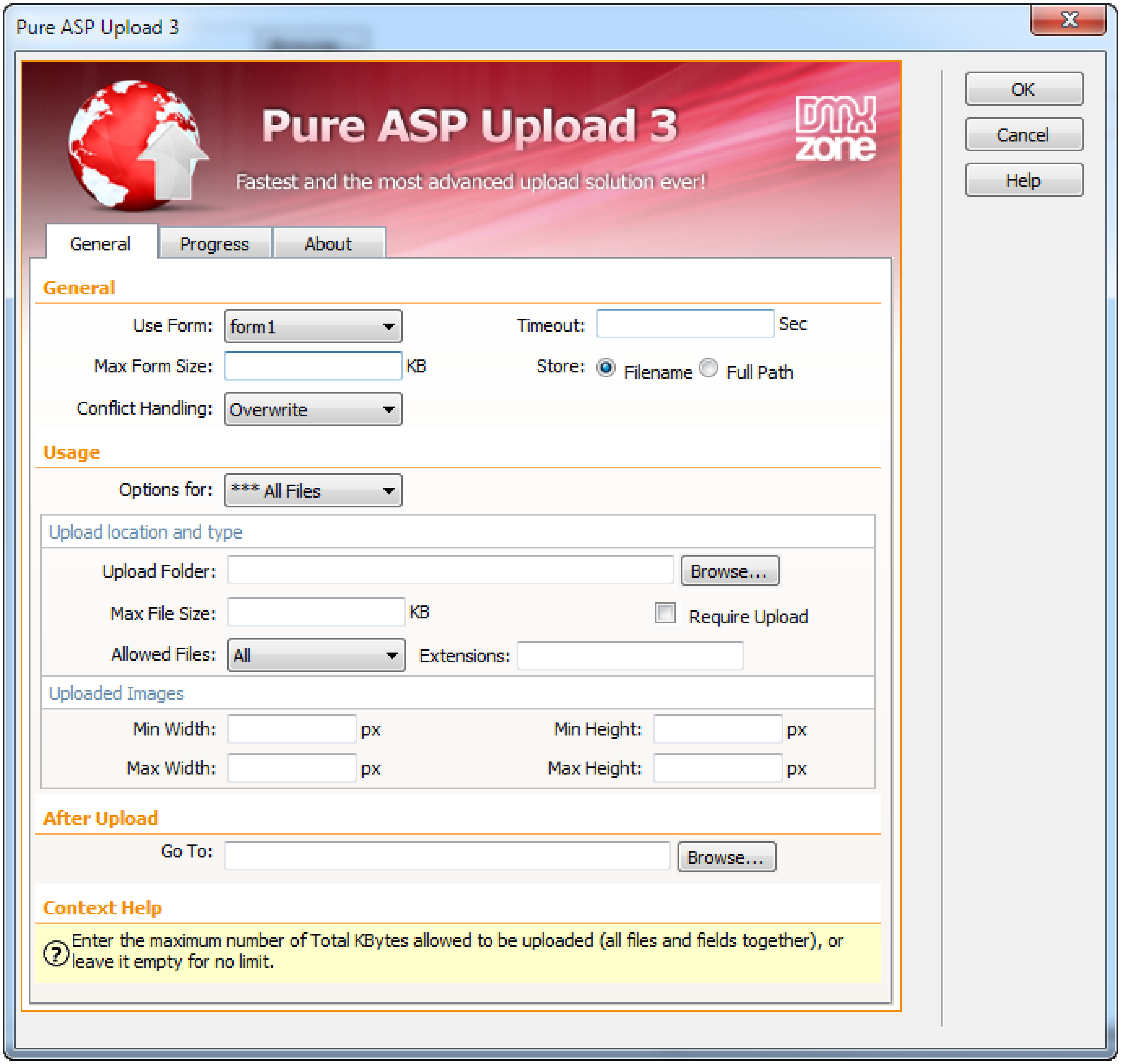


- **Fully object oriented** - Pure ASP Upload 3 is fully object oriented, making the extension more advanced than ever. It collects all the information of the files that you upload server-side. The extension collects data such as filenames, file sizes and the size of images from the upload class so you can store these values in your database.



```
'*** Pure ASP File Upload 3.1.0
' Process form form1
Dim pau, DMX_uploadAction, UploadRequest, UploadQueryString, pau_thePath, pau_nameConflict, pau_saveWidth, pau_saveHeight
Set pau = new PureUpload
pau.ScriptLibrary = "ScriptLibrary"
pau.ConflictHandling = "over"
pau.StoreType = "file"
pau.ProgressTemplate = "html5.htm"
pau.ProgressWidth = 350
pau.ProgressHeight = 150
pau.UploadFolder = ""
pau.ProcessUpload
pau.SaveAll
%>
```

- **Great Dreamweaver Integration** - Interactive dialog in Dreamweaver with all the options you'll need.



General

Use Form: form1

Timeout: Sec

Max Form Size: KB

Store: ● Filename ○ Full Path

Conflict Handling: Overwrite

Usage

Options for: *** All Files

Upload location and type

Upload Folder: Browse...

Max File Size: KB

□ Require Upload

Allowed Files: All

Extensions:

Uploaded Images

Min Width: px

Min Height: px

Max Width: px

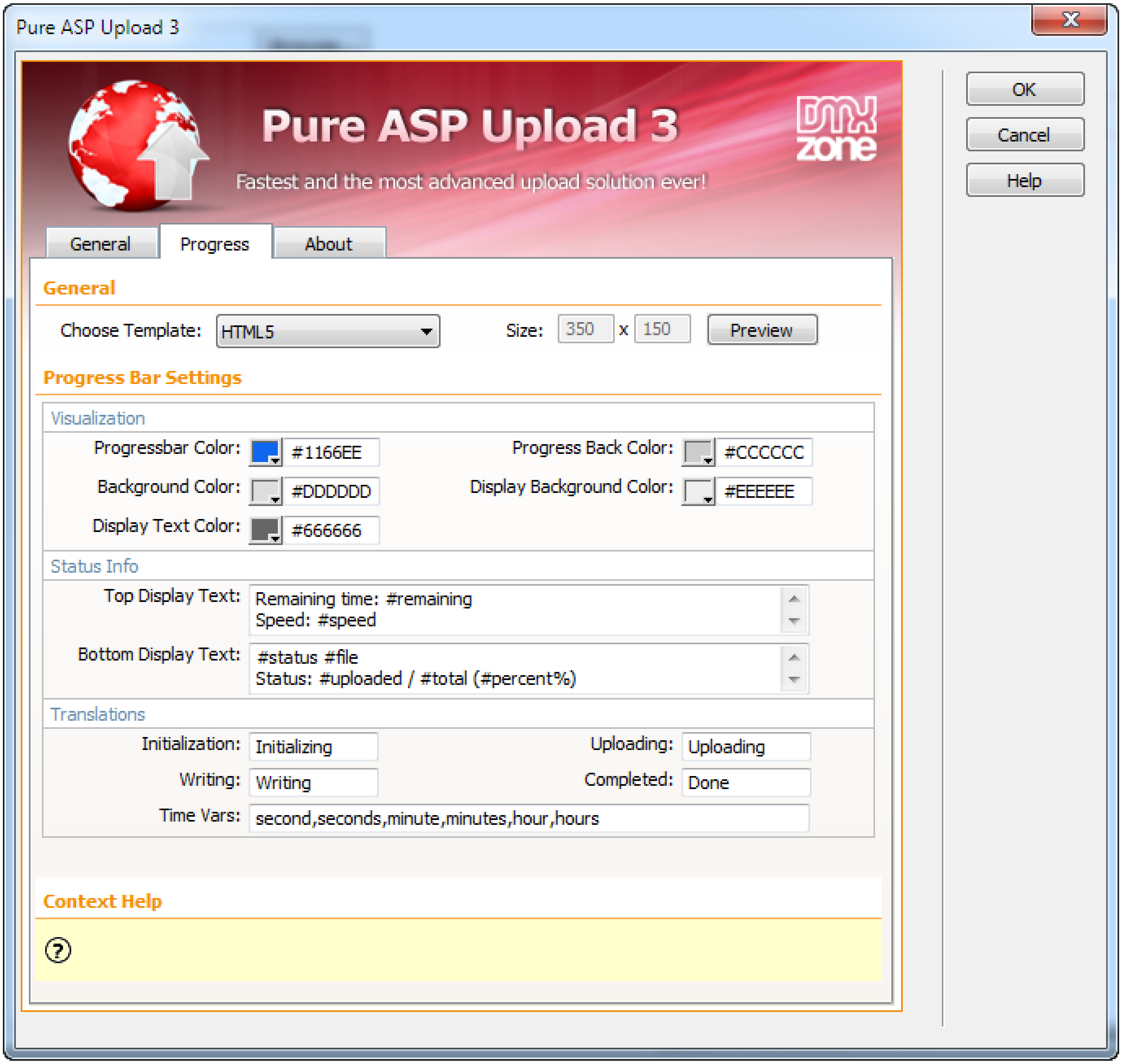
Max Height: px

After Upload

Go To: Browse...

Context Help

? Enter the maximum number of Total KBytes allowed to be uploaded (all files and fields together), or leave it empty for no limit.



General

Choose Template: HTML5

Size: 350 x 150

Progress Bar Settings

Visualization

Progressbar Color: #1166EE

Progress Back Color: #CCCCCC

Background Color: #DDDDDD

Display Background Color: #EEEEEE

Display Text Color: #666666

Status Info

Top Display Text: Remaining time: #remaining
Speed: #speed

Bottom Display Text: #status #file
Status: #uploaded / #total (#percent%)

Translations

Initialization: Initializing

Uploading: Uploading

Writing: Writing

Completed: Done

Time Vars: second,seconds,minute,minutes,hour,hours

- **Fully cross-browser compatible** - Runs perfectly on any up-to-date desktop browser as well as mobile devices.

Before you begin

Make sure to setup your site in Dreamweaver first before creating a page. If you don't know how to do that please read this.

Installing the extension

Read the tutorial at http://www.dmxzone.com/go/4671/how-to-install-extensions-from-dmxzone/

Login, download the product, and look up your serial at http://www.dmxzone.com/user/products

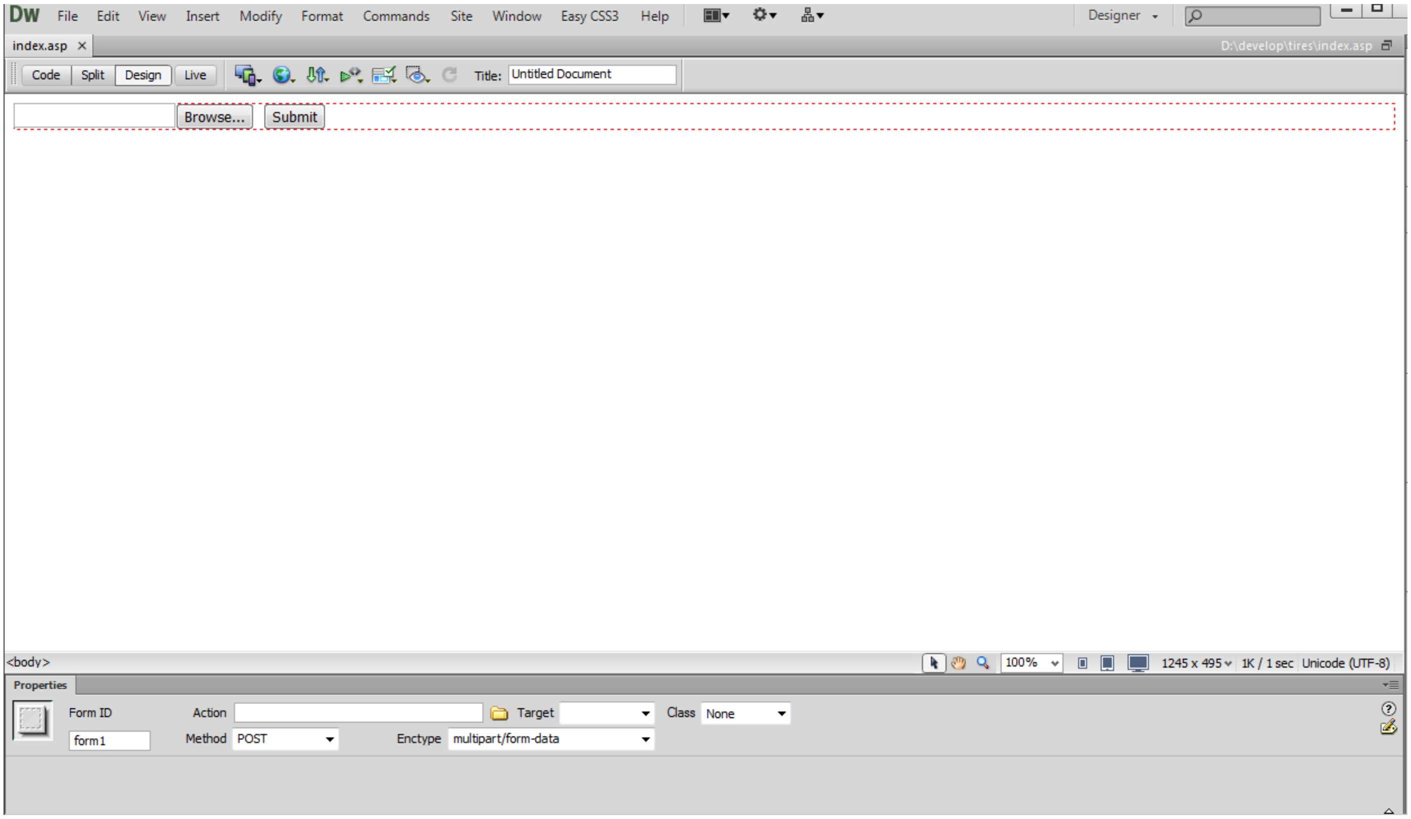
Get the latest extension manager from Adobe at http://www.adobe.com/exchange/em_download/

Open the .mxp file (the extension manager should start-up) and install the extension.

If you experience any problems contact our Live Support!

The Basics: Uploading Files with Pure ASP Upload 3

In this tutorial we will show you how to upload files using **Pure ASP Upload 3**.
*You can also check out the **video**.

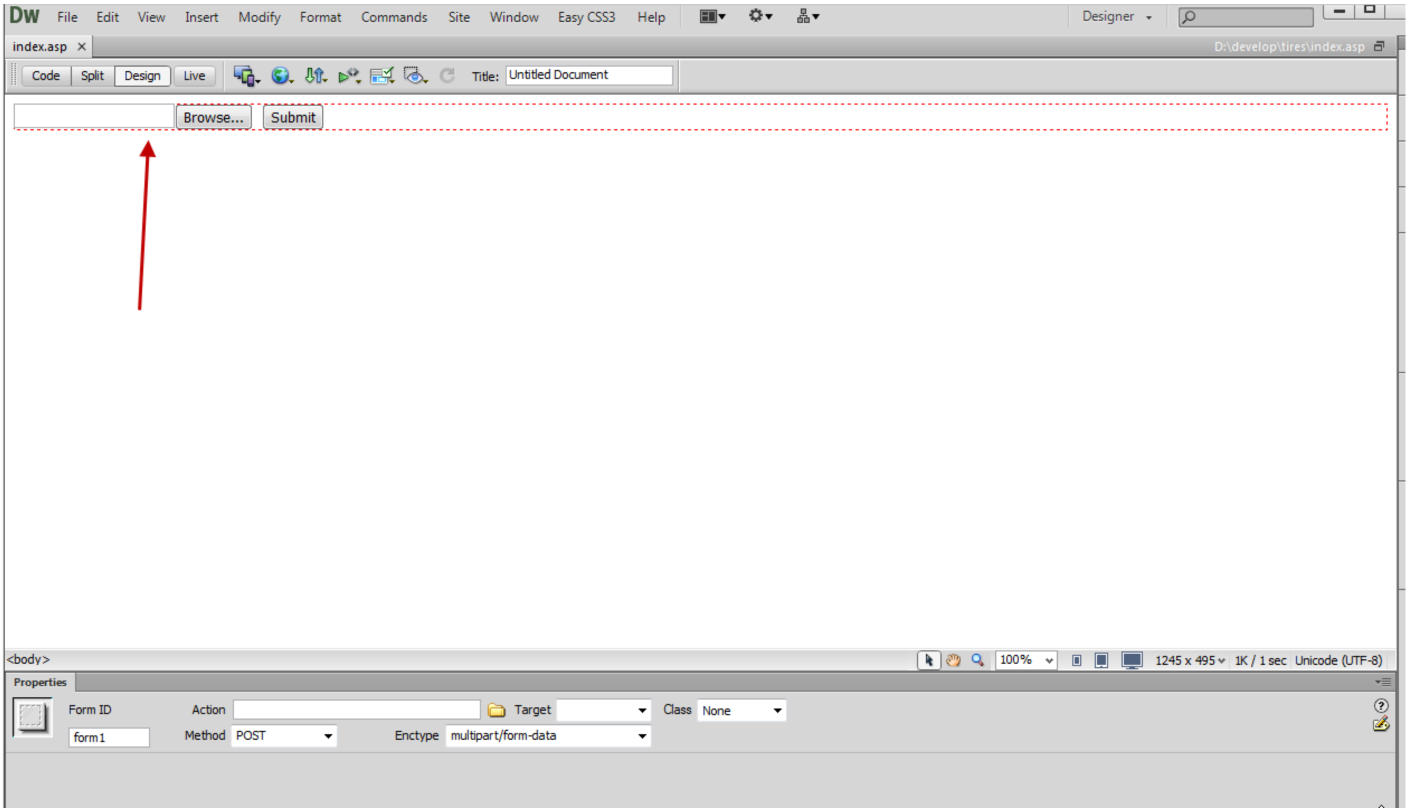


Title: Untitled Document

Form ID: form1
Action:
Target:
Class: None
Method: POST
Enctype: multipart/form-data

How to do it

1. We've already created a simple form that contains a file upload field.



Title: Untitled Document

Form ID: form1

Action:

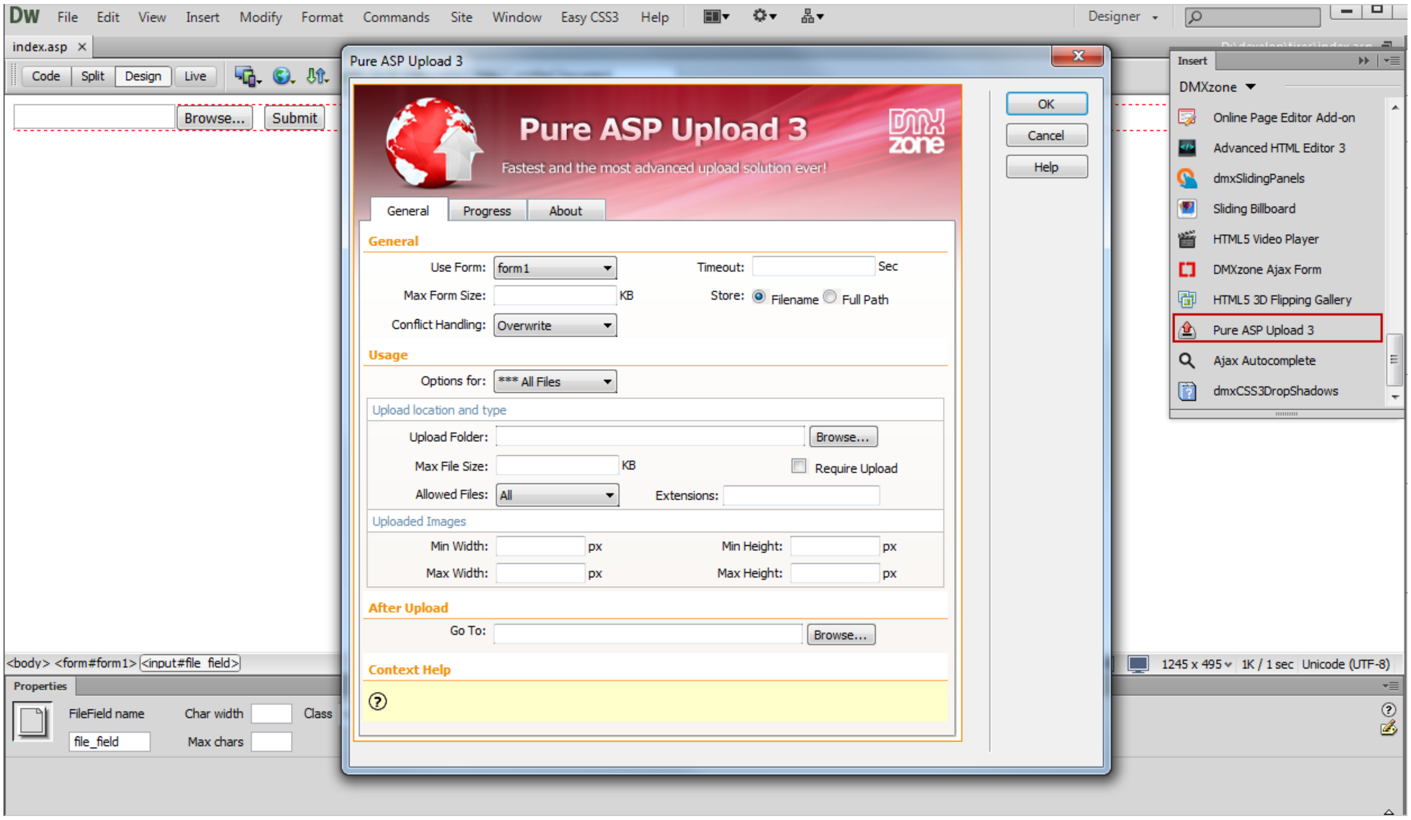
Method: POST

Target:

Enctype: multipart/form-data

Class: None

2. Click on the **Pure ASP Upload 3** icon and the window will appear.



Pure ASP Upload 3

Pure ASP Upload 3

Fastest and the most advanced upload solution ever!

General

Use Form: form1

Timeout: Sec

Max Form Size: KB

Store: ● Filename ○ Full Path

Conflict Handling: Overwrite

Usage

Options for: *** All Files

Upload location and type

Upload Folder:

Max File Size: KB

□ Require Upload

Allowed Files: All

Extensions:

Uploaded Images

Min Width: px

Min Height: px

Max Width: px

Max Height: px

After Upload

Go To:

Context Help

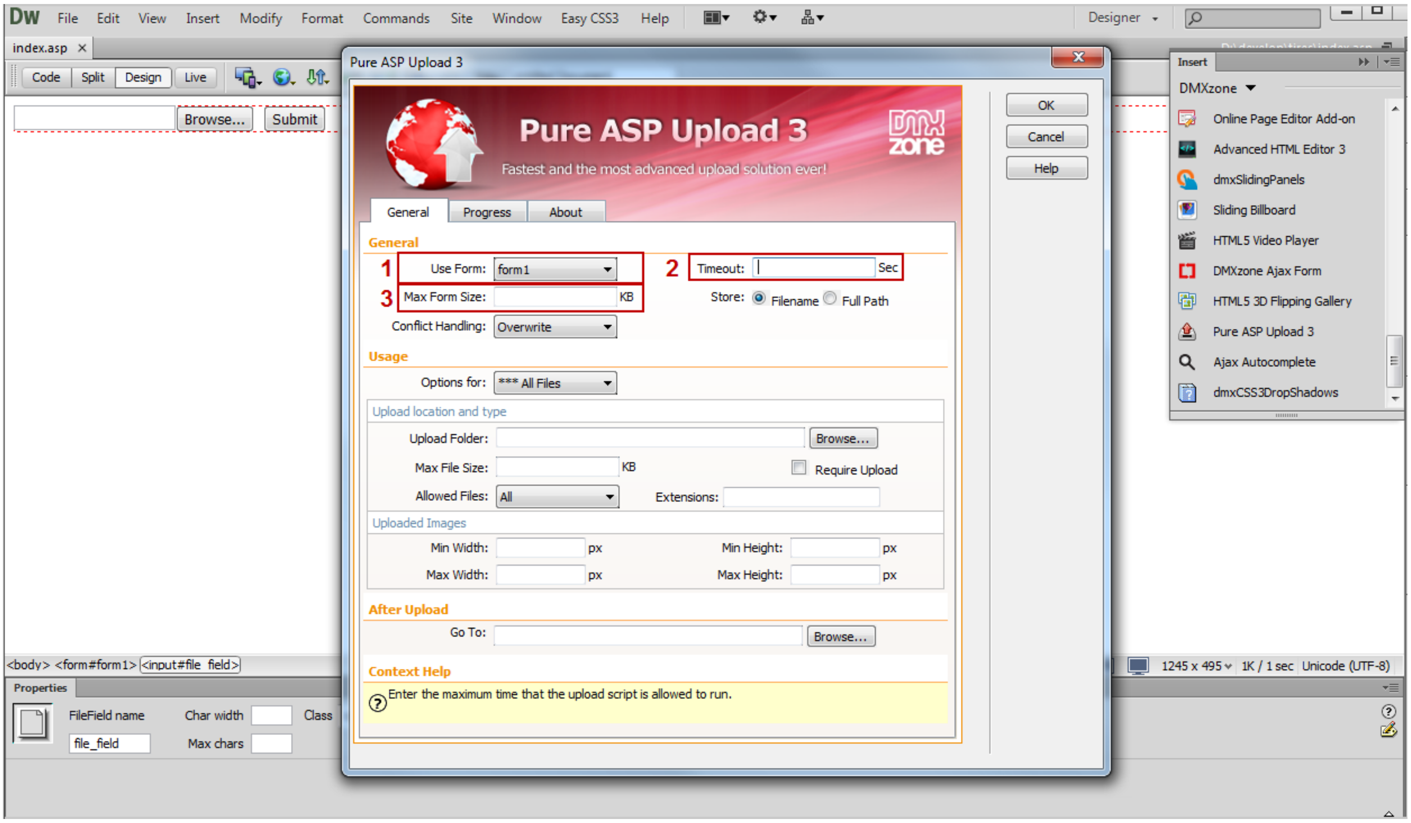
FileField name: file_field

Char width

Max chars

Class

3. In the general options **select your form (1)**. We will leave the **timeout (2)** (maximum time that the upload script is allowed to run) empty for no time limitation and the **max form size (3)**(the max form size for all the form fields allowed to be uploaded) blank for the same reason.



General

Use Form: form1

Timeout: Sec

Max Form Size: KB

Store: ● Filename ○ Full Path

Conflict Handling: Overwrite

Usage

Options for: *** All Files

Upload location and type

Upload Folder: Browse...

Max File Size: KB

□ Require Upload

Allowed Files: All

Extensions:

Uploaded Images

Min Width: px

Min Height: px

Max Width: px

Max Height: px

After Upload

Go To: Browse...

Context Help

Enter the maximum time that the upload script is allowed to run.

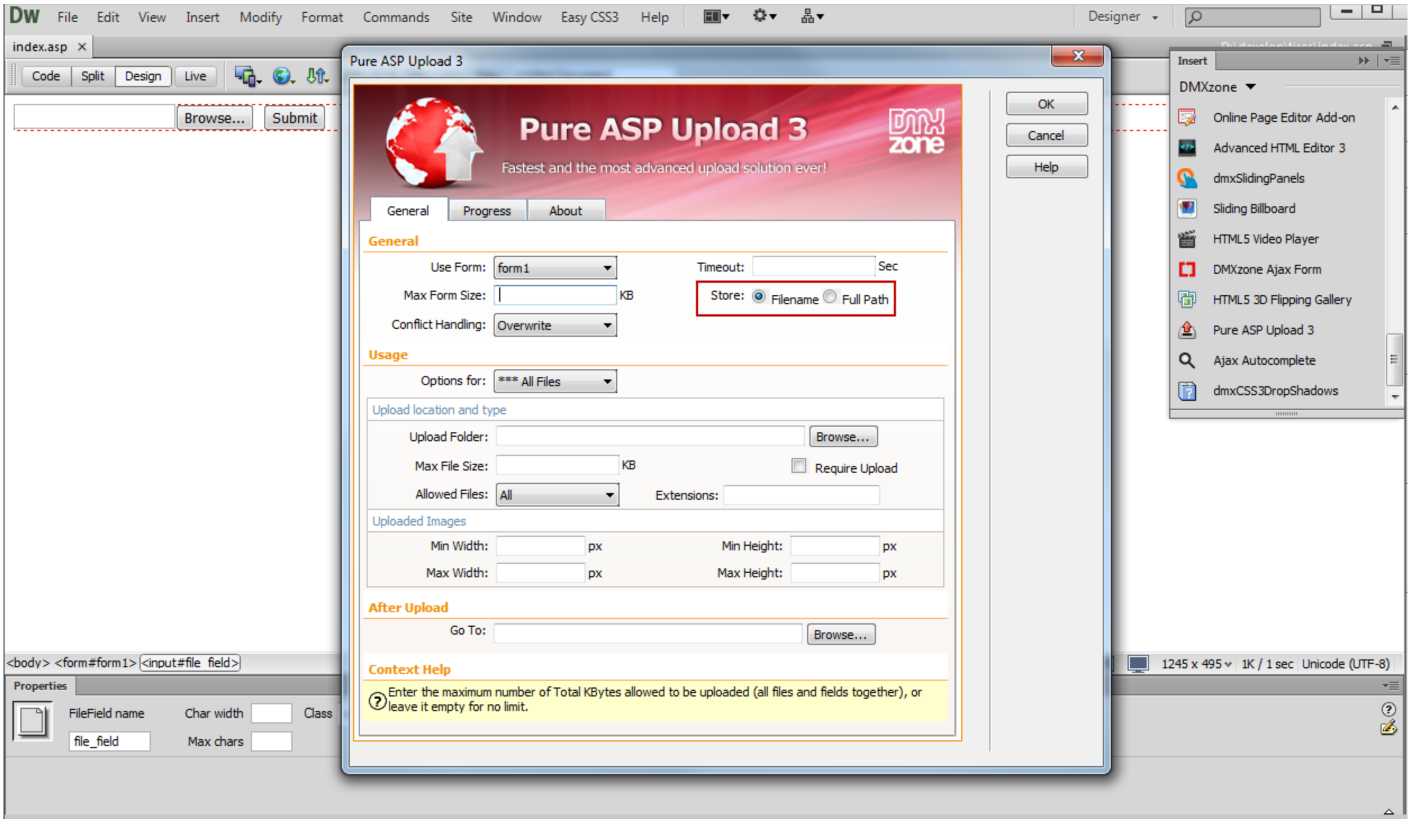
FileField name: file_field

Char width

Max chars

Class

4. Choose whether to store the **Filename** or the **Full path** if you are using a database to store the uploaded file names. In our case we choose to store the **Filenames**.



Properties: FileField name: file_field

General

- Use Form: form1
- Max Form Size: KB
- Conflict Handling: Overwrite
- Timeout: Sec
- Store: ● Filename ○ Full Path

Usage

- Options for: *** All Files

Upload location and type

- Upload Folder:
- Max File Size: KB
- □ Require Upload
- Allowed Files: All
- Extensions:

Uploaded Images

- Min Width: px
- Min Height: px
- Max Width: px
- Max Height: px

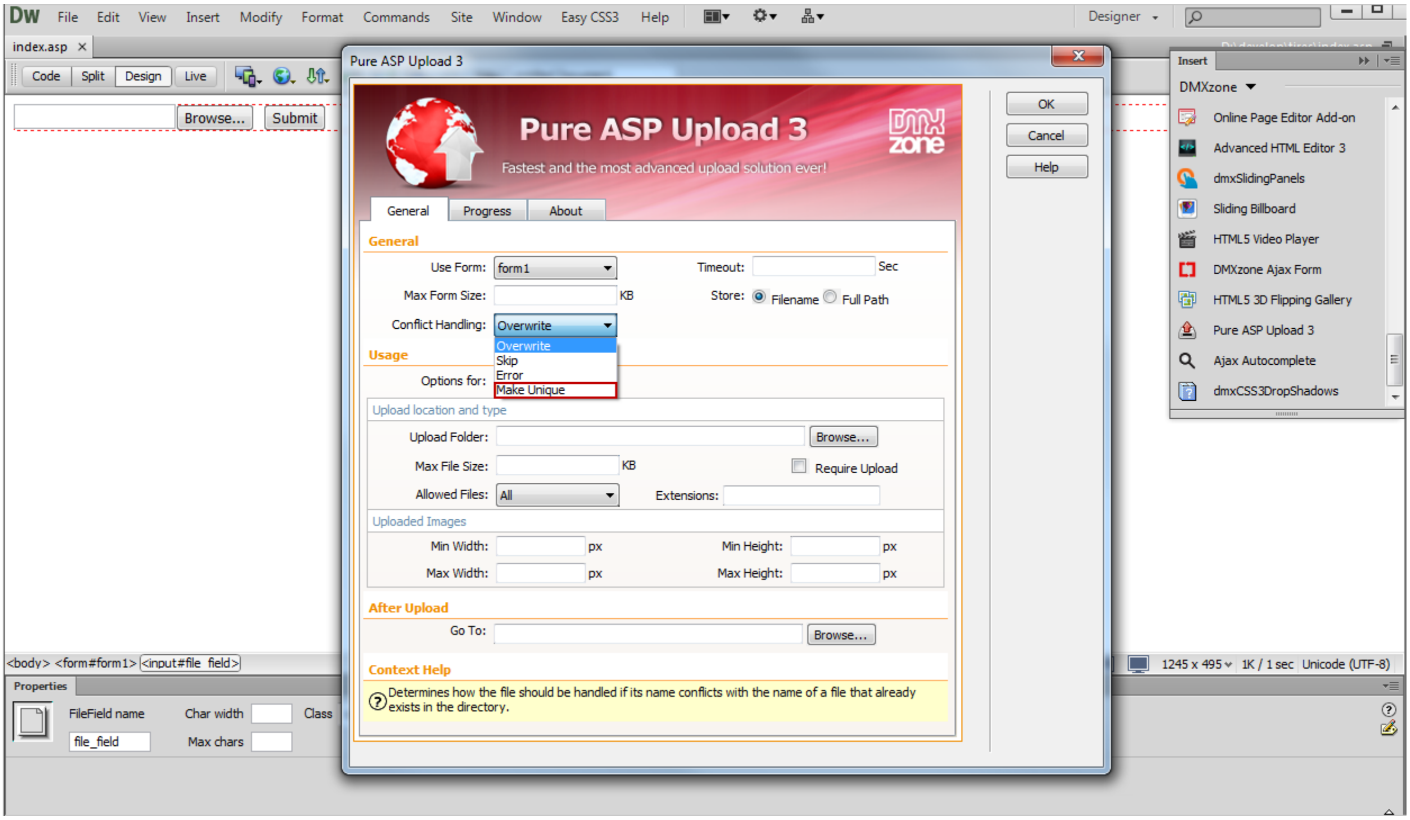
After Upload

- Go To:

Context Help

Enter the maximum number of Total KBytes allowed to be uploaded (all files and fields together), or leave it empty for no limit.

5. Select how the file should be handled if its name conflicts with a file that already exists in the upload folder. We choose the **Make unique option**.



Pure ASP Upload 3

Pure ASP Upload 3

Fastest and the most advanced upload solution ever!

General | Progress | About

General

Use Form: form1 — Timeout: Sec

Max Form Size: KB — Store: ● Filename ○ Full Path

Conflict Handling: Overwrite (Overwrite, Skip, Error, Make Unique)

Usage

Options for:

Upload location and type

Upload Folder: Browse...

Max File Size: KB — □ Require Upload

Allowed Files: All — Extensions:

Uploaded Images

Min Width: px — Min Height: px

Max Width: px — Max Height: px

After Upload

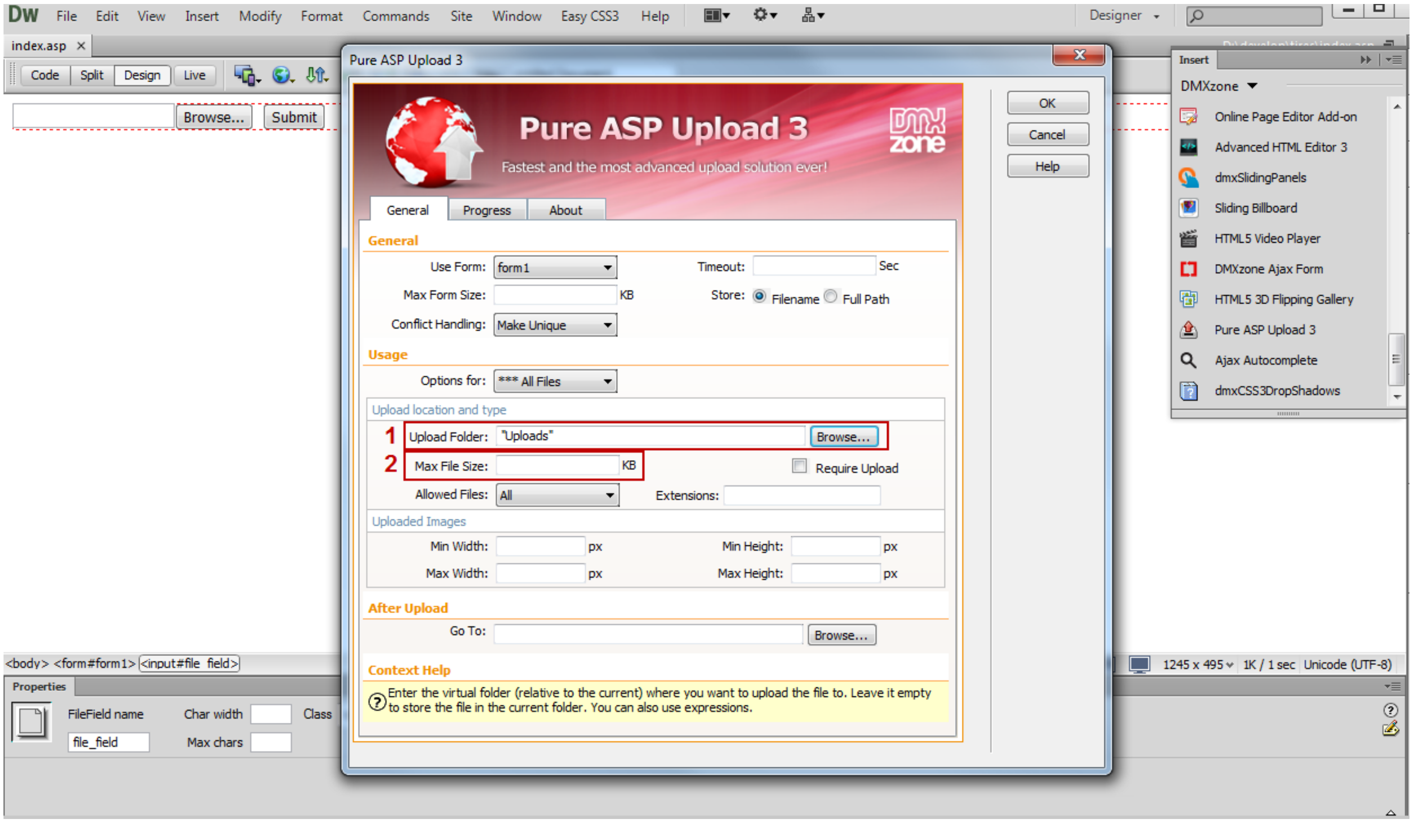
Go To: Browse...

Context Help

Determines how the file should be handled if its name conflicts with the name of a file that already exists in the directory.

FileField name: file_field — Char width — Max chars — Class

6. We choose a different folder from the **Upload folder field (1)** as we want to store the files in a different one. Also, we leave empty the **Max file size (2)** as we don't want any size limitation.



General

Use Form: form1

Timeout: Sec

Max Form Size: KB

Store: ● Filename ○ Full Path

Conflict Handling: Make Unique

Usage

Options for: *** All Files

Upload location and type

Upload Folder: "Uploads" Browse...

Max File Size: KB

□ Require Upload

Allowed Files: All

Extensions:

Uploaded Images

Min Width: px

Min Height: px

Max Width: px

Max Height: px

After Upload

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.

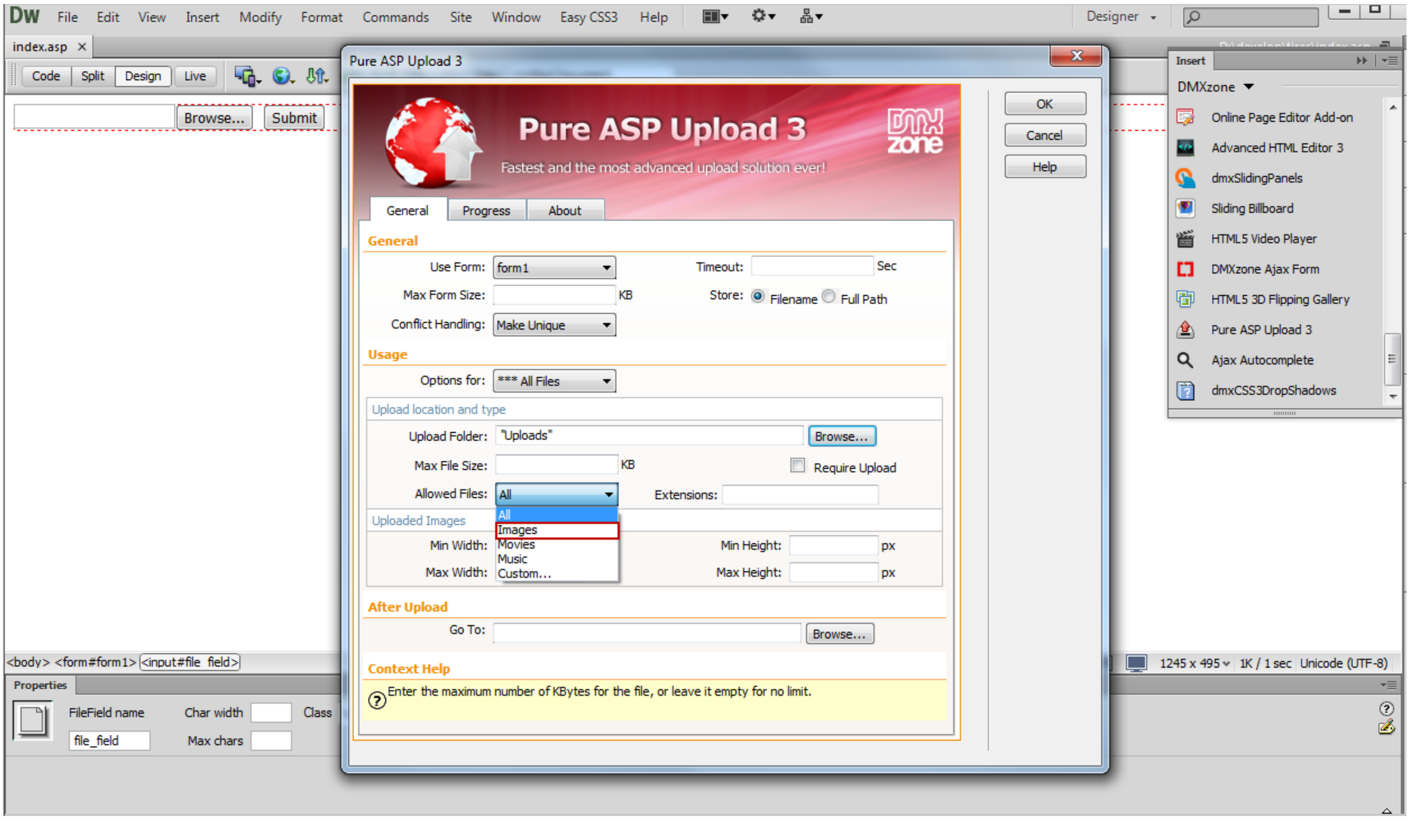
FileField name: file_field

Char width

Max chars

Class

7. For our tutorial we will limit the **uploaded file type** only to **images**, without any restrictions for the images size. The redirect URL is only needed if you are NOT using Insert/Update Record behavior.



Pure ASP Upload 3

General | Progress | About

General

Use Form: form1
Max Form Size: KB
Conflict Handling: Make Unique
Timeout: Sec
Store: ● Filename ○ Full Path

Usage

Options for: *** All Files

Upload location and type

Upload Folder: "Uploads" Browse...
Max File Size: KB
□ Require Upload
Allowed Files: All
Extensions:

All
Images
Movies
Music
Custom...

Uploaded Images

Min Width: px
Max Width: px
Min Height: px
Max Height: px

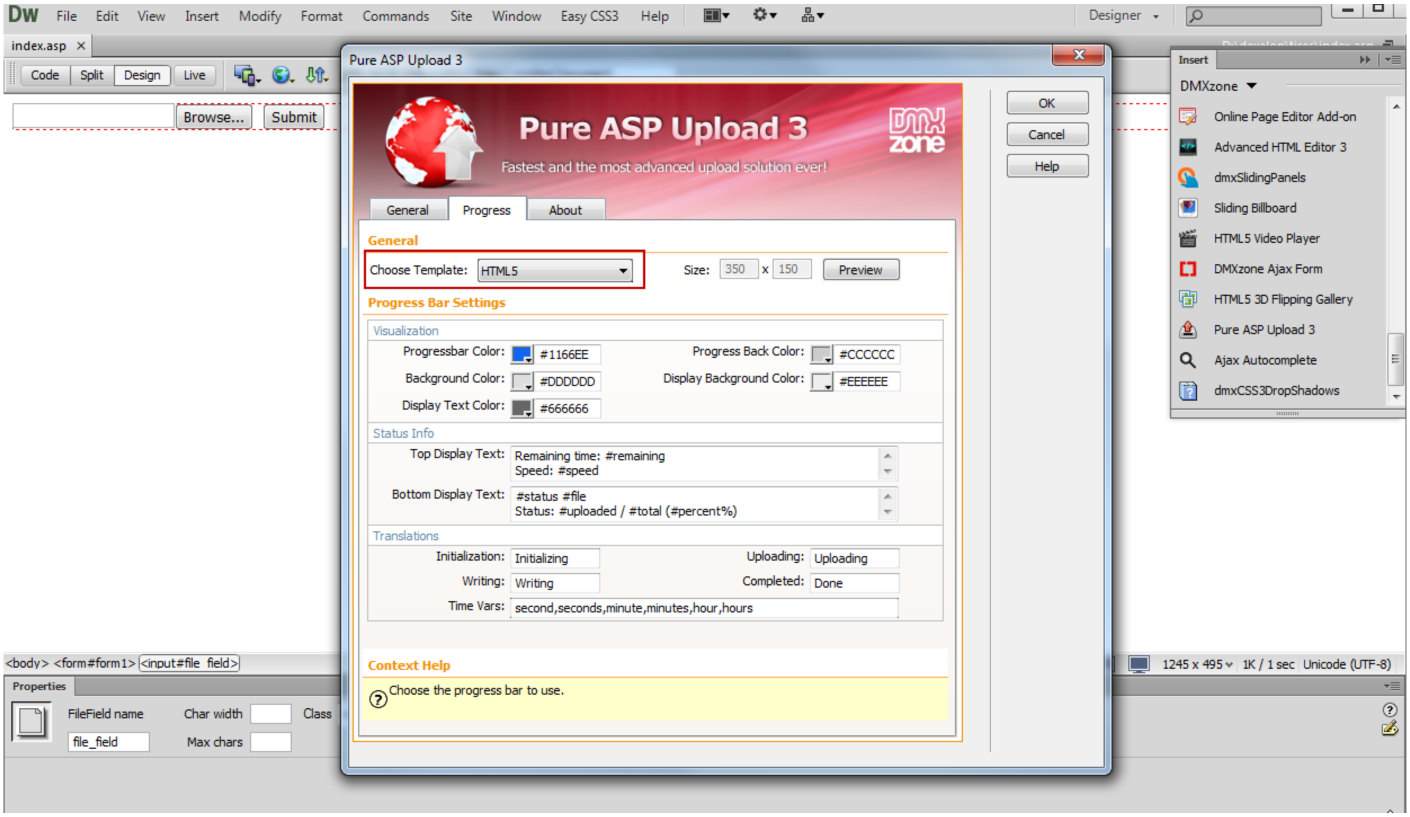
After Upload

Go To: Browse...

Context Help

Enter the maximum number of KBytes for the file, or leave it empty for no limit.

8. In the **Progress tab (1)** we will select one of the predefined progress bar designs, the **HTML5 one (2)**.



General

Choose Template: HTML5

Size: 350 x 150

Progress Bar Settings

Visualization

Progressbar Color: #1166EE
Progress Back Color: #CCCCCC
Background Color: #DDDDDD
Display Background Color: #EEEEEE
Display Text Color: #666666

Status Info

Top Display Text: Remaining time: #remaining
Speed: #speed

Bottom Display Text: #status #file
Status: #uploaded / #total (#percent%)

Translations

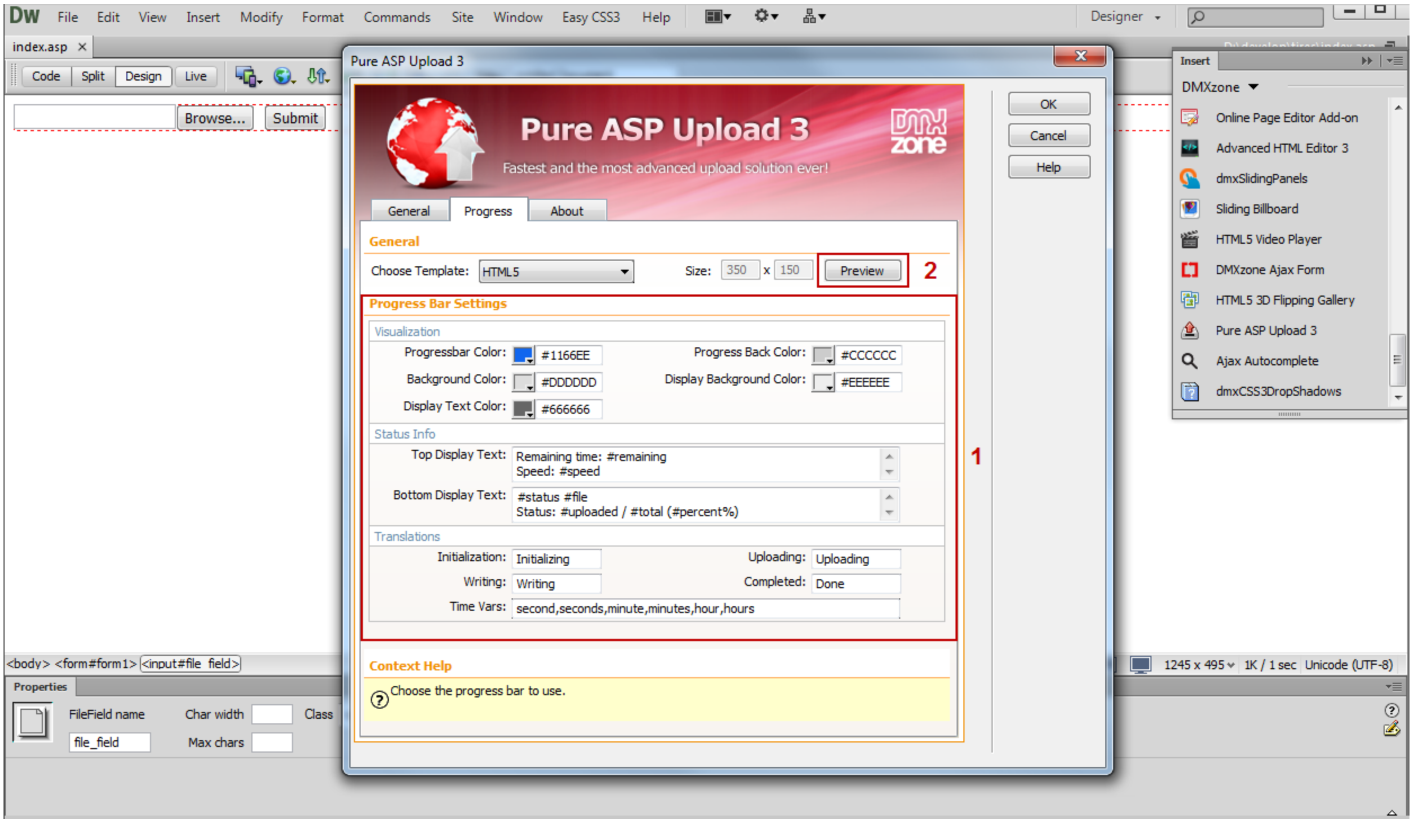
Initialization: Initializing
Uploading: Uploading
Writing: Writing
Completed: Done
Time Vars: second,seconds,minute,minutes,hour,hours

Context Help

Choose the progress bar to use.

FileField name: file_field
Char width
Class
Max chars

9. You can **customize the progress bar (1)** the way you want and **preview it directly in Dreamweaver (2)**.



General

Choose Template: HTML5 Size: 350 x 150 Preview

Progress Bar Settings

Visualization

Progressbar Color: #1166EE Progress Back Color: #CCCCCC

Background Color: #DDDDDD Display Background Color: #EEEEEE

Display Text Color: #666666

Status Info

Top Display Text: Remaining time: #remaining
Speed: #speed

Bottom Display Text: #status #file
Status: #uploaded / #total (#percent%)

Translations

Initialization: Initializing Uploading: Uploading

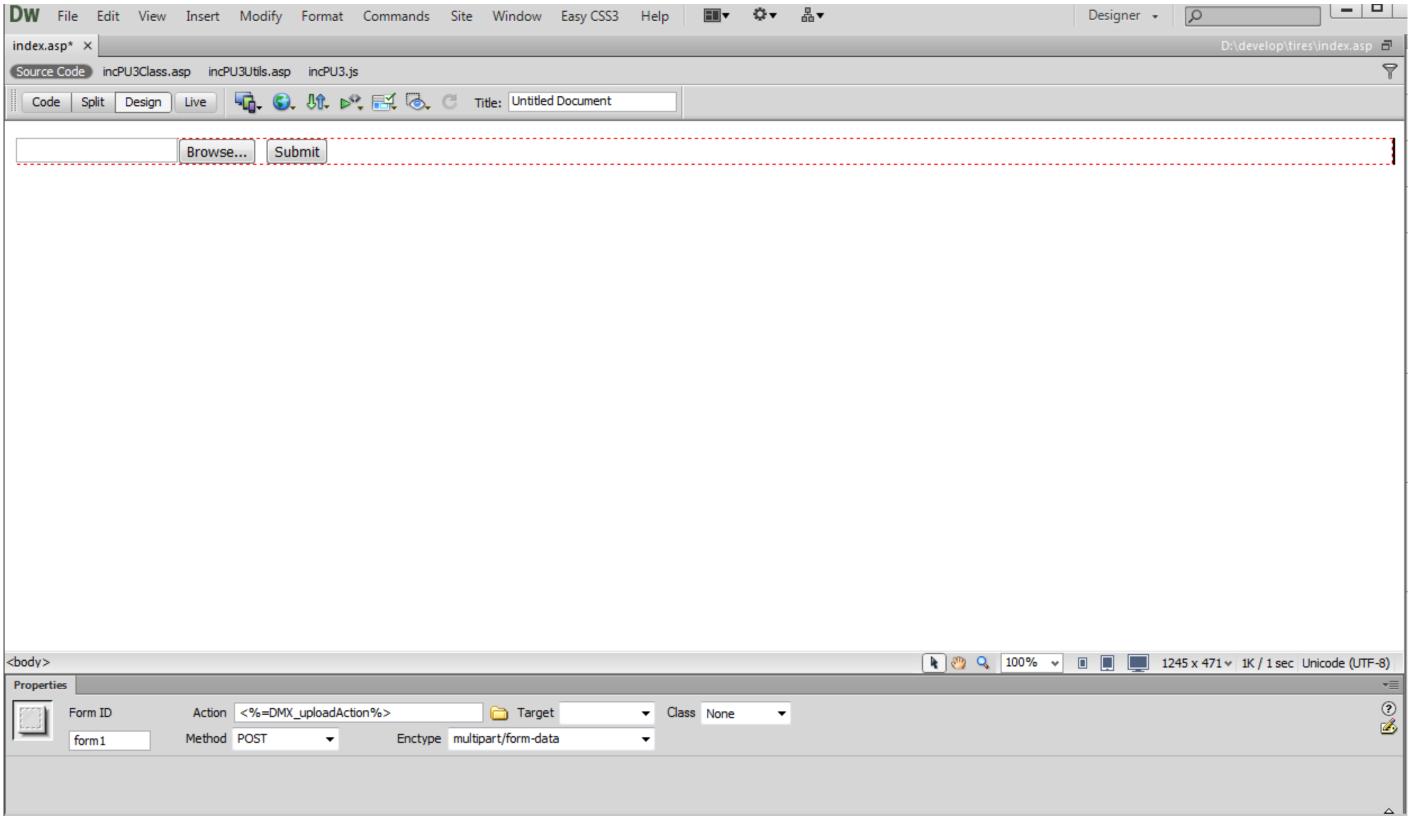
Writing: Writing Completed: Done

Time Vars: second,seconds,minute,minutes,hour,hours

Context Help

Choose the progress bar to use.

10. Now you can save your work and preview the result in the browser.



Title: Untitled Document

Form ID: form1

Action: <%=DMX_uploadAction%>

Target:

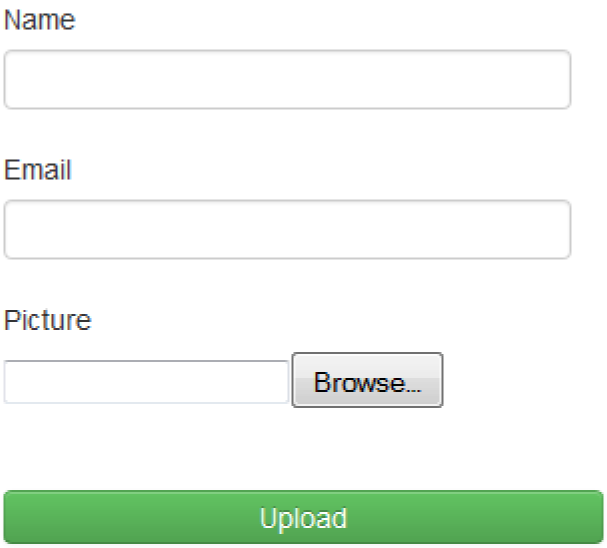
Class: None

Method: POST

Enctype: multipart/form-data

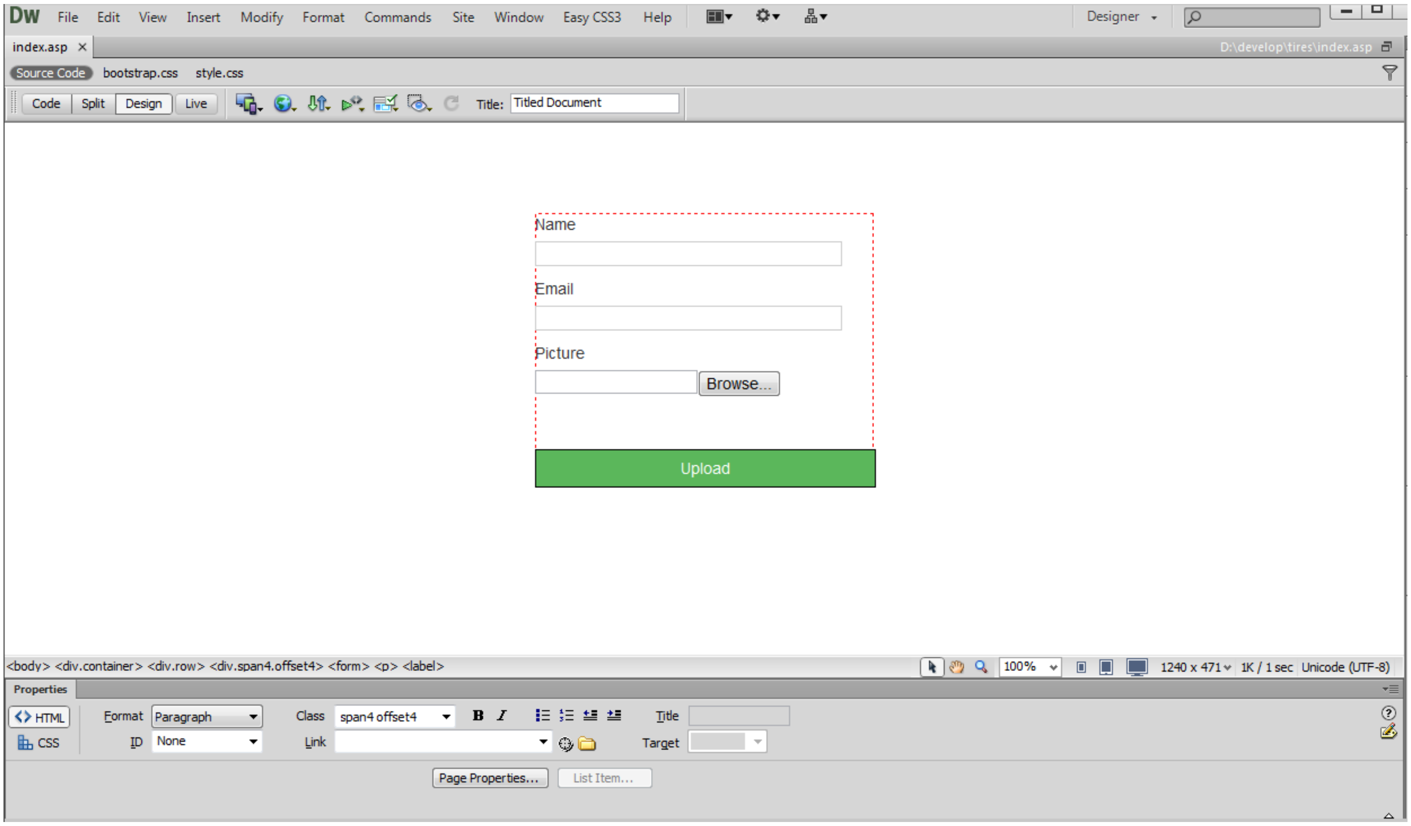
Advanced: Using Pure ASP Upload 3 with Insert Record

In this tutorial we will show you how to upload images and save their names, width and height into a database.
*You can also check out the **video**.



How to do it

1. We've already created a simple form with three fields in it.



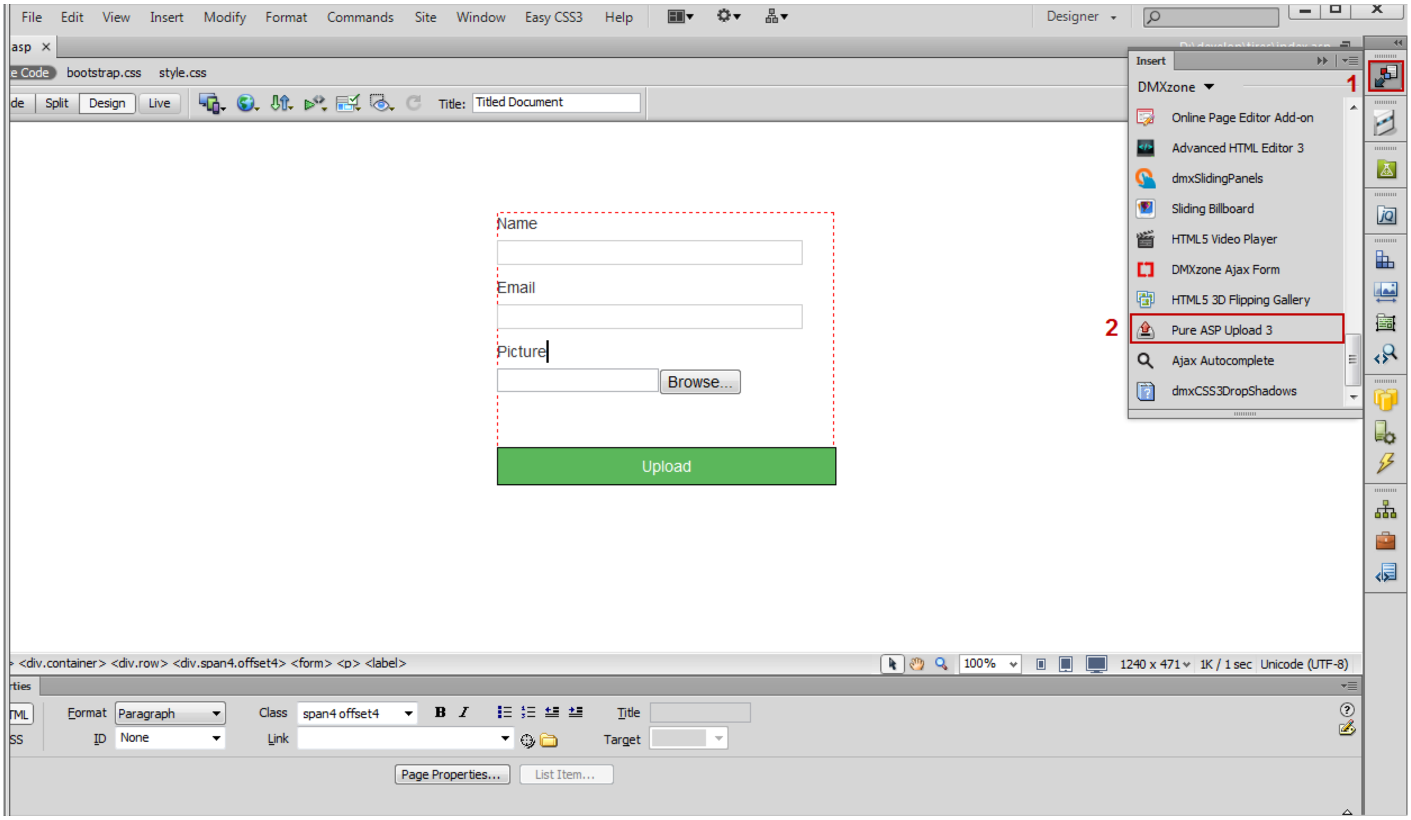
Title: Titled Document

<body> <div.container> <div.row> <div.span4.offset4> <form> <p> <label>

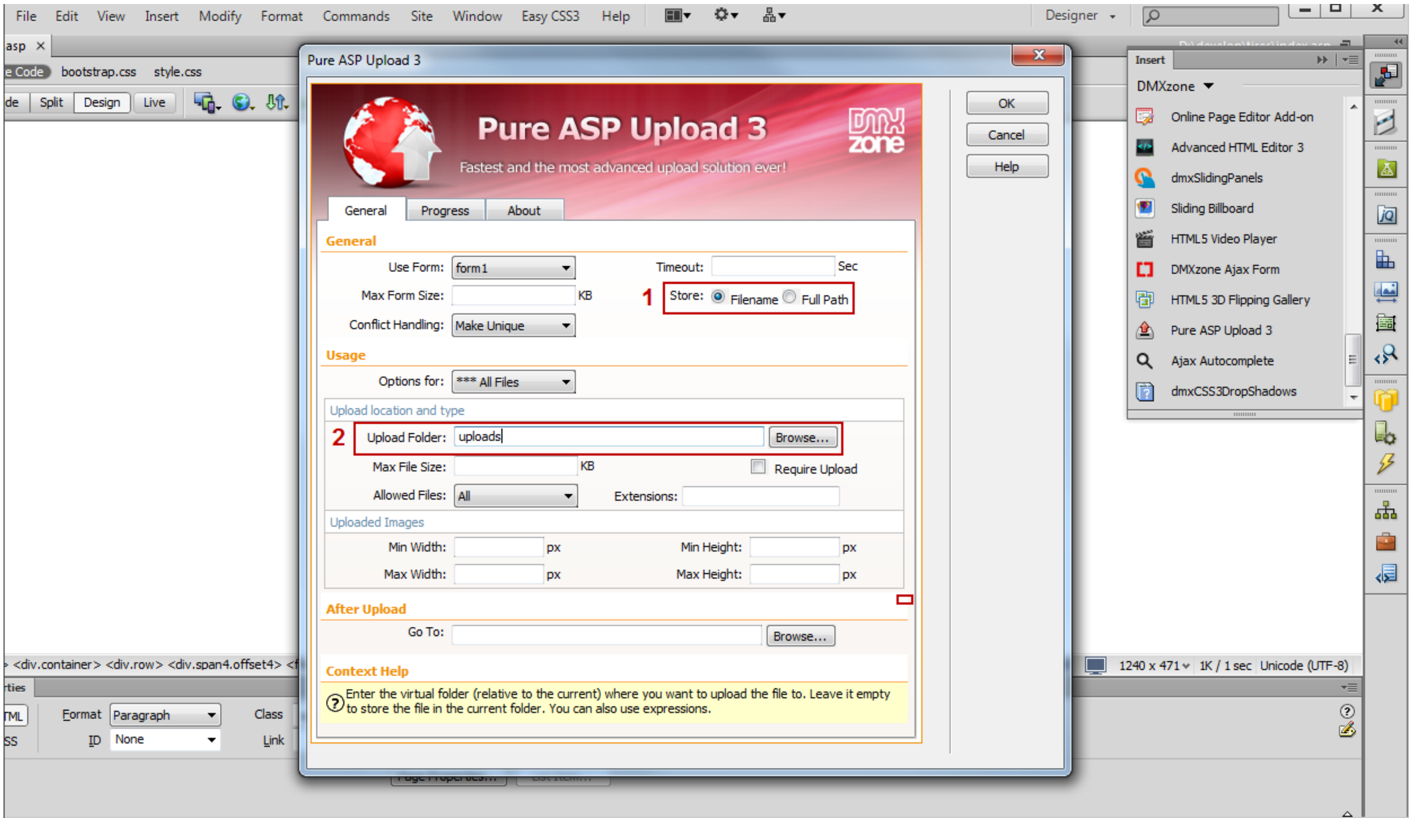
100% 1240 x 471 1K / 1 sec Unicode (UTF-8)

Format: Paragraph
ID: None
Class: span4 offset4
Link:
Title:
Target:

2. Click the **Insert button (1)** and select the **Pure ASP Upload 3 icon (2)** in order the window to appear.



3. We want to store just the **filename (1)** in our database, in a **different folder** than the current one **(2)**.



Pure ASP Upload 3

Pure ASP Upload 3
Fastest and the most advanced upload solution ever!

General | Progress | About

General

Use Form: form1
Max Form Size: KB
Conflict Handling: Make Unique
Timeout: Sec
1 Store: ● Filename ○ Full Path

Usage

Options for: *** All Files

Upload location and type

2 Upload Folder: uploads Browse...
Max File Size: KB
□ Require Upload
Allowed Files: All
Extensions:

Uploaded Images

Min Width: px
Min Height: px
Max Width: px
Max Height: px

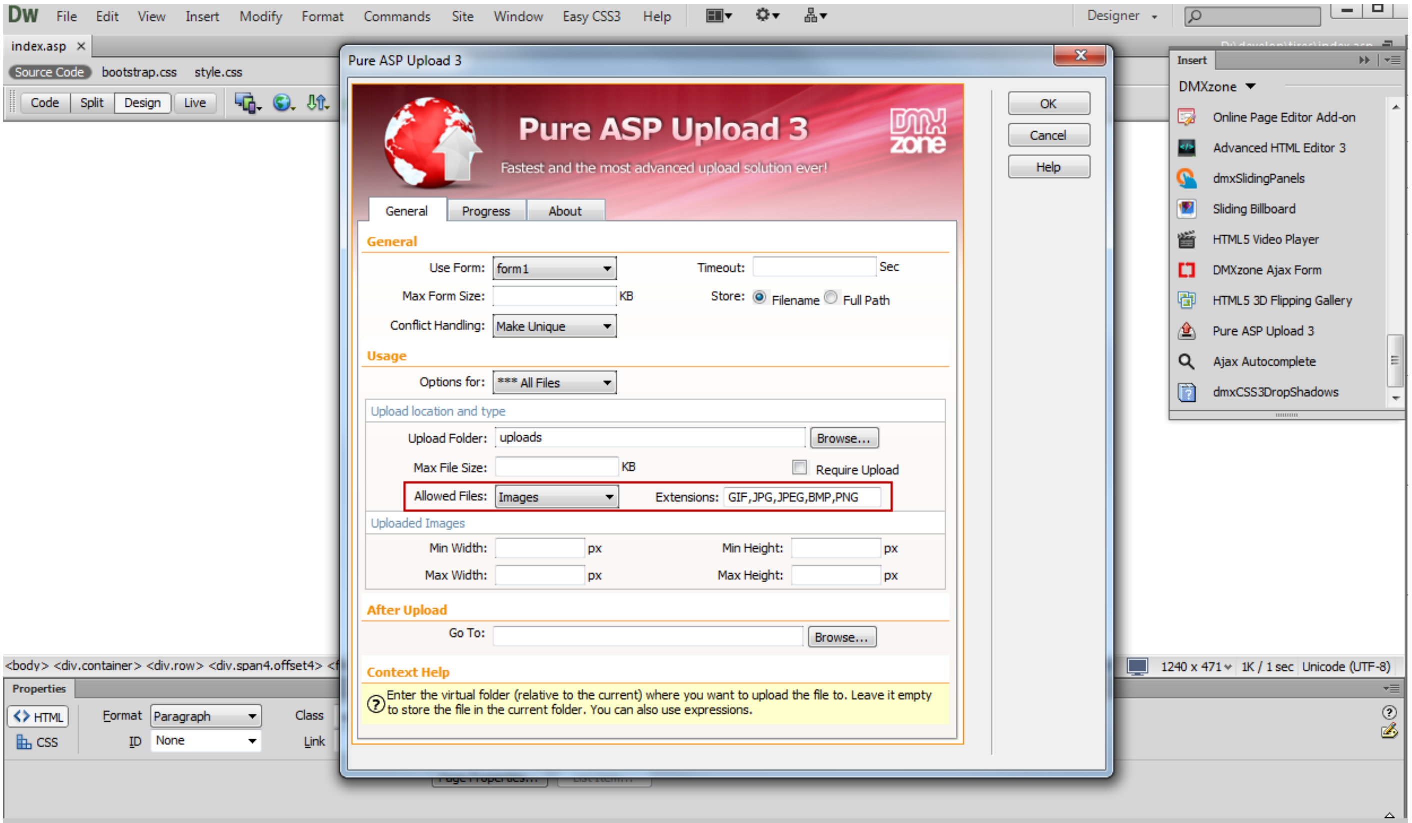
After Upload

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.

4. For our tutorial we limit the **uploaded file type** only to **images**.


DW File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
index.asp
Source Code bootstrap.css style.css
Code Split Design Live
OK
Cancel
Help
General Progress About
Insert
DMXzone
Online Page Editor Add-on
Advanced HTML Editor 3
dmxSlidingPanels
Sliding Billboard
HTML5 Video Player
DMXzone Ajax Form
HTML5 3D Flipping Gallery
Pure ASP Upload 3
Ajax Autocomplete
dmxCSS3DropShadows
<body> <div.container> <div.row> <div.span4.offset4>
1240 x 471 1K / 1 sec Unicode (UTF-8)
Properties
HTML Format Paragraph Class
CSS ID None Link

Pure ASP Upload 3

Pure ASP Upload 3

Fastest and the most advanced upload solution ever!

General

Use Form: form1

Timeout: Sec

Max Form Size: KB

Store: ● Filename ○ Full Path

Conflict Handling: Make Unique

Usage

Options for: *** All Files

Upload location and type

Upload Folder: uploads Browse...

Max File Size: KB

□ Require Upload

Allowed Files: Images

Extensions: GIF,JPG,JPEG,BMP,PNG

Uploaded Images

Min Width: px

Min Height: px

Max Width: px

Max Height: px

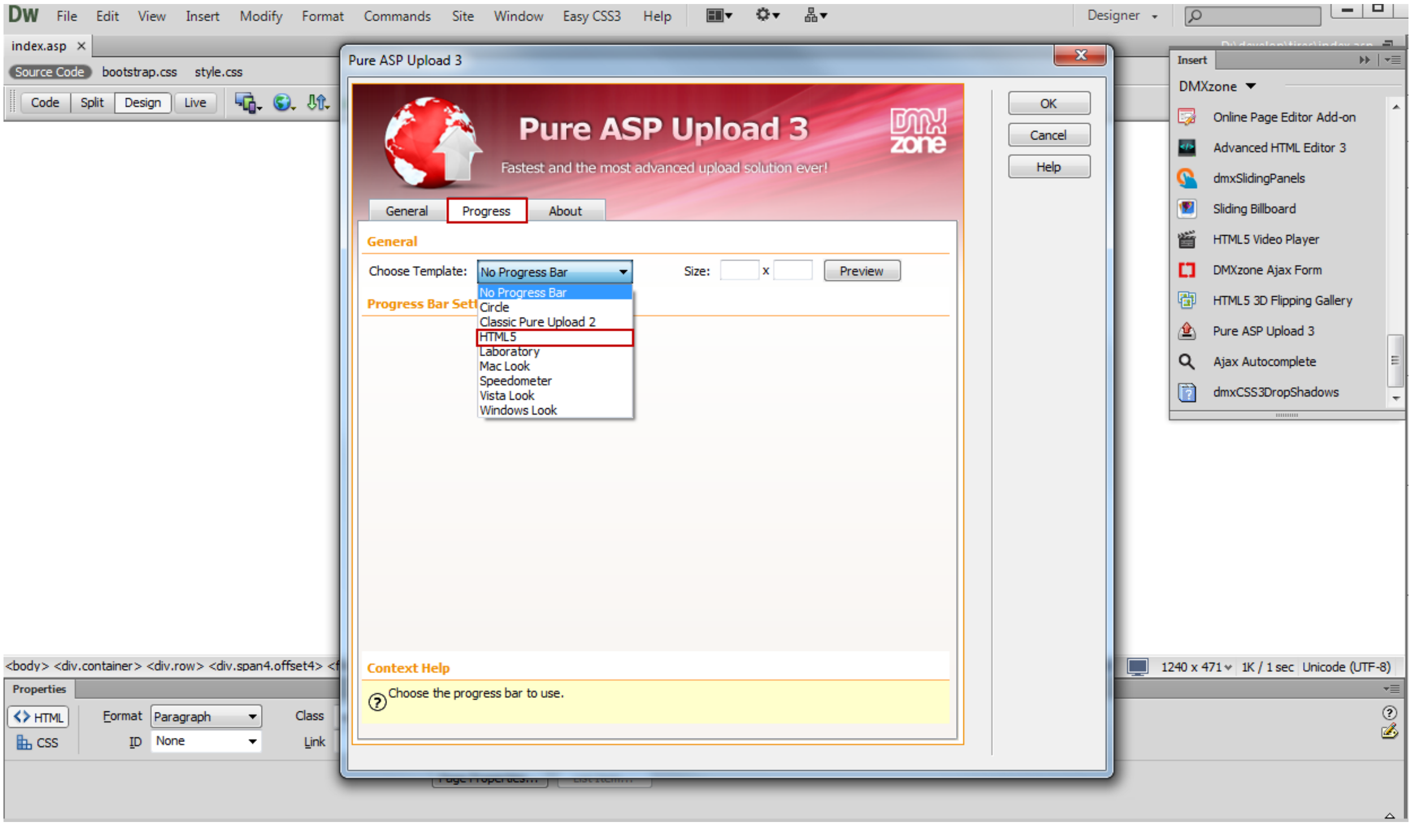
After Upload

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.

5. We choose the **HTML5 progress bar** from the **Progress tab**.



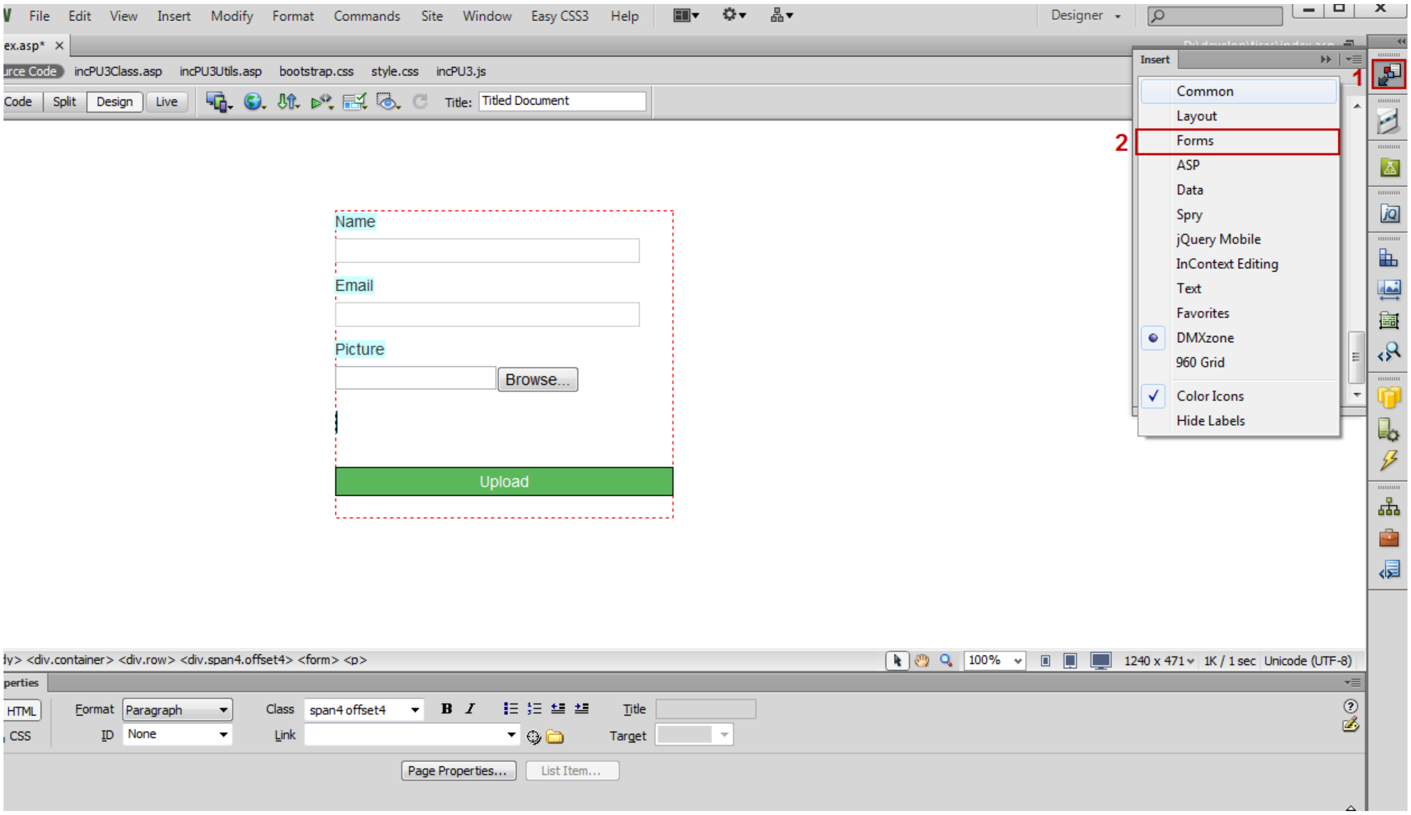
Pure ASP Upload 3

Choose Template: No Progress Bar

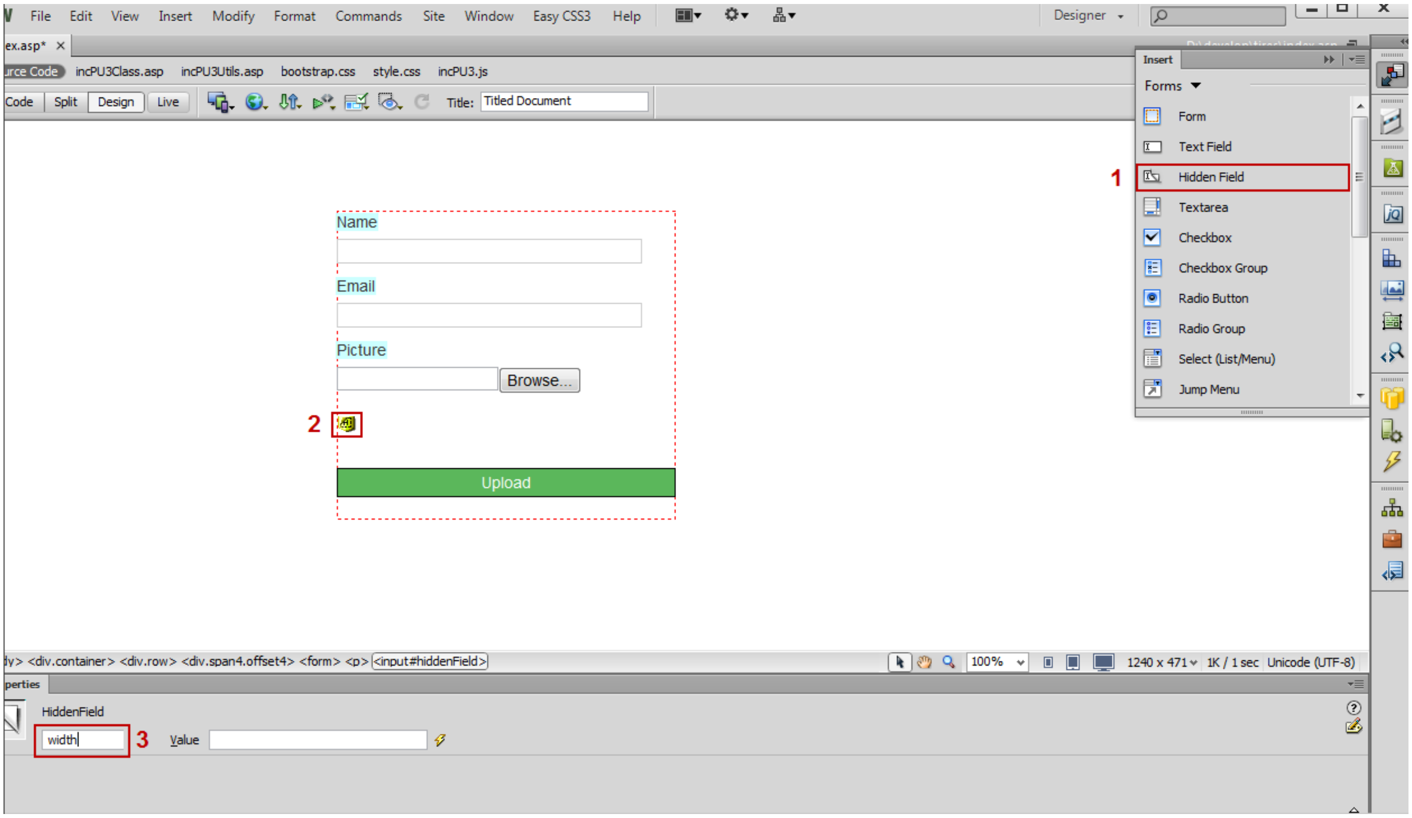
Context Help

Choose the progress bar to use.

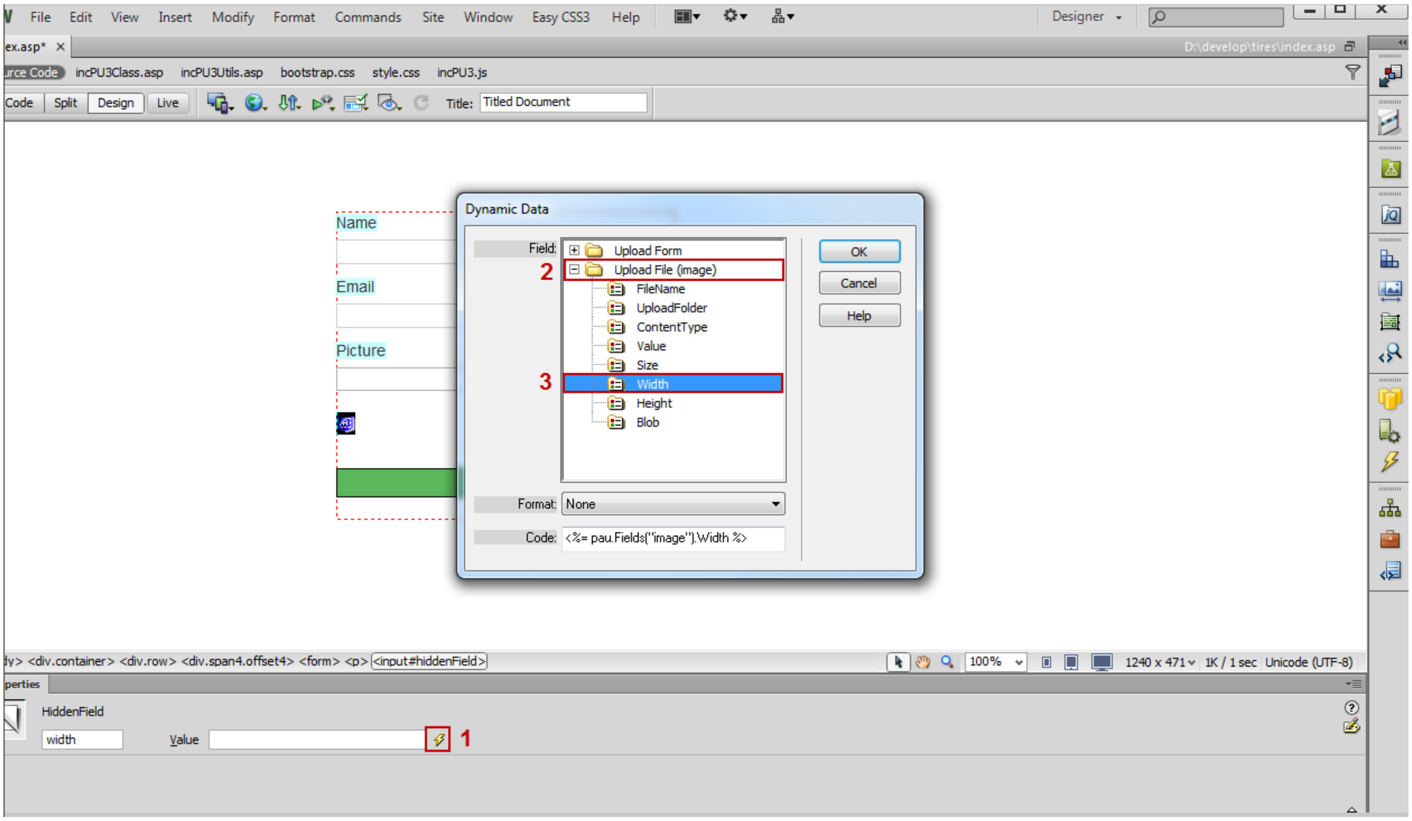
6. Now we are going to **add two hidden fields** that will be used to store the images width and height. From the **Insert menu (1)**, select the **Forms menu item (2)**.



7. Select the **hidden field (1)** and **click on it (2)** once it's added. We give this hidden field a proper name: **width (3)**.


File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
Code Split Design Live
Title: Titled Document
incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Insert
Forms
Form
Text Field
Hidden Field
Textarea
Checkbox
Checkbox Group
Radio Button
Radio Group
Select (List/Menu)
Jump Menu
1
Name
Email
Picture
Browse...
2
Upload
<div.container> <div.row> <div.span4.offset4> <form> <p> <input#hiddenField>
100%
1240 x 471
1K / 1 sec Unicode (UTF-8)
HiddenField
width
3
Value

8. Next, click the **Add dynamic item button (1)**, select **Upload file (2)** and choose the **Width binding (3)**. When you're done, click OK.



Dynamic Data

Field:
- Upload Form
- Upload File (image)
  - FileName
  - UploadFolder
  - ContentType
  - Value
  - Size
  - Width
  - Height
  - Blob

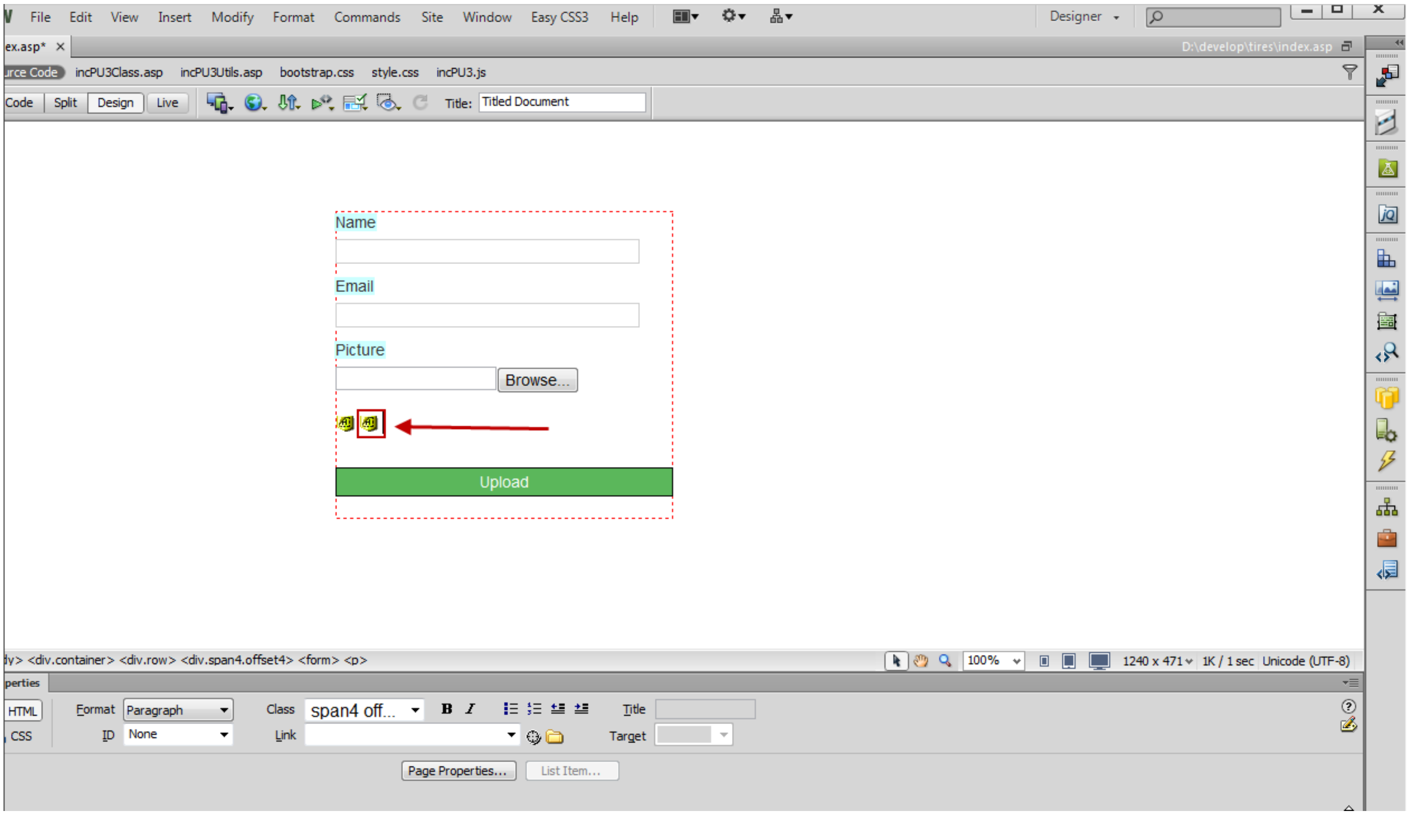
Format: None

Code: <%= pau.Fields("image").Width %>

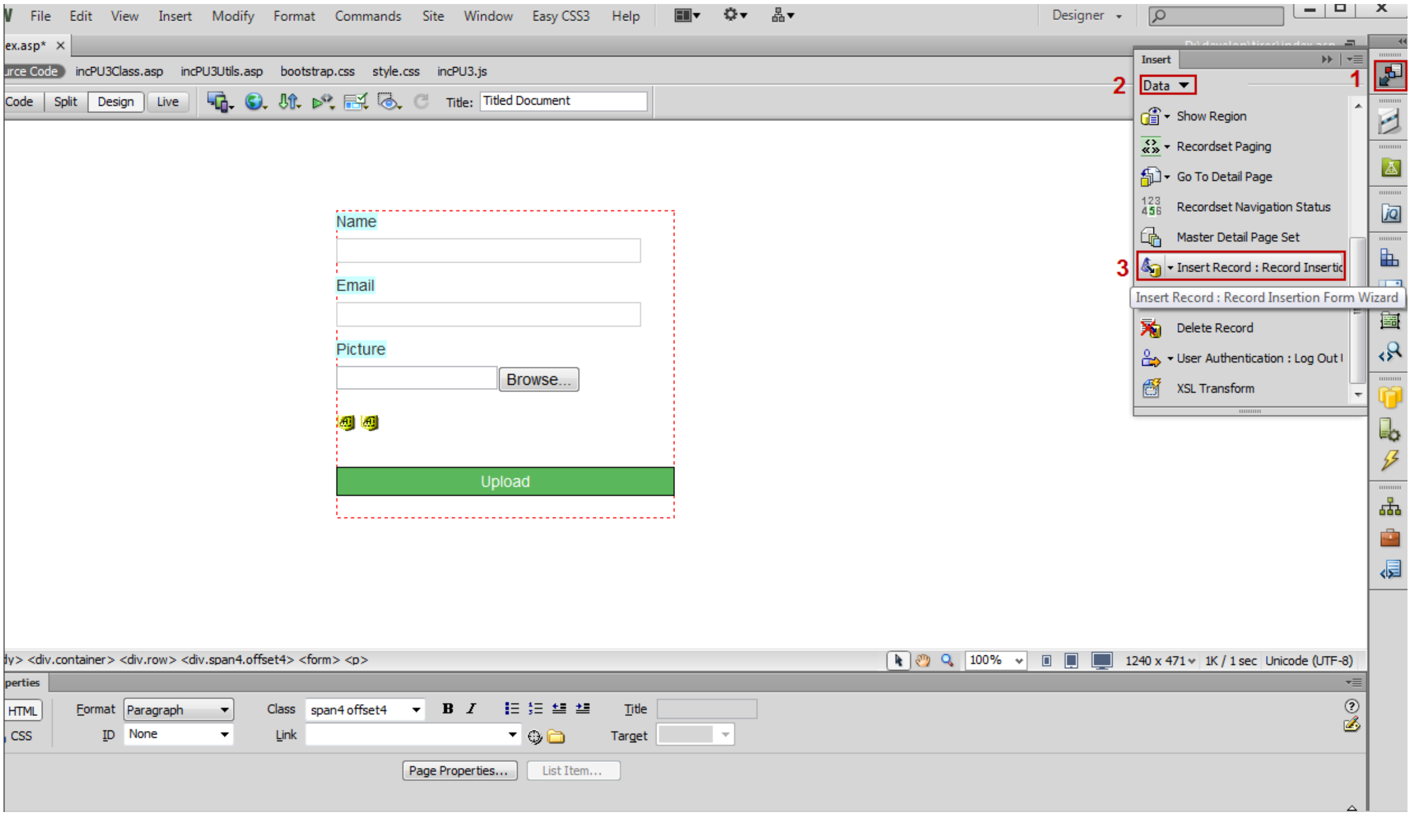
HiddenField: width

Value:

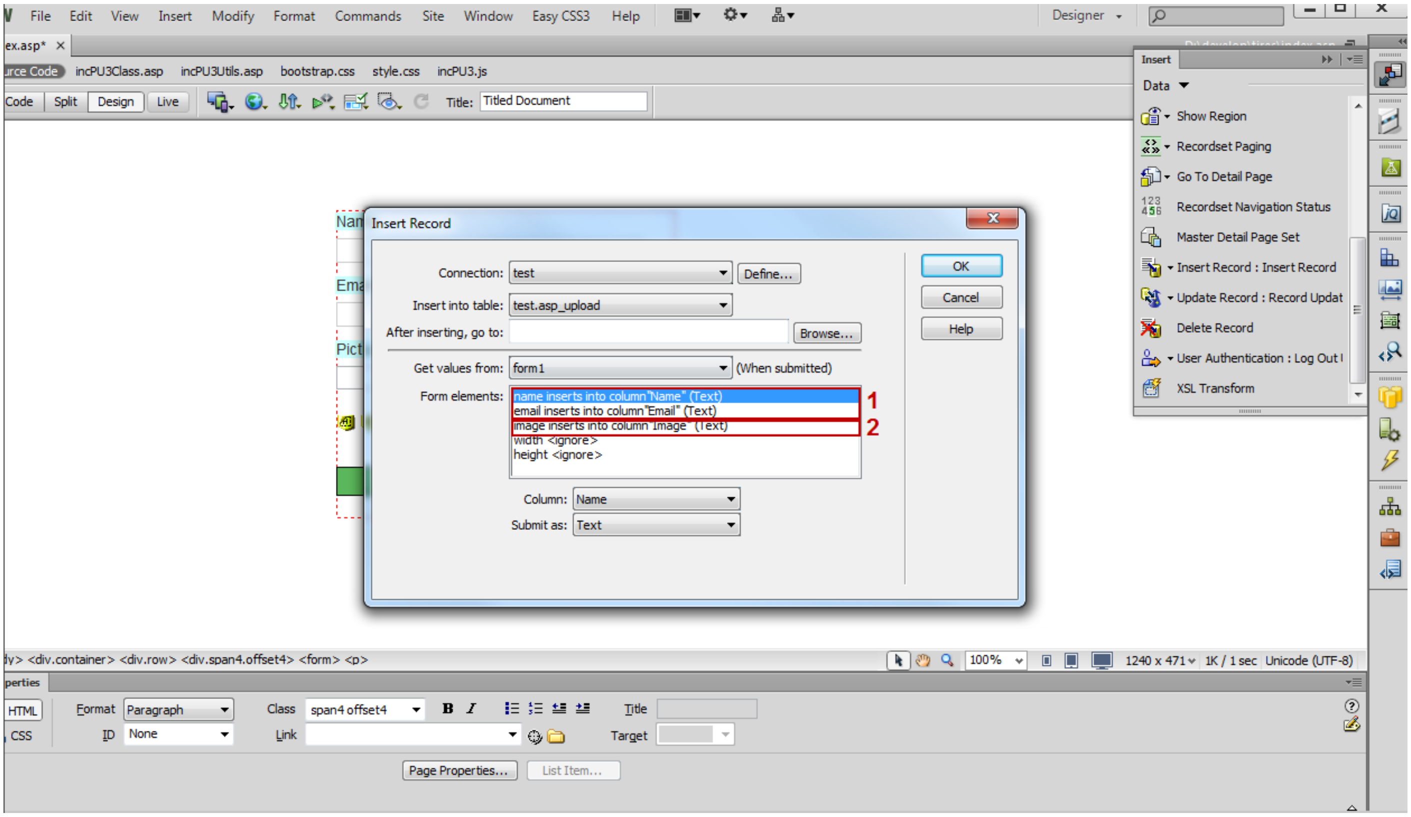
9. We do the same with the **height hidden field**.



10. Now, we're going to apply the **Insert record behavior**. From the **Insert menu (1)**, choose the **Data menu item (2)** and select **Insert Record (3)**.


File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
ex.asp*
incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Code Split Design Live
Title: Titled Document
Name
Email
Picture
Browse...
Upload
Insert
1
2
Data
Show Region
Recordset Paging
Go To Detail Page
Recordset Navigation Status
Master Detail Page Set
3
Insert Record : Record Insertio
Insert Record : Record Insertion Form Wizard
Delete Record
User Authentication : Log Out U
XSL Transform
<div.container> <div.row> <div.span4.offset4> <form> <p>
100%
1240 x 471
1K / 1 sec Unicode (UTF-8)
HTML
CSS
Format Paragraph
ID None
Class span4 offset4
Link
Title
Target
Page Properties...
List Item...

11. In the **Insert record window**, we set the right database columns for our **text fields (1)** and store the **Image name** into a database column named Image **(2)**.



Insert Record

Connection: test

Insert into table: test.asp_upload

After inserting, go to:

Get values from: form1 (When submitted)

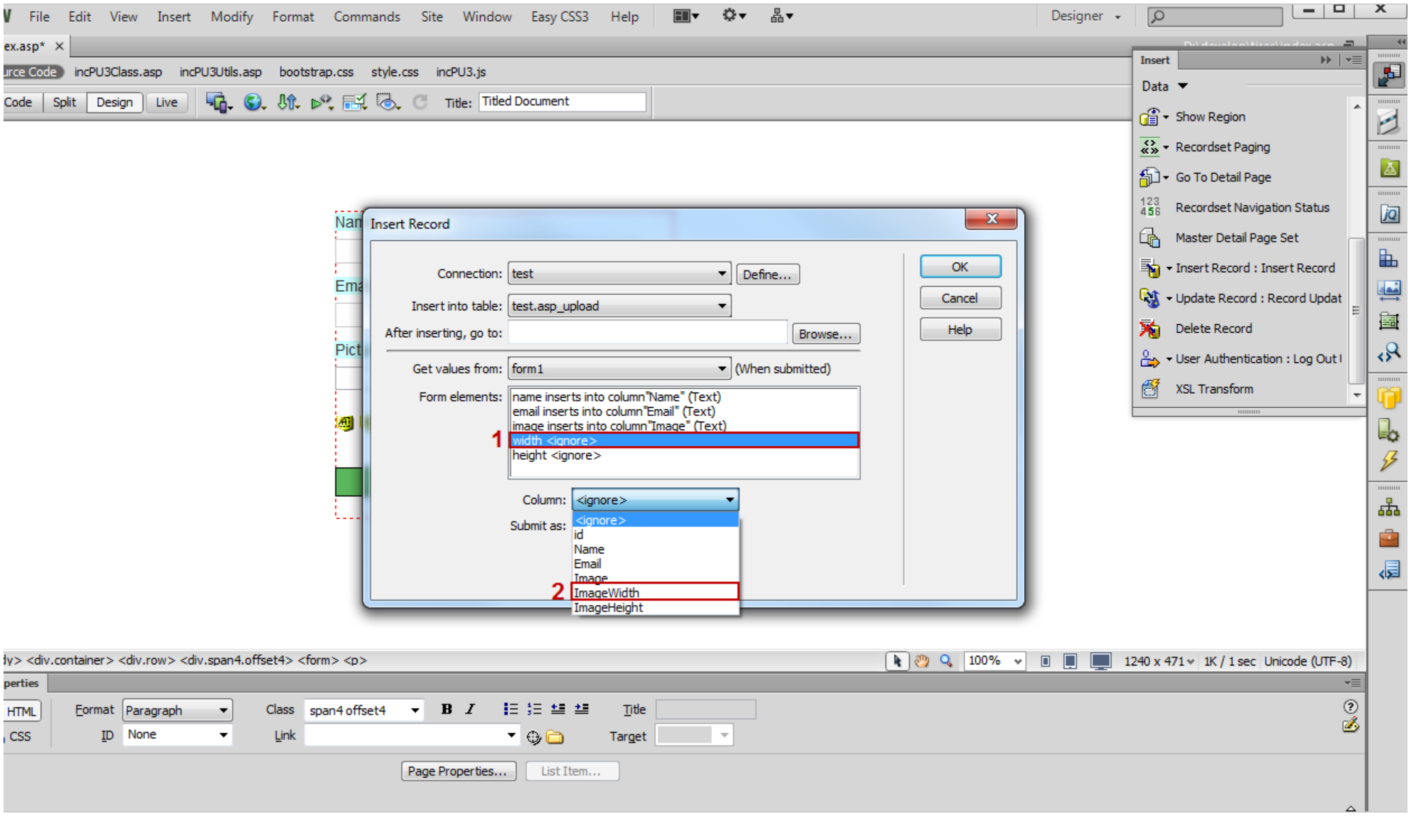
Form elements:

name inserts into column"Name" (Text)
email inserts into column"Email" (Text)
image inserts into column"Image" (Text)
width <ignore>
height <ignore>

Column: Name

Submit as: Text

12. In the next step we are going to store the **width hidden field (1)** value into the **ImageWidth database column (2)**.



Insert Record

Connection: test

Insert into table: test.asp_upload

After inserting, go to:

Get values from: form1 (When submitted)

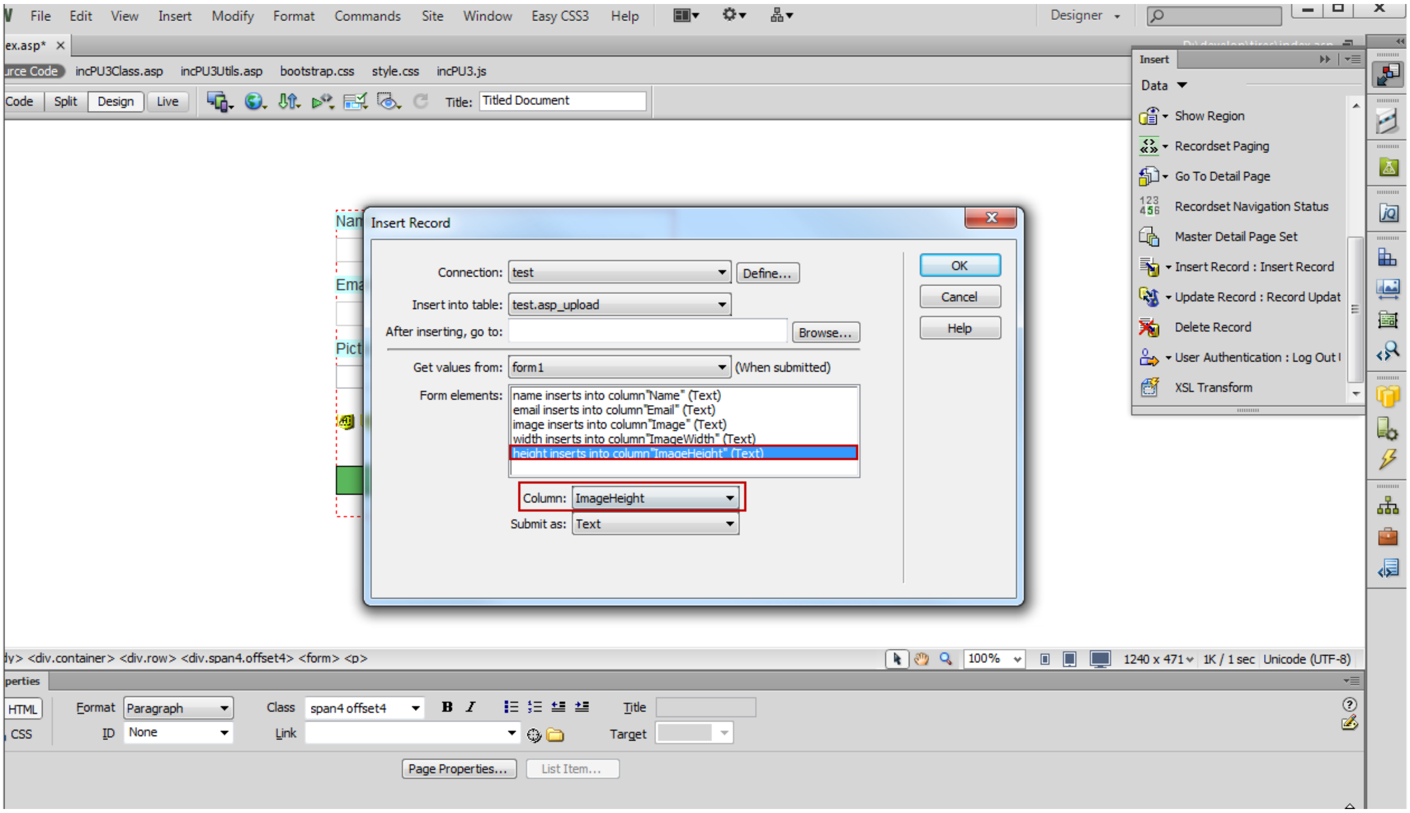
Form elements:
name inserts into column "Name" (Text)
email inserts into column "Email" (Text)
image inserts into column "Image" (Text)
1 width <ignore>
height <ignore>

Column: <ignore>

Submit as:

<ignore>
id
Name
Email
Image
2 ImageWidth
ImageHeight

13. We do the same for the **height hidden field**.



Insert Record

Connection: test

Insert into table: test.asp_upload

After inserting, go to:

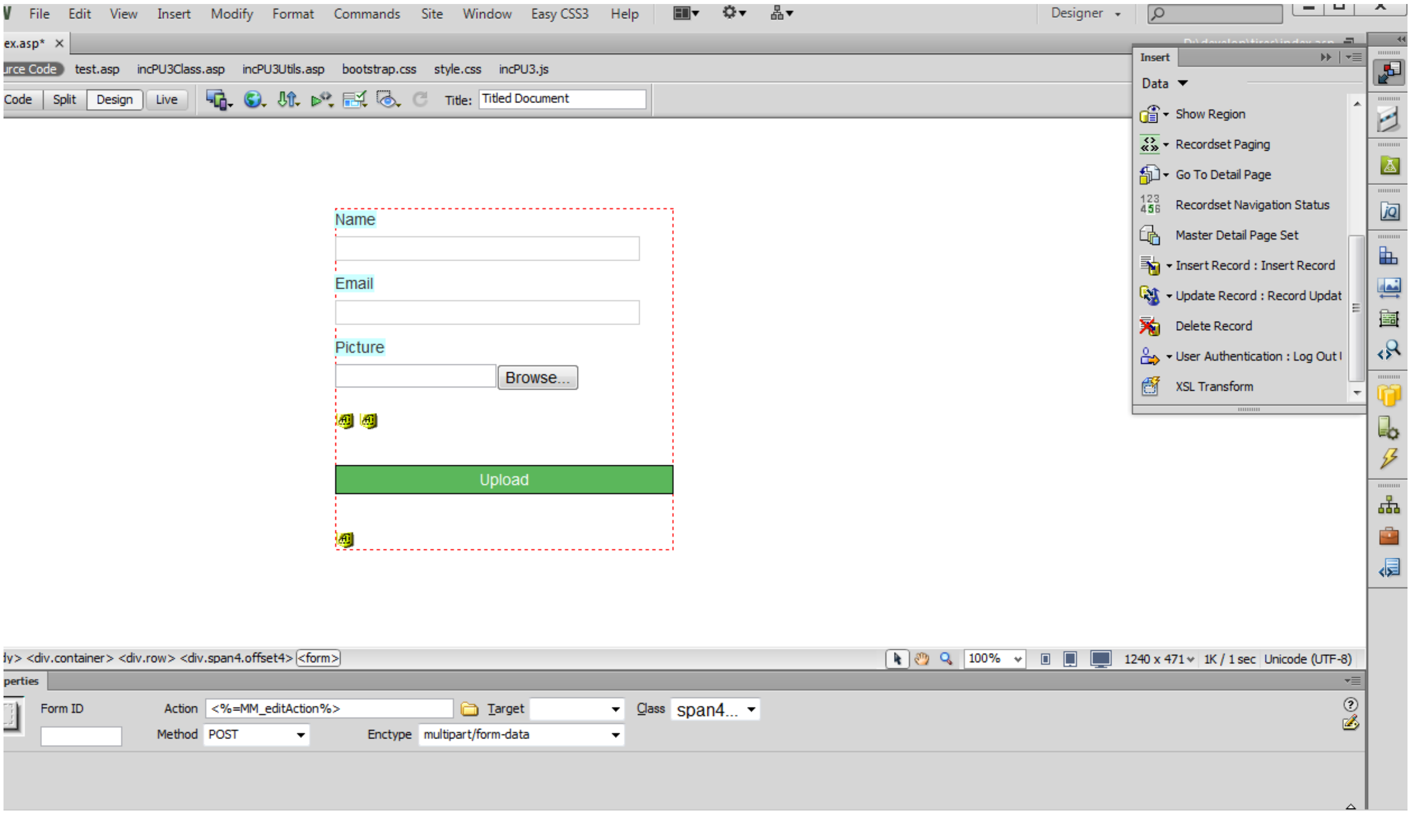
Get values from: form1 (When submitted)

Form elements:
name inserts into column"Name" (Text)
email inserts into column"Email" (Text)
image inserts into column"Image" (Text)
width inserts into column"ImageWidth" (Text)
height inserts into column"ImageHeight" (Text)

Column: ImageHeight

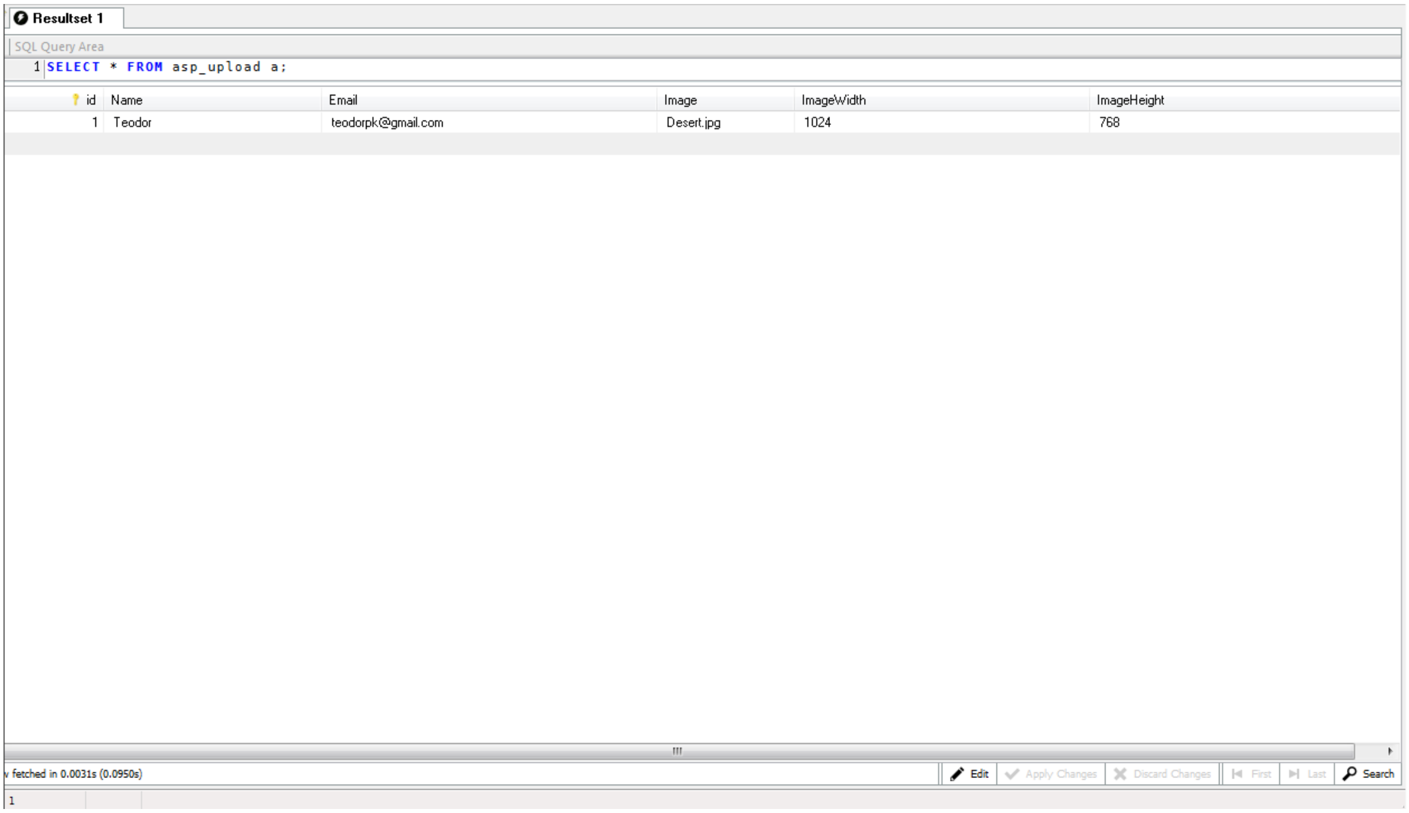
Submit as: Text

14. Now you can save your work and preview the result in the browser.



File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
ex.asp*
test.asp incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Code Split Design Live
Title: Titled Document
Insert
Data
Show Region
Recordset Paging
Go To Detail Page
Recordset Navigation Status
Master Detail Page Set
Insert Record : Insert Record
Update Record : Record Updat
Delete Record
User Authentication : Log Out I
XSL Transform
Name
Email
Picture
Browse...
Upload
<div.container> <div.row> <div.span4.offset4> <form>
100%
1240 x 471
1K / 1 sec Unicode (UTF-8)
Form ID
Action <%=MM_editAction%>
Target
Class span4...
Method POST
Enctype multipart/form-data

15. That's how you can upload images and save their names, width and height into a database.



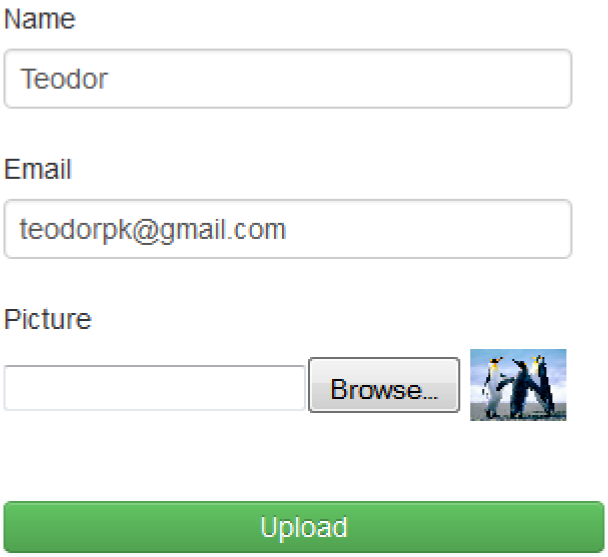
```
SELECT * FROM asp_upload a;
```

id	Name	Email	Image	ImageWidth	ImageHeight
1	Teodor	teodorpk@gmail.com	Desert.jpg	1024	768

```
v fetched in 0.0031s (0.0950s)
```

Advanced: Using Pure ASP Upload 3 with Update Record

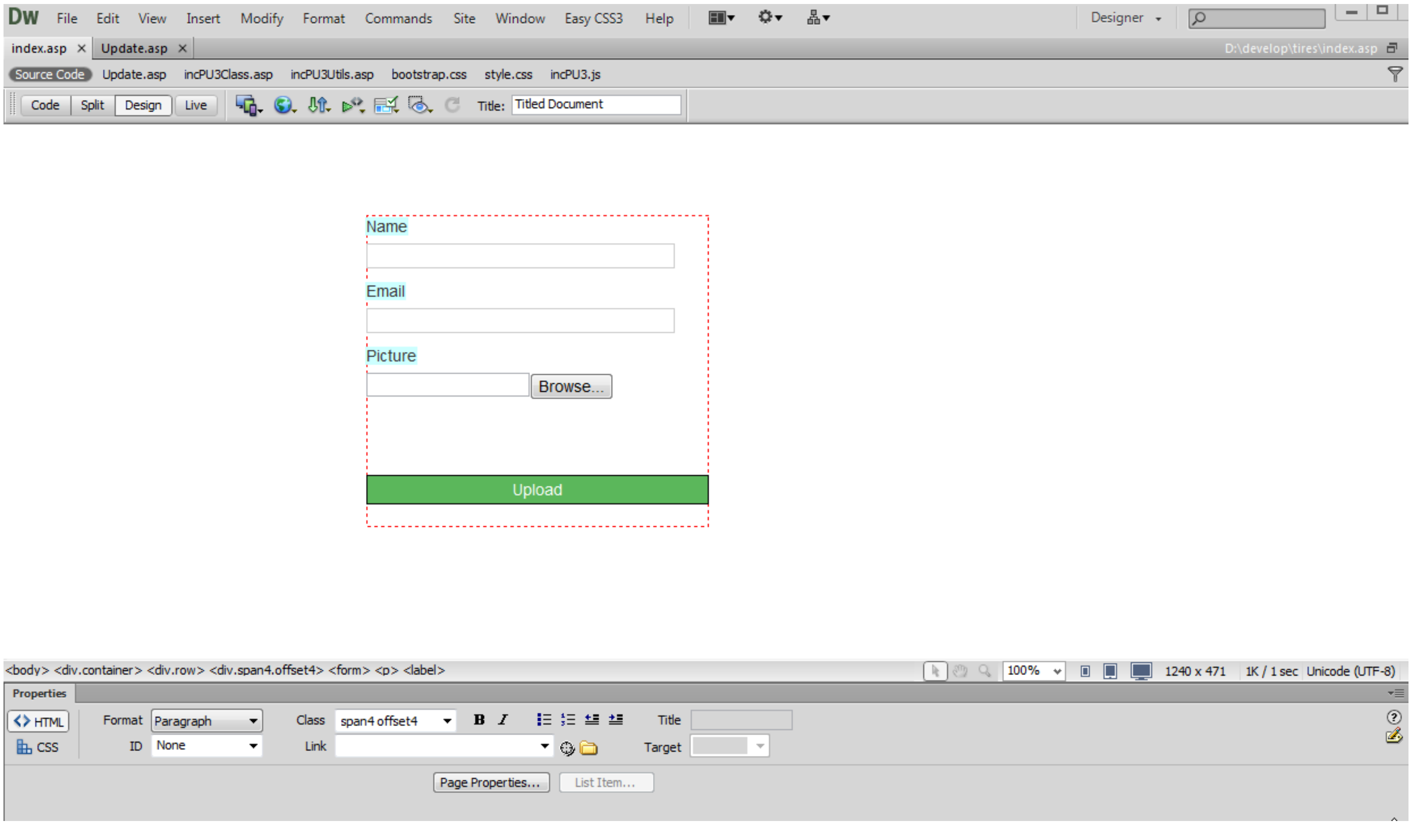
In this tutorial we will show you how to use **Pure ASP Upload 3** with Update Record server behavior.
*You can also check out the **video**.



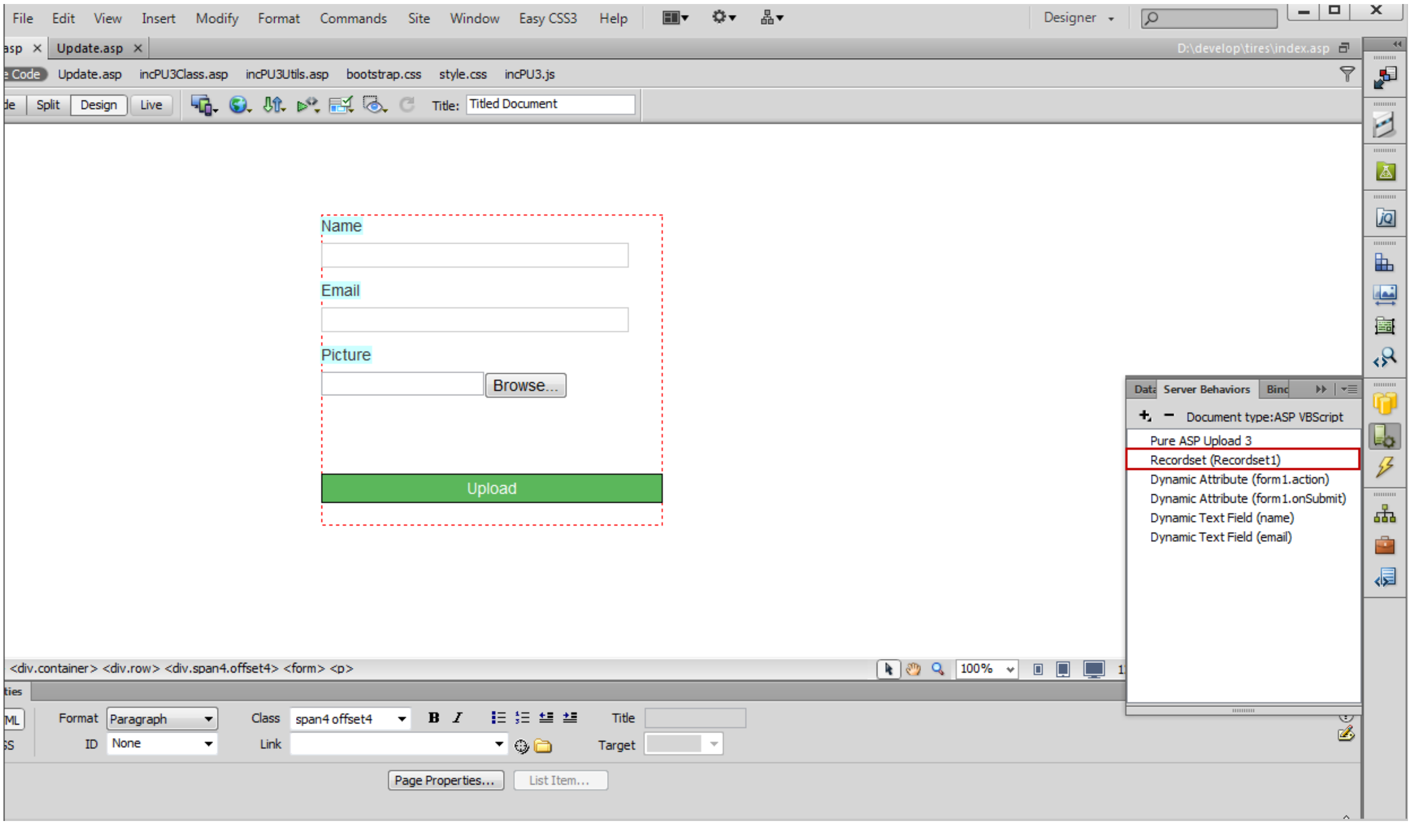
Name
Teodor
Email
teodorpk@gmail.com
Picture

How to do it

1. We've already created a form that contains three fields – name, email and picture._Now, we're going to **apply the Update record behavior** to this page so we can update the fields' info.


DW File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
index.asp
Update.asp
D:\develop\tires\index.asp
Source Code Update.asp incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Code Split Design Live
Title: Titled Document
Name
Email
Picture
Browse...
Upload
<body> <div.container> <div.row> <div.span4.offset4> <form> <p> <label>
100%
1240 x 471 1K / 1 sec Unicode (UTF-8)
Properties
HTML
CSS
Format Paragraph
ID None
Class span4 offset4
Link
Title
Target
Page Properties...
List Item...

2. We’ve already applied **Pure ASP Upload 3** to this form and created a **standard Dreamweaver recordset**.


File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
Update.asp
D:\develop\tires\index.asp
Code Update.asp incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Split Design Live
Name
Email
Picture
Browse...
Upload
Data Server Behaviors Bind
Page Properties...
List Item...
100%

Title: Titled Document

Document type:ASP VBScript

- Pure ASP Upload 3
- Recordset (Recordset1)
- Dynamic Attribute (form1.action)
- Dynamic Attribute (form1.onSubmit)
- Dynamic Text Field (name)
- Dynamic Text Field (email)

<div.container> <div.row> <div.span4.offset4> <form> <p>

Format Paragraph

ID None

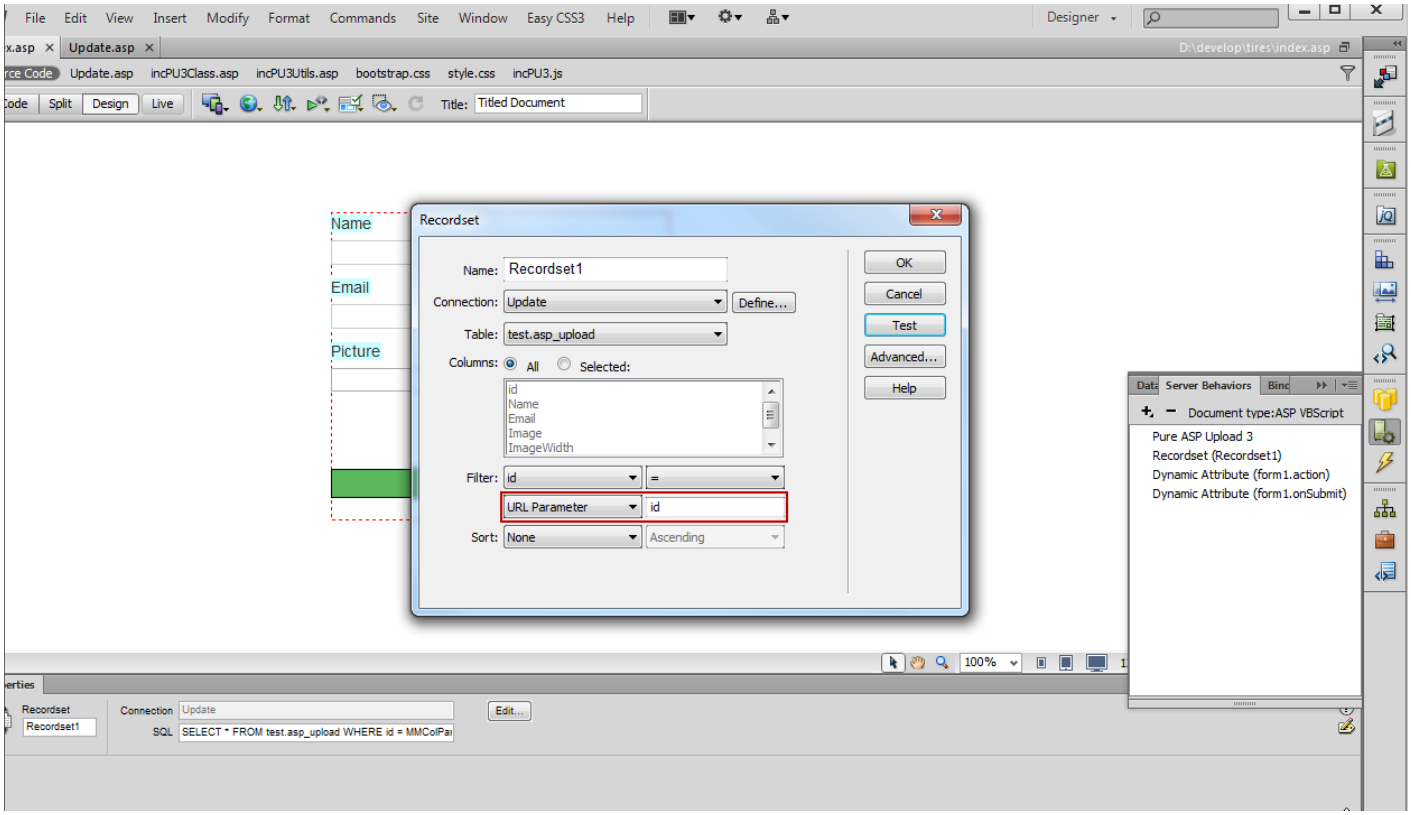
Class span4 offset4

Link

Title

Target

3. In the **Recordset window**, we're filtering the recordset by an URL parameter named **ID**.



Recordset

Name: Recordset1

Connection: Update

Table: test.asp_upload

Columns: ● All ○ Selected:

id
Name
Email
Image
ImageWidth

Filter: id =

URL Parameter id

Sort: None Ascending

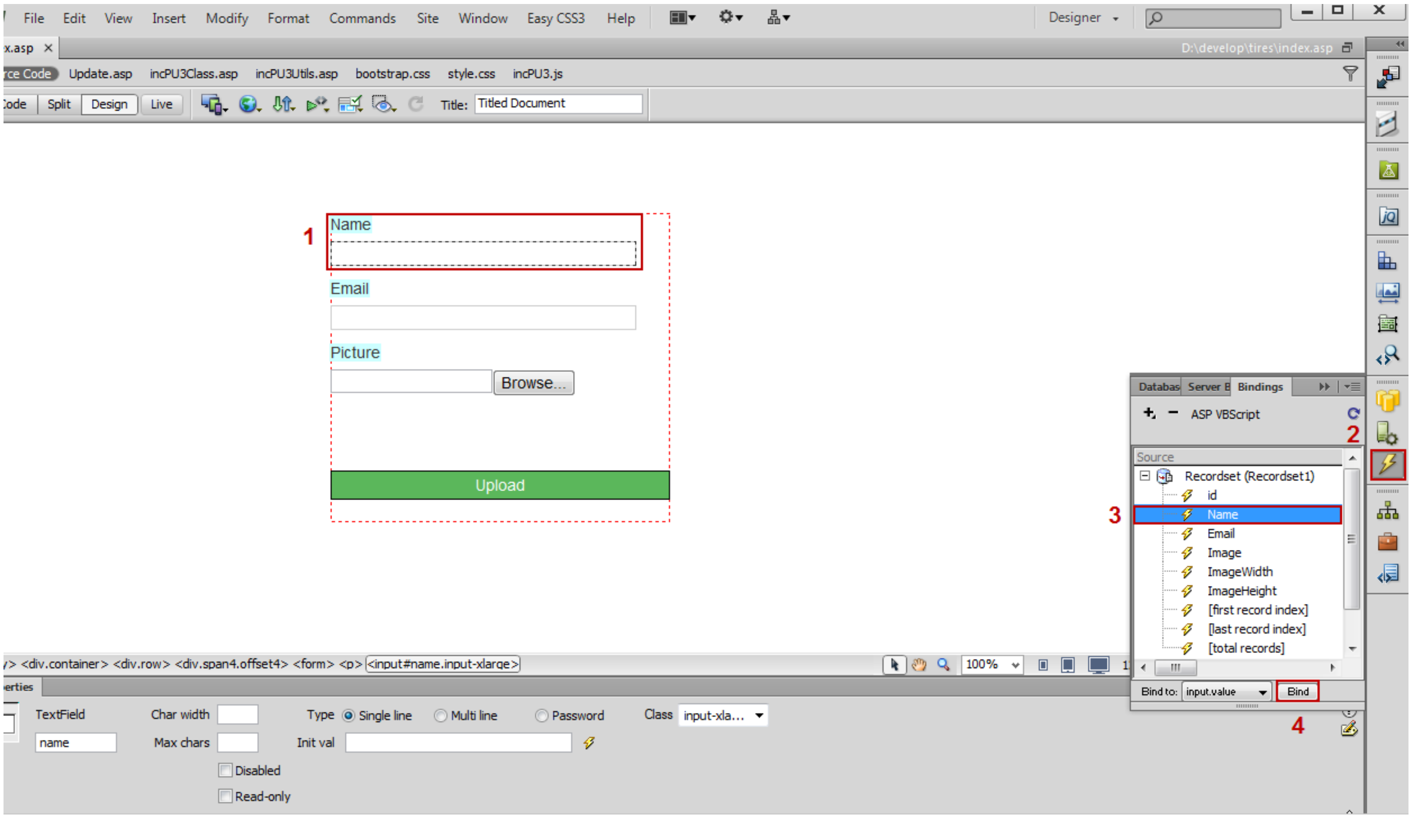
Pure ASP Upload 3
Recordset (Recordset1)
Dynamic Attribute (form1.action)
Dynamic Attribute (form1.onSubmit)

Recordset: Recordset1

Connection: Update

SQL: SELECT * FROM test.asp_upload WHERE id = MMColPar

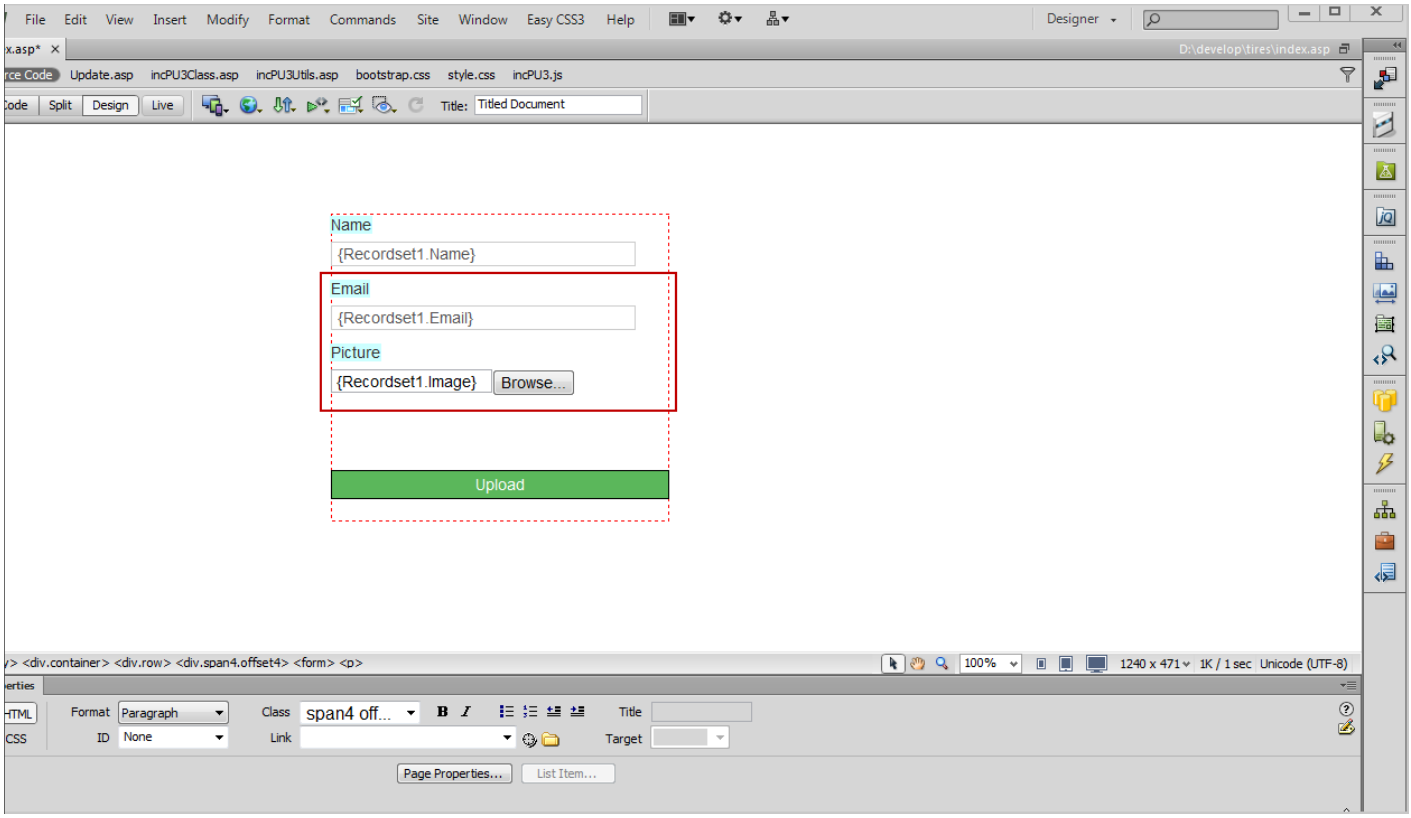
4. Next, select the **Name field (1)** and click the **Bindings button (2)**, choose the **Name binding (3)** and click the **Bind button (4)**.


File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
x.asp
D:\develop\tires\index.asp
rce Code Update.asp incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Code Split Design Live
Title: Titled Document
1
Name
Email
Picture
Browse...
Upload
Databas Server B Bindings
ASP VBScript
2
Source
Recordset (Recordset1)
id
3
Name
Email
Image
ImageWidth
ImageHeight
[first record index]
[last record index]
[total records]
Bind
4
<div.container> <div.row> <div.span4.offset4> <form> <p> <input#name.input-xlarge>
100%
erties

Bind to: input.value

TextField: name
Char width:
Max chars:
Type: ● Single line ○ Multi line ○ Password
Class: input-xla...
Init val:
□ Disabled
□ Read-only

5. We do the same for **the Email and Picture fields**, choosing the corresponding bindings - email and image.


File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
x.asp*
D:\develop\tires\index.asp
rce Code Update.asp incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Code Split Design Live
Browse...
Upload
<div.container> <div.row> <div.span4.offset4> <form> <p>
100% 1240 x 471 1K / 1 sec Unicode (UTF-8)
erties
HTML
CSS
Bold Italic
Link
Target
Page Properties...
List Item...

Title: Titled Document

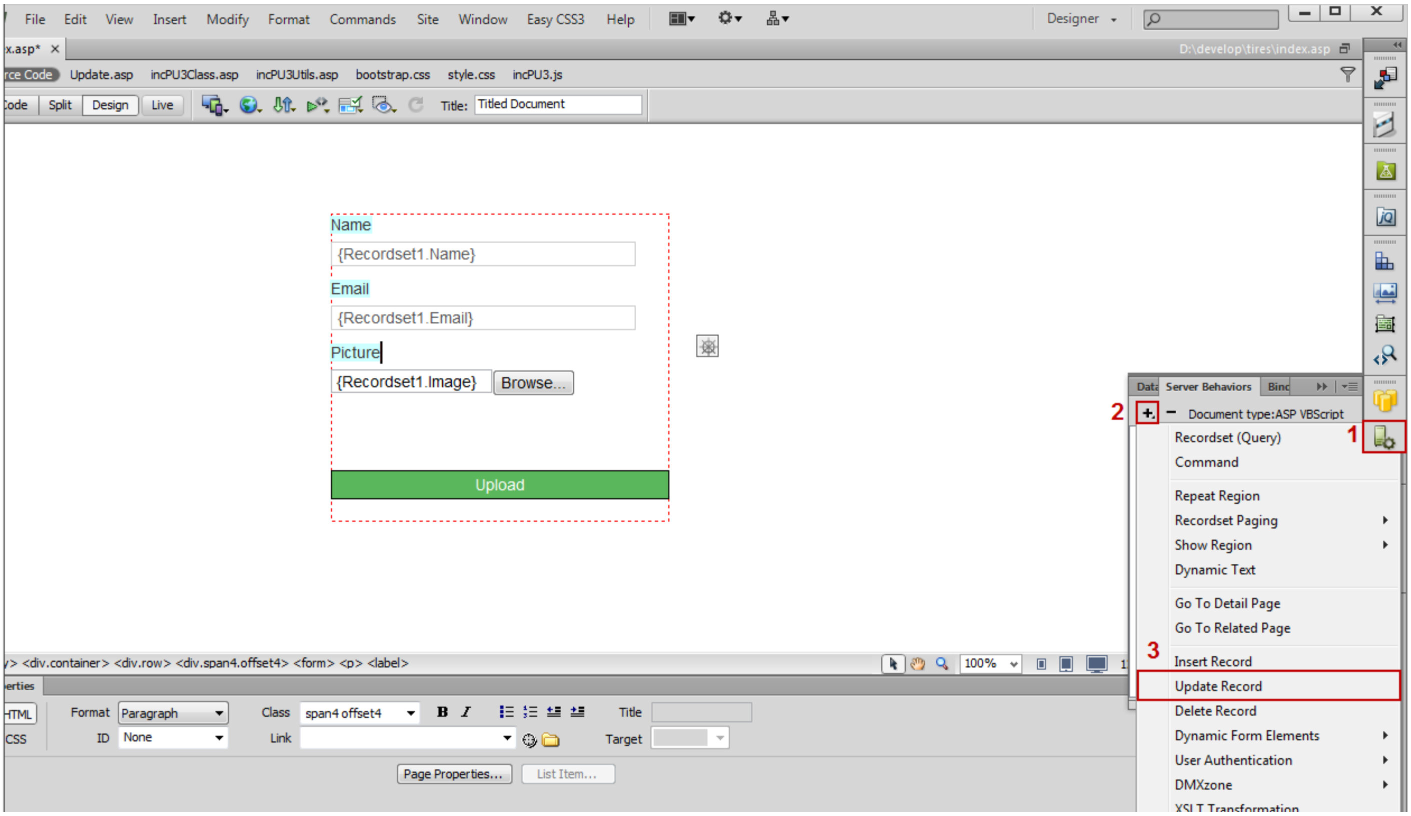
Name
{Recordset1.Name}

Email
{Recordset1.Email}

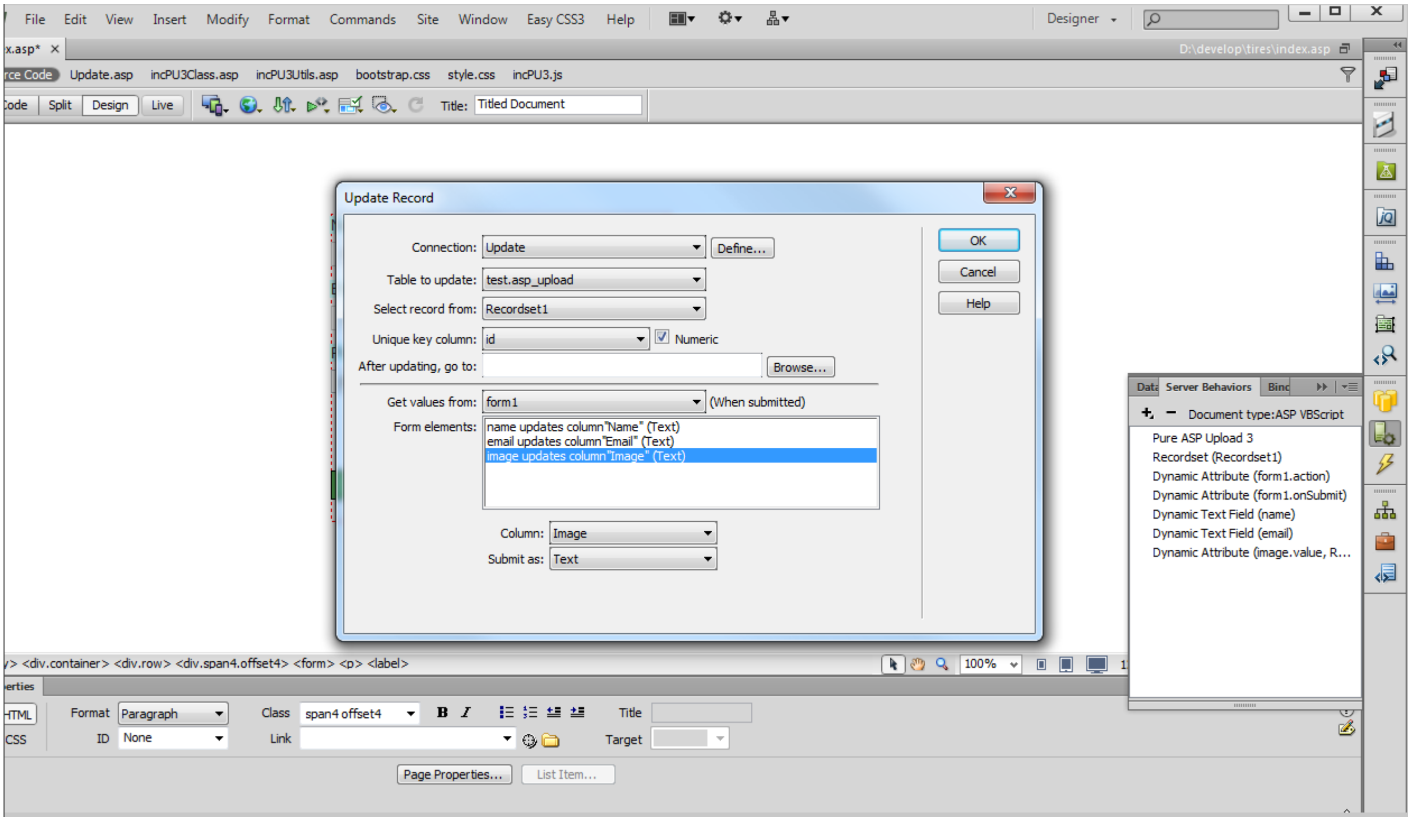
Picture
{Recordset1.Image}

Format: Paragraph
ID: None
Class: span4 off...
Title:

6. Now we are going to apply the **Update Record behavior**. Click the **Server behavior button (1)**, select the **Add item button (2)** and choose **Update record menu item (3)**.



7. In the Update record window, make sure that your form fields are assigned to the corresponding fields in your database and click the OK button.



Update Record

Connection: Update

Table to update: test.asp_upload

Select record from: Recordset1

Unique key column: id ☑ Numeric

After updating, go to:

Get values from: form1 (When submitted)

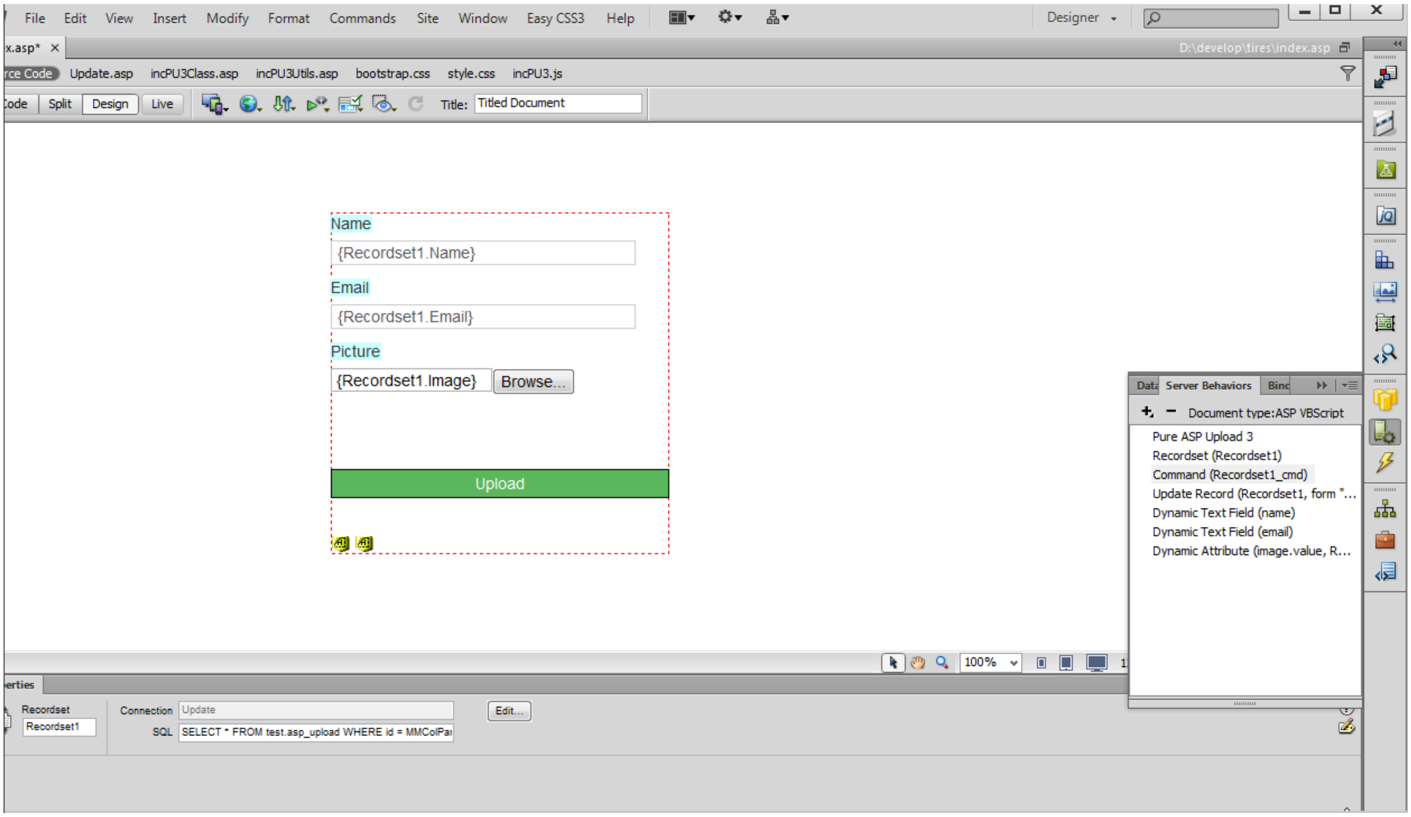
Form elements:
name updates column "Name" (Text)
email updates column "Email" (Text)
image updates column "Image" (Text)

Column: Image

Submit as: Text

Document type:ASP VBScript
Pure ASP Upload 3
Recordset (Recordset1)
Dynamic Attribute (form1.action)
Dynamic Attribute (form1.onSubmit)
Dynamic Text Field (name)
Dynamic Text Field (email)
Dynamic Attribute (image.value, R...

8. Now you can save your work and preview the result in the browser.



Title: Titled Document

Name

{Recordset1.Name}

Email

{Recordset1.Email}

Picture

{Recordset1.Image} Browse...

Upload

Document type:ASP VBScript

- Pure ASP Upload 3
- Recordset (Recordset1)
- Command (Recordset1_cmd)
- Update Record (Recordset1, form "...
- Dynamic Text Field (name)
- Dynamic Text Field (email)
- Dynamic Attribute (image.value, R...

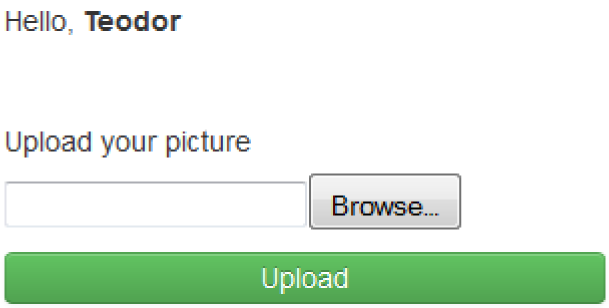
Recordset: Recordset1

Connection: Update

SQL: SELECT * FROM test.asp_upload WHERE id = MMColPar

Advanced: Uploading Files to a Dynamic Folder

In this tutorial we will show you how to upload files to a Dynamic folder using **Pure ASP Upload 3**.
*You can also check out the **video**.

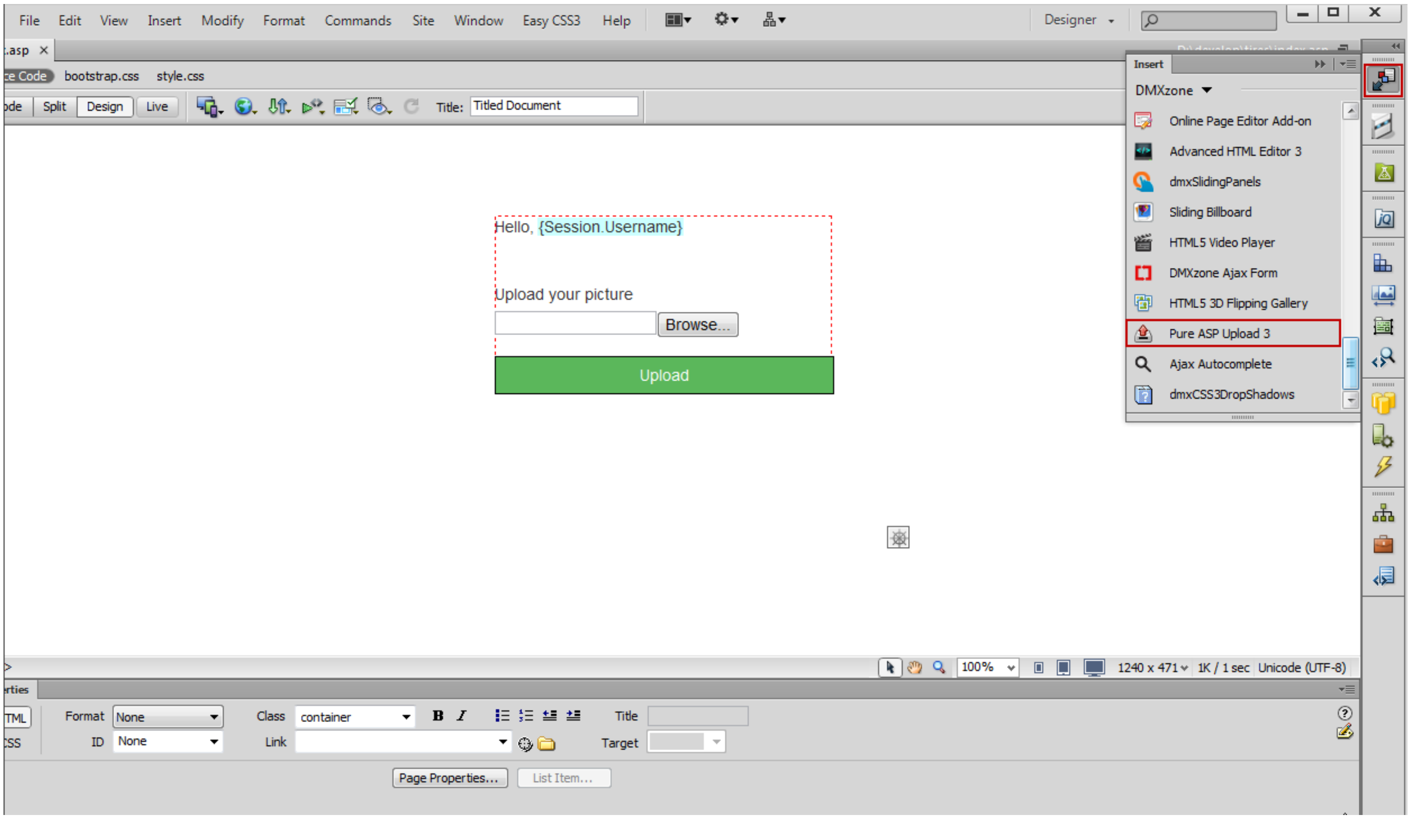


Hello, **Teodor**

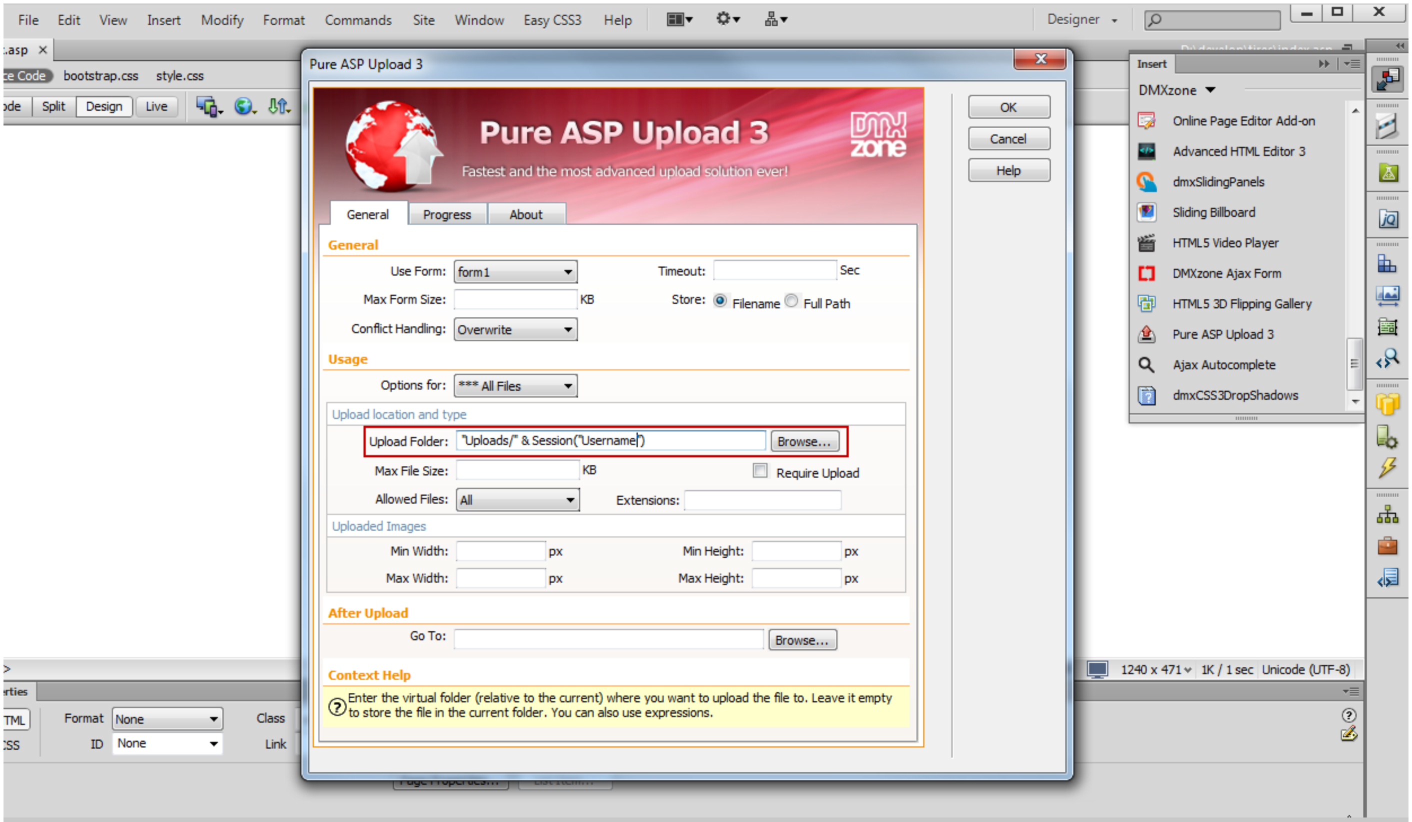
Upload your picture

How to do it

1. For this tutorial we use a **session variable** that stores a username. We are going to use the username as a folder name. Click the **Insert button (1)** and select the **Pure ASP Upload 3 icon (2)** to open its UI.



2. Our upload folder is named **"Uploads"** and we want to create a folder for each user inside it**.** Enter: **"Uploads/" & Session ("Username")**. Username is the session variable that stores the username. This will create a new folder for every user that tries to upload. Non-existing folders will be auto-created



Pure ASP Upload 3

Fastest and the most advanced upload solution ever!

General

Use Form: form1

Timeout: Sec

Max Form Size: KB

Store: ● Filename ○ Full Path

Conflict Handling: Overwrite

Usage

Options for: *** All Files

Upload location and type

Upload Folder: "Uploads/" & Session("Username")

Max File Size: KB

□ Require Upload

Allowed Files: All

Extensions:

Uploaded Images

Min Width: px

Min Height: px

Max Width: px

Max Height: px

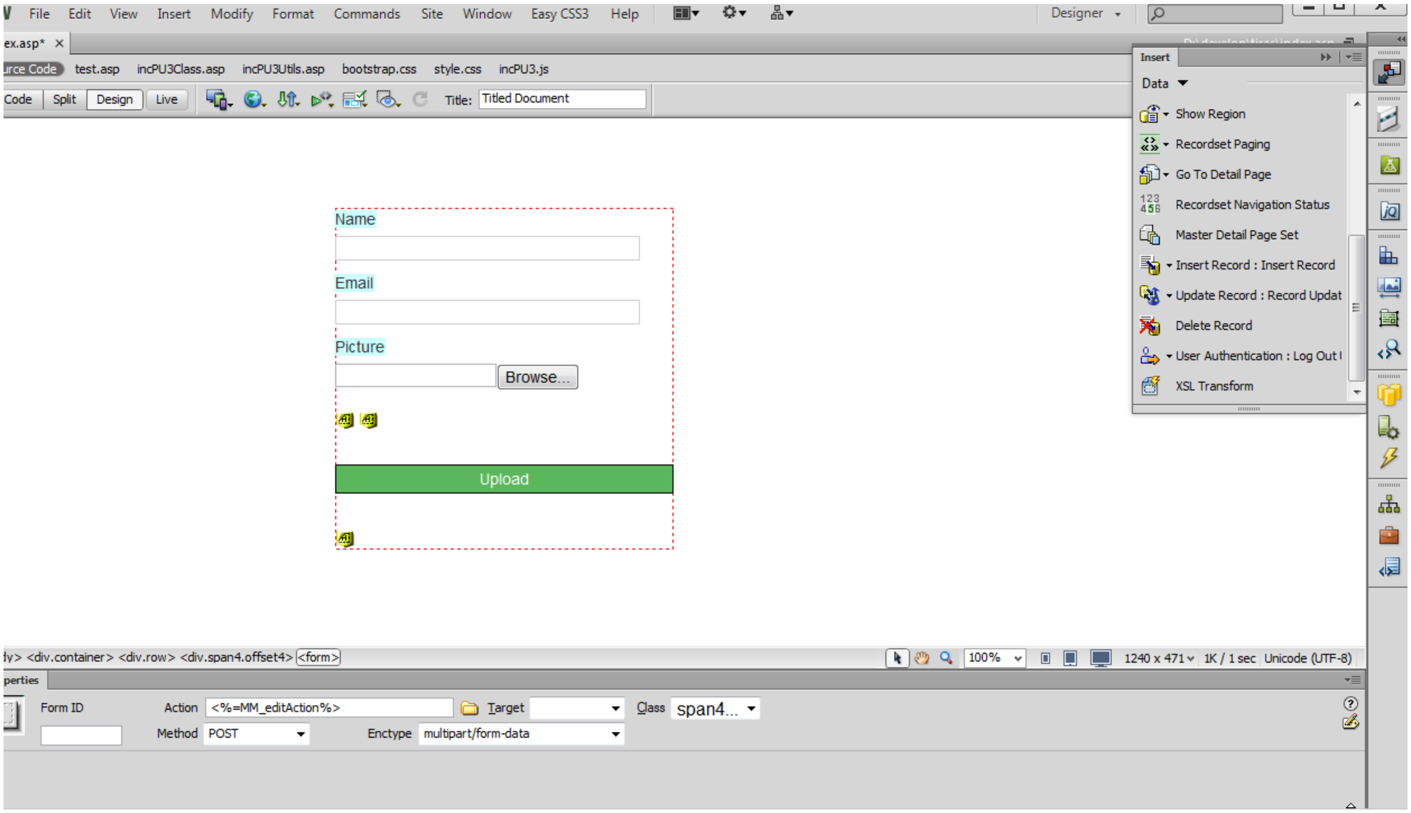
After Upload

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. Now you can save your work and preview the result in the browser.



File Edit View Insert Modify Format Commands Site Window Easy CSS3 Help
Designer
ex.asp*
urce Code test.asp incPU3Class.asp incPU3Utils.asp bootstrap.css style.css incPU3.js
Code Split Design Live
Title: Titled Document
Name
Email
Picture
Browse...
Upload
Insert
Data
Show Region
Recordset Paging
Go To Detail Page
Recordset Navigation Status
Master Detail Page Set
Insert Record : Insert Record
Update Record : Record Updat
Delete Record
User Authentication : Log Out I
XSL Transform
dy> <div.container> <div.row> <div.span4.offset4> <form>
100%
1240 x 471 1K / 1 sec Unicode (UTF-8)
perties

Form ID

Action <%=MM_editAction%>

Target

Class span4...

Method POST

Enctype multipart/form-data

Advanced: Configure Uploads for IIS Web Applications

This article describes how to configure Internet Information Services (IIS) to allow more-secure file uploads through a Web application. Many Web applications such as Content Management Systems require supporting file uploads to the Web server that uses the Web application. Allowing files to be uploaded to the Web server that uses the Web application has security ramifications for the server, and you must understand all the implications for allowing this. This article guides you through securing your Web application uploads through IIS configuration. If your Web application has an automated installer, you can also incorporate the configuration in this article into your installer.

Note Some Web applications use databases to manage uploaded content. However, this article focuses on applications that use the file system.

How to do it

Create a separate folder for your uploaded content and change the NTFS file permissions on the upload folder

By doing this, you can configure the behavior of uploaded content differently from the rest of your Web application. Grant the upload folder Read and Write permissions for the IIS worker process identity. For IIS 6.0 in Windows Server 2003, you can use the IIS_WPG user group for this. For IIS 7.0 and later, you can use the IIS_IUSRS user group.

For more information about IIS_WPG, visit the following Microsoft Web page:

Configuring Application Pool Identity in IIS 6.0

For more information about IIS_ISURS, visit the following Microsoft Web page:

Understanding the Built-In User and Group Accounts in IIS 7.0

For more information about how to help secure files with NTFS permissions, visit the following Microsoft Web page:

Securing Files with NTFS Permissions

Note: In some cases, such as when impersonation is used, you would need to give Write access for the authenticated user context as well.

Disallow Script Permissions on the upload folder

Uploaded content for most Web applications are static content, such as images and documents. Uploaded content is not meant to be content that can be run, such as scripts or executable files. Therefore, it is important not to grant Script Permissions on this folder. Otherwise, users who can upload content can execute scripts in the context of your worker process identity on the server. If your Web application has logic to restrict uploads by file name extensions, you should use this restriction as a secondary measure. You should still make sure that your application's upload directory has script permissions disabled.

To disable script permissions in IIS Manager User Interface (inetmgr) in IIS 5.x and 6.0, follow these steps:

1. Click **Start**, and then click **Run**.
2. Type inetmgr in the **Open** box, and then click **OK**.
3. In the tree view in the navigation pane, select the path of the upload directory of your Web application.
4. Right-click this path, and then click **Properties**.
5. Click the **Directory** tab, and then select **None** in the **Execute Permissions** list.

For more information about how to how to set IIS permissions for specific objects, click the following article number to view the article in the Microsoft Knowledge Base:

324068 How to set IIS permissions for specific objects

Alternatively, you can disable script permissions by using metabase configuration in IIS 6.0 by setting **AccessFlags** property's **AccessScript** flag to **False** at the upload directory level. For more information and for sample scripts that can be changed for this use, visit the following Microsoft Web page:

AccessFlags Metabase Property (IIS 6.0)

To disable script permissions in configuration for IIS 7.0 and later versions, you have to set the **accessPolicy** flag on the **handlers** section not to have the **Script** value.

For more information about how to do this in IIS 7.0 and later versions, visit the following Microsoft Web page:

Configure Request Restrictions for a Handler Mapping

Note Make sure that you read the **Script** value for the access flags.

For more information about how to set permissions, visit the following Microsoft Web page:

Securing Sites with Web Site Permissions

The Web application should restrict uploads to authenticated and authorized users only

This gives the server administrator the ability to audit uploads through the Web application. In the case a user is trying malicious activity, it gives the server administrator an easy mechanism to keep the application functional while blocking out users who are trying malicious activity. When users can upload scripts and execute them through the Web application, authentication should be required and the IIS application pool identity hosting the Web application should not be an Administrative account.

Advanced: Uploading Large Files on Windows 2003 Server

In this article we will show you how to upload large files on Windows 2003 server.

For IIS6.0 users, the **AspMaxRequestEntityAllowed** property specifies the maximum number of bytes allowed in the entity body of an ASP request. If a Content-Length header is present and specifies an amount of data greater than the value of **AspMaxRequestEntityAllowed**, IIS returns a 403 error response.

This property is related in function to **MaxRequestEntityAllowed**, but is specific to ASP request. Whereas you might set the **MaxRequestEntityAllowed** property to 1 MB at the general World Wide Web Publishing Service (WWW Service) level, you may choose to set **AspMaxRequestEntityAllowed** to a lower value, if you know that your specific ASP applications handle a smaller amount of data.

How to do it

1. **IIS6.0 prevents the upload of files more than +200Kb.** So you need to make some changes in the default IIS settings first.

2. Before you edit the file, be sure to stop the IIS service first or else you won't be able to save the file.

3. Open your **metabase.XML** which is located in **c:\Windows\System32\Inetsrv** find the line "**AspMaxRequestEntityAllowed**" and change it to "1073741824". This is 1GB - of course you can enter another value to suite your needs.

Advanced: Uploading Large Files on IIS7 (Windows Vista/Windows 2008)

In this article we will show you how to upload large files on IIS7 (Windows Vista/Windows 2008). In order to be able to upload large files you need to change a special option in IIS called ***maxAllowedContentLength*** to allow uploading larger files.

How to do it

1. Click Start. In the Start Search box, type Notepad. Right-click Notepad, and then click Run as administrator.
   *__Note__ If you are prompted for an administrator password or for a confirmation, type the password, or click Continue.

2. On the File menu, click Open. In the File name box, type **%windir%\system32\inetsrv\config\applicationhost.config**, and then click Open.

3. In the **ApplicationHost.config** file, locate the **<requestLimits>** node.

4. Remove the **maxAllowedContentLength** property. Or, add a value that matches the size of the Content-Length header that the client sends as part of the request. By default, the value of the **maxAllowedContentLength** property is 30000000.

For example, modify the following configuration data inside the **<requestFiltering>** section.

```
<requestLimits maxAllowedContentLength =“<length>“ />
```

5. Save the **ApplicationHost.config** file.

For more information see: http://support.microsoft.com/kb/942074/

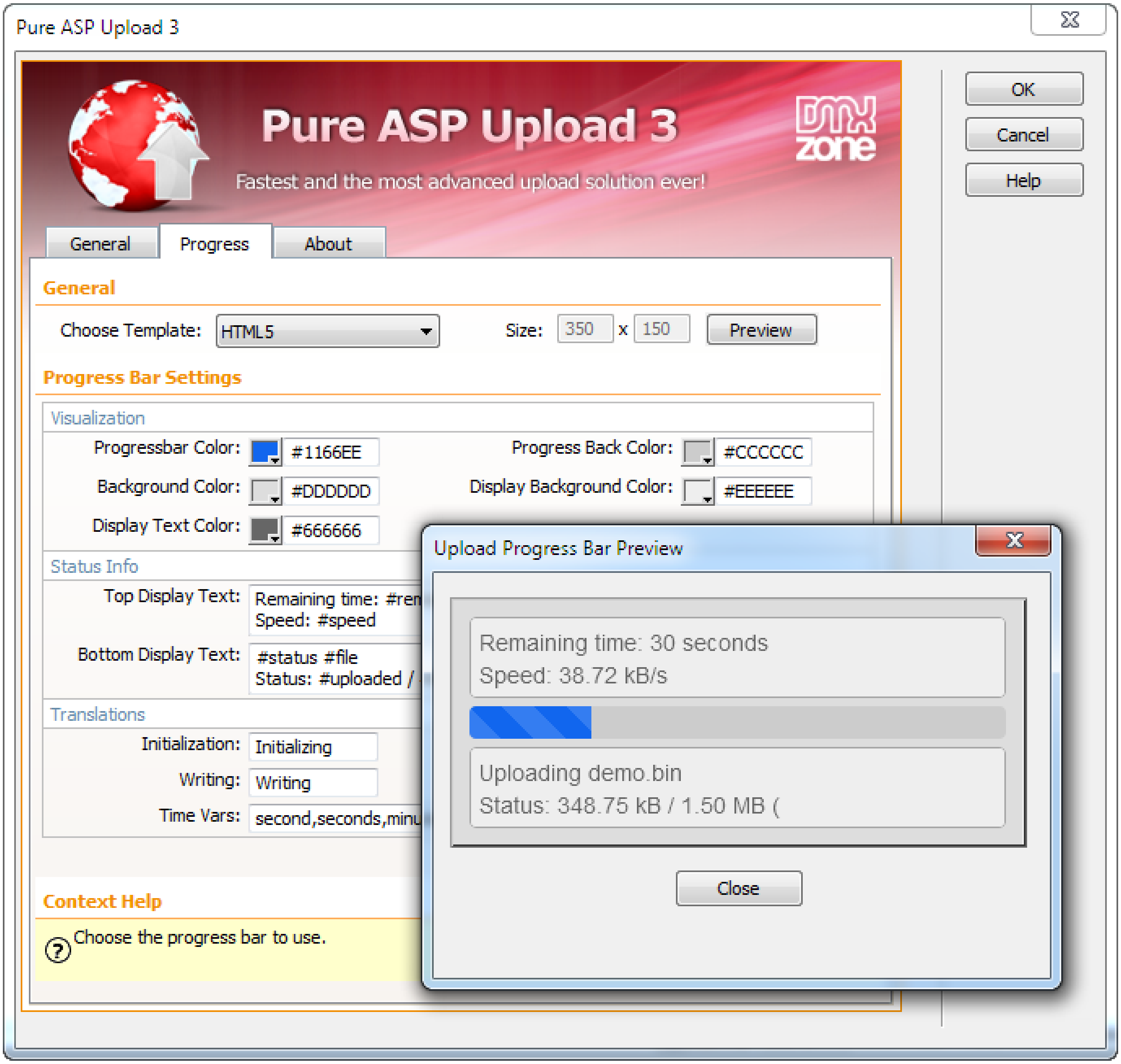
More info about content filtering in IIS:

http://learn.iis.net/page.aspx/143/how-to-use-request-filtering/

http://www.iis.net/ConfigReference/system.webServer/security/requestFiltering/requestLimits

Reference: Customizable Progress Bar

The build-in Dreamweaver menu enables you to select multiple predefined Progress bars and customize them. You can preview your progress bar design directly in Dreamweaver by clicking the "preview" button.


Pure ASP Upload 3
Pure ASP Upload 3
Fastest and the most advanced upload solution ever!
OK
Cancel
Help
General
Progress
About
Preview
Upload Progress Bar Preview
Close

General

Choose Template: HTML5 Size: 350 x 150

Progress Bar Settings

Visualization

Progressbar Color: #1166EE Progress Back Color: #CCCCCC

Background Color: #DDDDDD Display Background Color: #EEEEEE

Display Text Color: #666666

Status Info

Top Display Text: Remaining time: #rem
Speed: #speed

Bottom Display Text: #status #file
Status: #uploaded /

Translations

Initialization: Initializing

Writing: Writing

Time Vars: second,seconds,minu

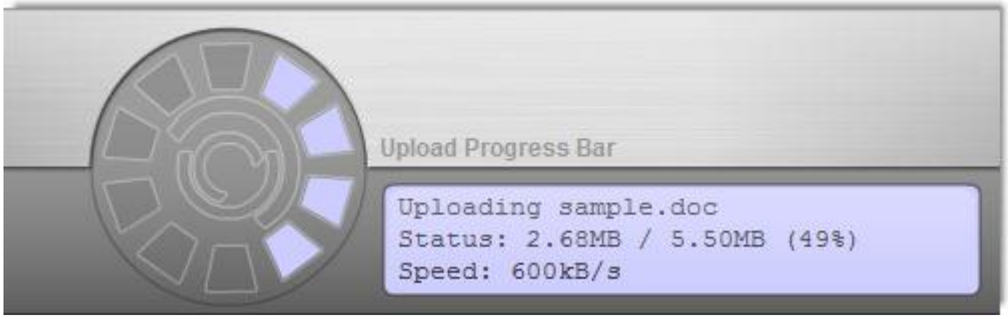
Context Help

Choose the progress bar to use.

Remaining time: 30 seconds
Speed: 38.72 kB/s

Uploading demo.bin
Status: 348.75 kB / 1.50 MB (

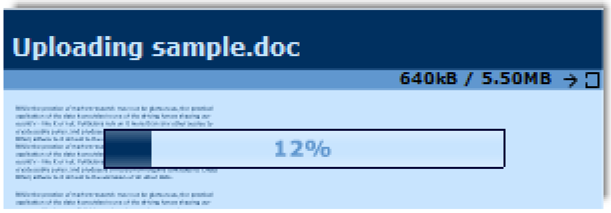
Circle



Upload Progress Bar

```
Uploading sample.doc
Status: 2.68MB / 5.50MB (49%)
Speed: 600kB/s
```

Classic Pure Upload 2

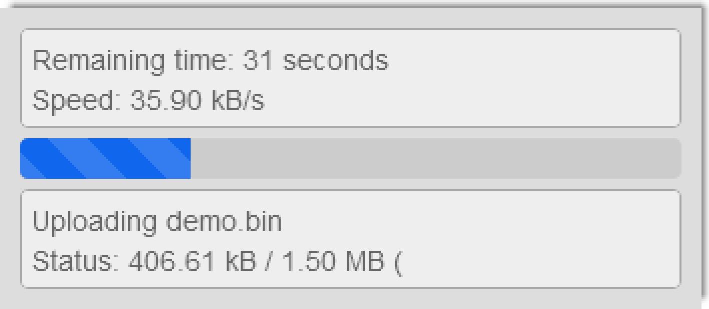


Uploading sample.doc

640kB / 5.50MB

12%

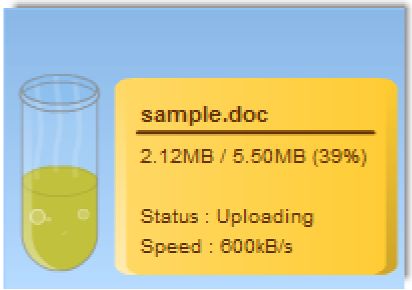
HTML5



Remaining time: 31 seconds
Speed: 35.90 kB/s

Uploading demo.bin
Status: 406.61 kB / 1.50 MB (

Laboratory

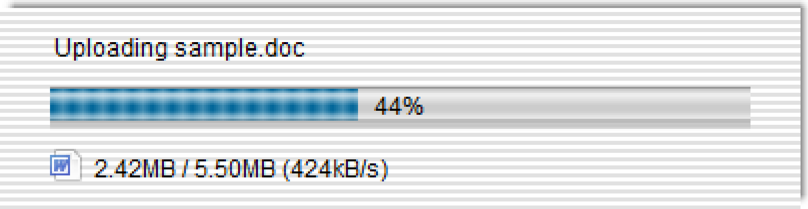


sample.doc

2.12MB / 5.50MB (39%)

Status : Uploading
Speed : 600kB/s

Mac Look

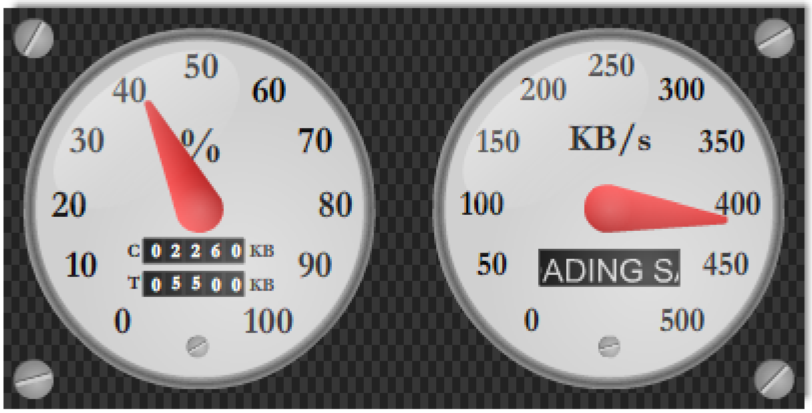


Uploading sample.doc

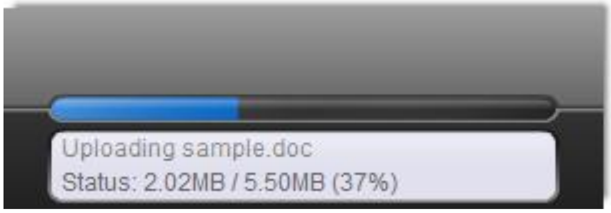
44%

2.42MB / 5.50MB (424kB/s)

Speedometer

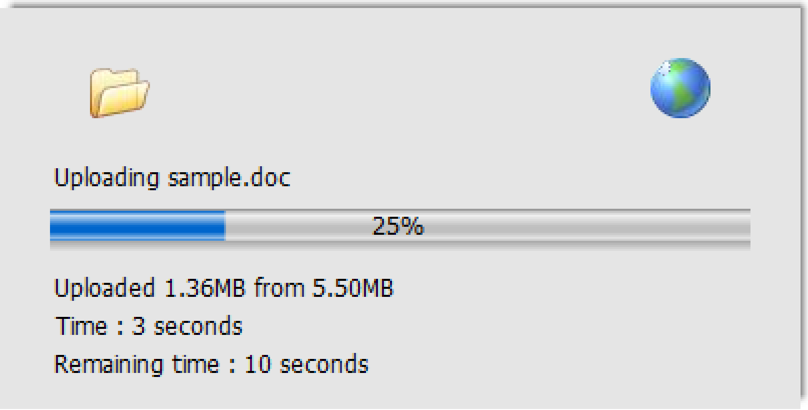


Vista Look



Uploading sample.doc
Status: 2.02MB / 5.50MB (37%)

Windows Look



Uploading sample.doc

25%

Uploaded 1.36MB from 5.50MB
Time : 3 seconds
Remaining time : 10 seconds

Video: Uploading Files with Pure ASP Upload 3

With this **movie** we will show you how to upload files using **Pure ASP Upload 3**.

Video: Using Pure ASP Upload 3 with Insert Record

With this **movie** we will show you how to upload images and save their names, width and height into a database.

Video: Using Pure ASP Upload 3 with Update Record

With this **movie** will show you how to use **Pure ASP Upload 3** with Update Record server behavior.

Video: Uploading Files to a Dynamic Folder

With this **movie** we will show you how to upload files to a Dynamic folder using **Pure ASP Upload 3**.