

# Create upload file with VB.NET

Uploading files is a common feature used by users to upload documents to server-based applications. This is an essential part of many applications from basic to more complex. In this article, I will show you how to add Upload File to ASP.NET applications and create it in VB.NET.

Uploading files is a common feature used by users to upload documents to server-based applications. This is an essential part of many applications from basic to more complex. In this article, I will show you how to add Upload File to ASP.NET applications and create it in VB.NET. Through the VB.NET code example of this document, you can create a function similar to C # (if that's the language you normally use) by changing a bit in the code.

Depending on the different applications, users will have many different options on the website. Some systems such as document management system, content management system, request management system . will allow users to upload different documents.

In the example of this article, you will have to create a simple ASP.NET web application to allow uploading a file to the server.

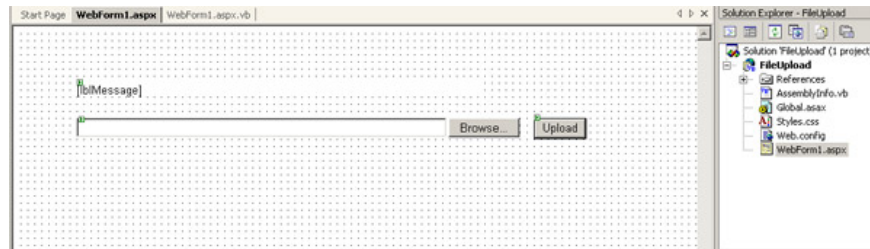
Open Visual Studio.Net and create a new Project with the following settings:

1. **Project Type** : Visual C # Projects
2. **Templates** : ASP.NET Web Application
3. **Location** : http:/// localhost / FileUpload

On the web form:

1. Click Toolbox, select the HTML section, navigate to the File Field control and drag and drop it onto the form.
2. Right-click the control and set " **Run as Server Control** ".
3. Change the control's Name attribute to ' **File1** '.
4. Go to Toolbox again, select Web Forms section, find a Button, and drag it onto the form.
5. Set Text property to ' **Upload** ' and ID to " **cmdUpload** ".

The screen will display as follows:



Add the following code to the form definition of the .aspx file:

```
encType = "multipart / form-data"
```

As a result, the entire form tag will look like this:

```
method = "post"
encType = "multipart / form-data"
runat = "server">
```

Add the following code to the description of the .vb file:

```
Dim sFileDir As String = "C:"
Dim lMaxFileSize Long = 4096
```

Remember that the value above will be editable depending on the application you use. You can also make them dynamic and the application will read these values ??from a database or from an XML file.

Add the following code to the top of the page .vb:

```
Imports System.IO
```

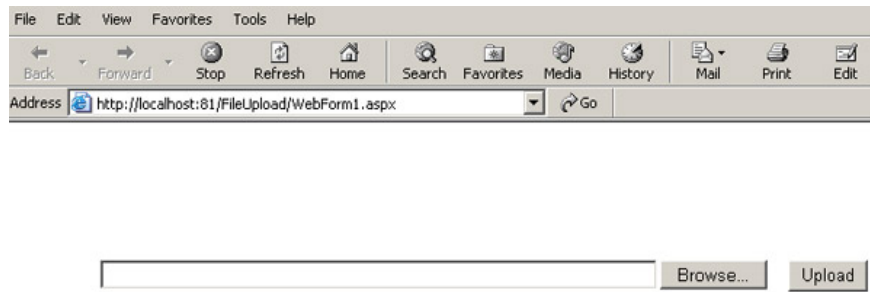
Add the following .vb code page:

```
Private Sub DeleteFile (ByVal strFileName As String) If strFileName.Trim (). Length > 0 Then File.Delete (strFileName) End Sub
```

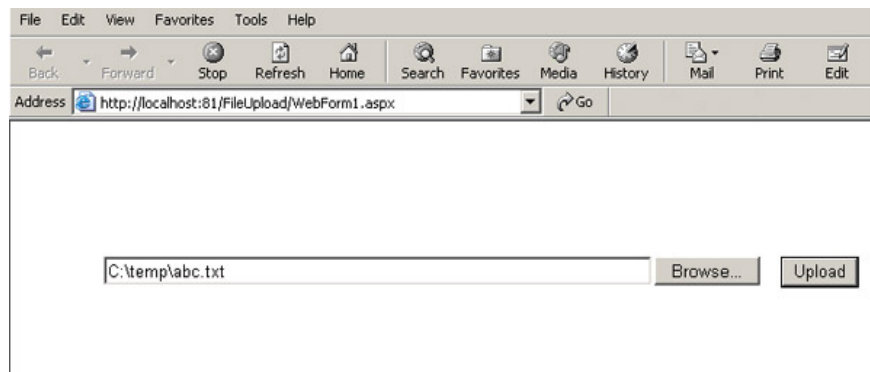
Add the following code to the .vb file:

```
Private Sub cmdUpload_Click (ByVal sender As System.Object, _ ByVal e As System.EventArgs) Handles cmdUpload.Click 'handle that file ?ã ???c ch?n và nó là t?p tin h?p l? If (Not File1.PostedFile Is Nothing) _ And (File1.PostedFile.ContentLength > 0) Then 'determine file name Dim sFileName As String = _ System.IO.Path.GetFileName (File1.PostedFile.FileName) Try If File1.PostedFile.ContentLength = lMaxFileSize Then 'save file on disk File1.SaveAs (sFileDir + sFileName + File1.PostedFile.Extension) lblMessage.Visible = True lblMessage.Text = "File:" + sFileDir + sFileName + File1.PostedFile.Extension Else 'reject file lblMessage.Visible = True lblMessage.Text = "File Size if Over " + lMaxFileSize + " Bytes" End If Catch exc As Exception 'in case c?a l? lblMessage.Visible = True lblMessage.Text = "An Error Occured. Please Try Again" End Try Else lblMessage.Visible = True lblMessage.Text = "Nothing to upload. Please Browse" End If
```

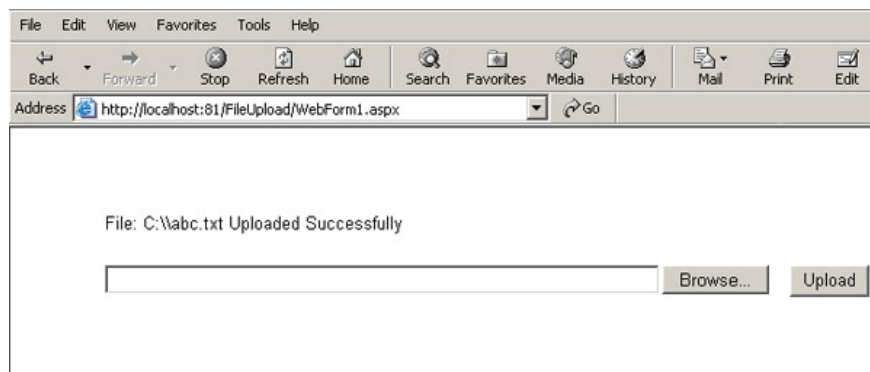
Press F5 to compile and run the project. The interface screen will look like this:



Click Browse and select a file to upload. The interface screen is as follows:



Click Upload. You will receive a message displayed above:



You should check to see if the file has been copied to the directory defined in the code.

## Working

The value of `sFileDir` and `IMaxFileSize` is written in the fixed code above instead of being called from the database or XML configuration file. `sFileDir` specifies the location on the server so that the upload file can be saved. `IMaxFileSize` specifies the maximum file size for uploading.

The `DeleteFile` procedure is used to delete files that have been copied to the server. As part of the normal cleanup process, after the upload file has determined the location, the file will be put into the database or somewhere on the server depending on the needs of the application. In this example, the file should not be

copied to another location because you cannot call this procedure unless an error occurs and need to transfer the file. DeleteFile can be called after the file has been moved to the database or to another location to intentionally remove redundant files. It accepts a full name (directory name and file name) as an argument and needs to be verified that the file is actually existent and the length of the argument is greater than 0. Then it will try to delete file using FileInfo object.

When the user clicks on cmdUpload, you must first check if the file already exists. If the file already exists, you decide the file name without the directory (File1.PostedFile.FileName attribute stores the location and name of the file on the client) by using System.IO.Path.GetFileName. You will then have to verify that the file size is not larger than the maximum allowed capacity. Then, save the file to the location specified on the server using File1.PostedFile.SaveAs method and randomly put the directory and file name on it. When the file is saved, please give the user a confirmation message that the file has been uploaded successfully. If an error occurs, you must delete the file and display an error message in the lblMessage label.

**Note:** When files are uploading, remember that ASP.NET limits the file size for uploading to 4MB (4096 KB). If you try to upload a larger file, you may encounter an error message. You can change this setting by resetting maxRequestLength in the httpRuntime element of the Machine.config file.

You finished reading the article "**Create upload file with VB.NET**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.