

Librairie de Classes pour la connexion FTP en ASP.NET

(sous Visual Studio.NET et en VB.NET)

Le transfert de données d'une application ASP.NET vers un serveur FTP est souvent nécessaire dans le cadre d'applications distribuées ou d'interconnexion de différents types de systèmes (Windows, Unix, Mainframe, ...).

Objectif

Dans le cadre de système d'information hétérogène le transfert de fichiers entre Systèmes différents se fait couramment en FTP.

Passage de données fichiers par exemple :

- Windows vers Unix
- Windows vers AS400
- Windows vers Serveur Web déporté (cas de site Hébergé)

Imaginons le cadre d'une entreprise avec un portail Intranet et un Site Web public, on souhaite faire une gestion sur l'Intranet des données publiées sur le serveur Internet, le transfert peut se faire alors de 2 façons :

- Webservice d'écoute sur le serveur Web qui attend de recevoir un message suivi du transfert des données
- Transfert des fichiers directement par FTP du serveur Intranet vers le serveur WEB

On peut même associer les 2, le Webservice pour les données qui seront stockées en Base de Données et le FTP pour le transfert des données Fichiers (Images, PDF, fichiers DOC, ...).

Pré-Requis

Les classes ajoutées dans cette librairies ne sont pas de moi et de ce fait je vous fourni le lien afin de récupérer les dernières versions ou de remercier comme il se doit les créateurs.

Version VB.NET de Pieter Philippaerts (<http://www.allapi.net/classlib/class.php?id=23>)

Version C# de Pieter Philippaerts (<http://www.allapi.net/classlib/class.php?id=24>)

Composantes

Cette librairie est composée de plusieurs classes, les classes sont celles disponibles sur le site précité (en VB.NET) :

- **DataConnection** : Classe Permettant de gérer la connexion et le transfert de Données.
- **FileItem** : Classe Permettant de gérer les infos sur les fichiers.
- **FtpConnection** : Classe Permettant de gérer la connexion au Serveur FTP.

On trouve aussi dans cette classe le fichier d'exemple fourni par l'auteur pour travailler avec ces Classes :

- **TestApp.vb**

Les fichiers sont ceux d'origine et rien n'y a été modifié, sauf le rajout du cartouche pour Visual SourceSafe du type :

```
*****
' $Archive: $
' $Author: $
' $Date: $ $Revision: $
' Description : Classe Permettant de gérer la connexion
' au Serveur FTP
' *****
```

Cet entête permet de laisser la gestion des paramètres avec un \$ à **SourceSafe**.

Pour cette partie, tout est expliqué ici :

AJOUTER DES ENTÊTES DANS UNE SAUVEGARDE SUR BASE VISUAL SOURCESAFE (<http://www.aspfr.com/article.aspx?Val=435>)

Ainsi, à partir de ces classes (non modifiées), j'ai créé un projet de type Bibliothèque de classes qui vous fourni la DLL directement utilisable dans votre solution. Celle-ci est téléchargeable ici directement :

Librairie de Classes pour transfert FTP
(http://fromelard.free.fr/Scripts/ASPNET_FTPConnexion)

Utilisation dans votre Solution

Après avoir référencé ce projet dans votre solution (Voir : http://www.asp-php.net/tutorial/asp.net/solution_globale.php), il vous suffit alors d'ajouter une fonction de ce type dans votre propre projet.

```
' Methode d'utilisation dans votre projet ASP.NET
' -----
Private Function UploadFichier(ByVal CheminFichier As String, ByVal nomfichier As String) As Boolean
' Prend le fichier du chemin transmis et l'Upload sur le serveur de voulu
Dim Envoie As New FtpConnection()

' Parametres de la connexion

' Login et Mot de passe du compte FTP utilisé
Envoie.Username = UserFTPServer
Envoie.Password = passwdFTPServer

' Adresse du Serveur FTP (IP ou URL)
Envoie.Hostname = HostFTPServer

' Si autre port que le 21 (par défaut)
Envoie.Port = PortFTPServer

' Si besoin de passer en PASV il faut le mettre à true
Envoie.Passive = False

'connexion avec le serveur
Envoie.Connect()

If Envoie.IsConnected Then
' Changement de répertoire courant si besoin
Envoie.ChangeWorkDir(RemotePathFTPServer)

' Envoie le fichier transmis a partir de son répertoire courant
' vers le fichier de destination en mode ASCII (texte)
Envoie.UploadFile(CheminFichier, nomfichier, FtpConnection.StreamModes.ASCII)
Envoie.Disconnect()
Return True
Else
Return False
End If
End Function

' -----
' Utilisable de la facon suivante :

Dim ResulUpload As Boolean = False
ResulUpload = UploadFichier("C:\MonFichier.txt", "NomDestination.txt")
```

Conclusion

Cet article n'a pour but que de vous montrer que le développement en ASP.NET ne veut plus dire (comme anciennement en ASP) faire de simple pages WEB, mais vraiment développer des applications complètes mélangeant une partie WEB avec des traitements en Batch (par exemple) pour mettre à jour des serveurs Unix via ce process FTP (par exemple).

F___