

.. 0 · 0 · 0 0
République Algérienne Démocratique et Populaire
0 · 0 · 0 00

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N°Réf :.....

Centre Universitaire de Mila

Institut des sciences et de la technologie Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de licence
En:Informatique générale**

Thème

**Réalisation d'un client ftp compatible au
serveur Filezilla server (java)**

Préparé par :

- *Zenir Oussama*
- *Chaker Amine*
- *Chelihi Hassan*

Encadré par:TALAI MERIEM

Année universitaire :2013/2014

Sommaire

<i>Introduction générale</i>	7
------------------------------------	---

Chapitre 1 : Le protocole FTP

1. présentation de FTP.....	8
2. Types de connexions FTP.....	9
3. Étapes d'une connexion FTP.....	10
4. Les modes de FTP.....	11
4.1 MODE Actif.....	12
4.2 MODE passif.....	13
4.3 Connexion FTP entre serveurs.....	14
5. Les commandes FTP externe (utilisateur).....	15
6. Les commandes FTP interne	17
7. Les réponses FTP.....	22
8. paramètres de transfert des fichiers.....	23
9. conclusion.....	24

Chapitre 2 : Installation et configuration de fileZilla server

1. Présentation.....	26
2. Installation de FileZilla Server	26
3. Interface d'administration	28
4. Les paramètres généraux de FileZilla Server	30
4.1. General settings	31
4.1.1 General settings / Passive mode settings	33
4.1.2 General settings / Security settings	33
4.1.3 General settings / Admin interface settings	34
4.1.4 General settings / Logging	36
4.1.5 General settings / GSS Settings	38
4.1.6 General settings / Speed Limits	38
4.1.7 General settings / Filetransfer compression	40
4.1.8 General settings / SSL/TLS settings	41
4.1.9 General settings / Autoban	43
5. La première connexion FTP	44

6. Configuration d'un serveur FTP.....	46
7. Sécurité des serveurs FTP.....	47
8.conclusion.....	47

Chapitre3 :Implèmentation de client ftp

1.Introduction.....	49
2. Environnement de développement.....	49
2.1 Moyen matériel.....	49
2.2 Moyen logiciel	49
3.Programmation du protocole FTP en JAVA.....	50
3.1 Socket et server Socket.....	50
3.1.1 Socket.....	50
3.1.2 Server Socket.....	50
3.2 Les flus d'entrée/sortie.....	50
3.2.1 les flux d'entrer.....	50
3.2.2 les flux de sortie.....	51
3.3 Fonctionnement de client.....	51
4.Présentation de client FTP.....	52
4.1. Barre de menu.....	53
4. 2. Barre de connexion.....	53
4.3. Les panneaux.....	53
4.4 Le button connexion.....	54
4.5 Le button déconnexion.....	55
5. Les opération de client.....	56
5.1 Création de dossier.....	56
5.2 Téléchargement/Upload.....	57
5.2.1 Téléchargement.....	57
5.2.2 Upload.....	57
6. Conclusion.....	58
Conclusion generale.....	59

Liste des figures

Chapitre 1 : Le protocole FTP

Figure 1 : Connexions de contrôle et de données pour le transfert FTP...	08
Figure 2 : Enchaînement du mode actif de FTP	13
Figure 3 : Enchaînement du mode passif de FTP.....	14
Figure 4 : Model d'usage ftp (cas de 2 serveurs)	15

Chapitre 2 : Installation et configuration de filleZilla server

Figure 5 : l'icône FilleZilla.....	26
Figure 6 : le menu démarrer.....	27
Figure 7 : FilleZilla server FTP server.....	28
Figure 8 : FilleZilla server FTP server.....	28
Figure 9 : l'interface d'information de serveur.....	28
Figure 10 : l'interface d'administration de serveur.....	29
Figure 11 : le menu détaillé.....	29
Figure 12 : le menu server.....	30
Figure 13 : le menu edit.....	30
Figure 14 : le menu edit.....	30
Figure 15 : FilleZilla server options.....	31
Figure 16 : IP bindings.....	32
Figure 17 : IP filter.....	32
Figure 18 : passive mode settings.....	33
Figure 19 : security settings.....	34
Figure 20 : admin interface settings.....	35
Figure 21 : l'interface logging.....	36
Figure 22 : répertoire logs.....	37
Figure 23 : l'interface GCC Settings.....	38

Figure 24 : l' interface speed limits.....	39
Figure 25 :l'interfacefiletransfer compression.....	41
Figure 26 :l'interface SSL/TLS settings.....	42
Figure 27 : l'interface generatecertificate.....	43
Figure 28 : l'interface Autoban.....	44
Figure 29 : l'interface d'administration du serveur FTP.....	45

Chapitre 3 : implémentation de client ftp

Figure 30 : interface client.....	53
Figure 31 : connexion au serveur.....	54
Figure 32 : message d'information.....	55
Figure 33 : déconnexion au serveur.....	55
Figure 34 : créer un dossier.....	56
Figure 35 : entrer le nom de dossier.....	56
Figure 36 : boite de dialogue.....	57
Figure 37 : envoyer le fichier.....	58

Introduction générale

L'Internet est aujourd'hui le premier réseau mondial accessible à tout type d'utilisateurs et d'entreprises. De même, les applications Internet sont très diverses et c'est précisément ce qui en fait sa force. Néanmoins, le World Wide Web est l'application la plus utilisée. C'est une technologie qui permet à partir d'un logiciel client appelé navigateur (ou browser) d'accéder facilement à des documents stockés sur un serveur connecté à l'Internet.

Le modèle applicatif d'Internet et celui du client/serveur, ou un programme client permet à un utilisateur de soumettre des requêtes à un serveur, de visualiser le résultat, le serveur étant un programme qui tourne sur un ordinateur dans le but de répondre à des requêtes de logiciel client qui tournent sur d'autres ordinateurs. Un document est l'unité fournie par le serveur en réponse à une requête du client.

Sur l'Internet, de nombreux fichiers peuvent être téléchargés, le moyen le plus sûr de téléporter des fichiers est d'utiliser FTP qui dérive des services à partir du protocole TCP. Les données sont délivrées dans l'ordre de leur émission.

Notre projet consiste à la réalisation d'un client FTP compatible avec le serveur FileZilla Server qui permet de créer un serveur et la mise à disposition de fichier que ce soit sur un réseau local, un réseau d'entreprise ou sur Internet en accès libre ou privé pour effectuer l'émission entre le serveur (créé par FileZilla) et le client.

Pour la réalisation effective de ce client FTP, nous avons élaboré une étude théorique relative aux notions utilisées. Les chapitres de ce travail sont présentés selon le plan suivant :

Dans le premier chapitre on a donné une présentation générale du protocole FTP.

Le deuxième chapitre présente le FileZilla server et les différentes étapes pour installer un serveur et configurer ensuite les paramètres généraux de ce serveur.

Quant au dernier chapitre, il implémente un client FTP compatible avec FileZilla server.

Chapitre 1

Le protocole FTP

1. Présentation de FTP

FTP (File Transfer Protocol) ou protocole de transfert de fichiers, est un protocole de communication qui opère au niveau de la couche application du modèle OSI. Il définit la façon selon laquelle des données doivent être échangées sur un réseau TCP /IP.

Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur. Ce mécanisme de copie est souvent utilisé pour alimenter un site web hébergé chez un tiers.

FTP obéit au modèle client-serveur : Le serveur est un ordinateur sur lequel fonctionne un logiciel lui-même appelé serveur FTP, qui rend publique une arborescence de fichiers similaire à un system de fichiers WINDOWS. Pour accéder à un serveur FTP, on utilise un logiciel client FTP à travers son interface graphique ou via des lignes de commande qu'il identifie.

L'évolution du protocole FTP, comme tout autre protocole utilisé sur Internet, passe par des publications sous forme de RFC (Request For Comments). Les RFCs sont un ensemble de documents de référence auprès de la Communauté Internet et qui décrivent, spécifient, aident à l'implémentation, standardisent et débattent de la majorité des normes, standards, technologies et protocoles liés à Internet et aux réseaux en général.

FTP est issu de la RFC 114 créée le 16 avril 1971. Cette spécification fut remplacée par la RFC 765 en juin 1980. Elle fut elle-même rendue obsolète par la RFC 959 en octobre 1985, version finale de la spécification. Plusieurs RFC viennent compléter cette spécification, comme la RFC 2228 de juin 1997 pour l'ajout d'extensions de sécurité ou la RFC 2428 de septembre 1998 qui ajoute la prise en charge du protocole IPv6 et définit un nouveau type de mode passif.

L'utilisation du protocole FTP assure :

- un partage de fichiers entre machines distantes
- une indépendance aux systèmes de fichiers des machines clients et serveur
- un transfert des données de manière efficace sans altérer les fichiers ou perdre des informations

2. Types de connexions FTP

Par convention, deux ports sont attribués (wellknown ports) pour les connexions FTP : le port 21 pour la transmission des commandes et le port 20 pour le transfert des données (du côté du serveur).

En effet, lors d'une transmission via FTP, deux canaux de transmission sont ouverts :

- Un canal pour les commandes (canal de contrôle)
- Un canal pour les données

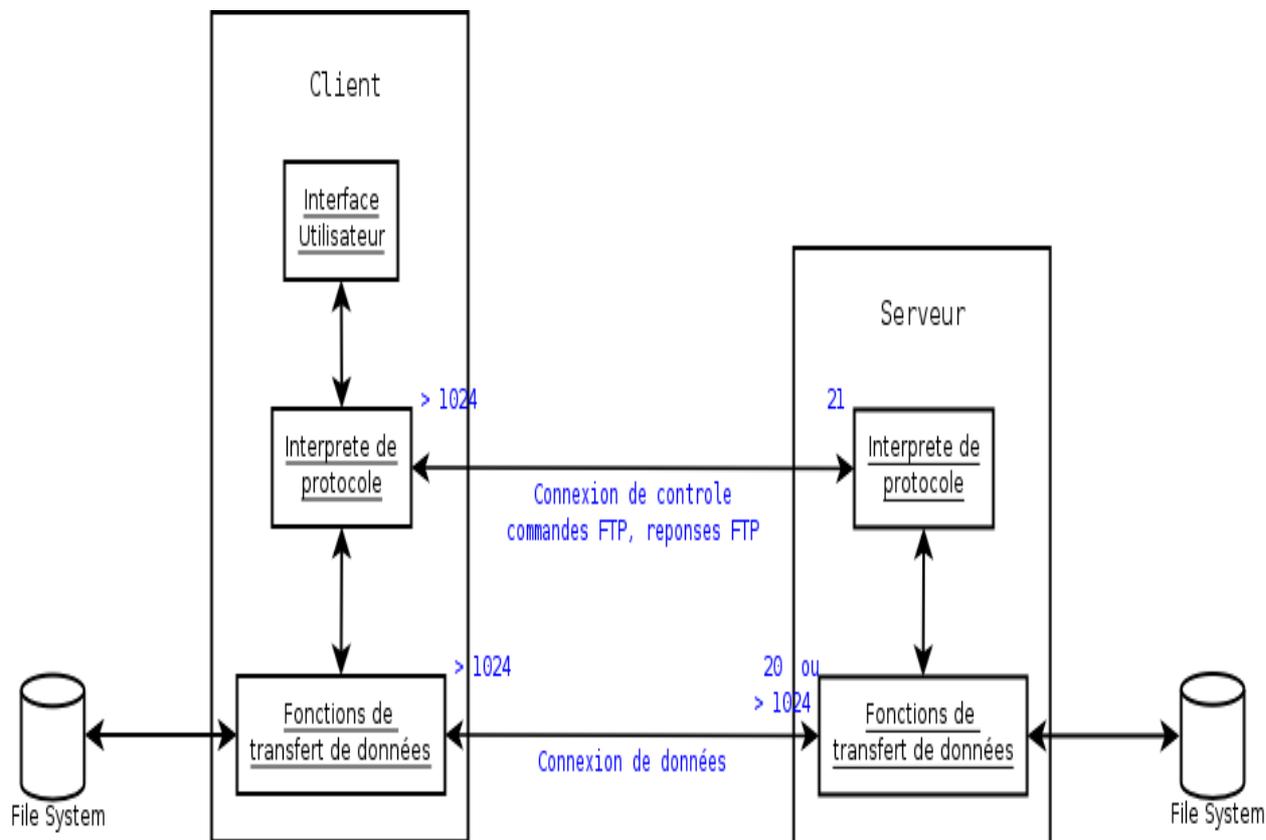


Figure 1 : Connexions de contrôle et de données pour le transfert FTP [1]

Pour gérer ces deux types de connexion, le client comme le serveur dispose de deux processus différents (Figure 1) :

- Le DTP (Data Transfer Process) est le processus chargé d'établir la connexion et de gérer le canal de données. Le DTP côté serveur est appelé SERVER-DTP, le DTP côté client est appelé USER-DTP
- Le PI (Protocol Interpreter) est l'interpréteur de protocole permettant de commander le DTP à l'aide des commandes transmises sur le canal de contrôle. Il est différent sur le client et sur le serveur :

Le SERVER-PI est chargé d'écouter les commandes provenant d'un USER-PI sur le canal de contrôle sur un port donné, d'établir la connexion pour le canal de contrôle, de recevoir sur celui-ci les commandes FTP du USER-PI, d'y répondre et de piloter le SERVER-DTP

Quant au USER-PI, il est chargé d'établir la connexion avec le serveur FTP, d'envoyer les commandes FTP, de recevoir les réponses du SERVER-PI et de contrôler le USER-DTP si besoin.

3. Étapes d'une connexion FTP

Comme pour tout autre protocole, il y a trois étapes principales pour définir une bonne session entre un utilisateur et le serveur, à savoir :

- Établissement de la connexion
- Transfert des données
- Déconnexion de la session

Nous explorons ces étapes à travers les commandes manipulées par FTP. Cependant, dans l'étape de transfert des données nous ne détaillons que l'authentification du client auprès du serveur puisque le transfert des données s'opère au niveau de la connexion de données selon deux modes détaillés dans la section suivante.

Étape 1 : Établissement de la connexion à la demande du client

Num	Source ad	Port	Destination ad	Port	Flags	Long	Données
1	192.168.0.204	1387	195.98.246.30	21	SYN	0	Requête client connexion TCP 21
2	195.98.246.30	21	192.168.0.204	1387	ACK SYN	0	Serveur Ok
3	192.168.0.204	1387	195.98.246.30	21	ACK	0	Client Ok

Le client envoie un paquet SYN, le serveur répond par SYN/ACK et le client confirme par ACK. Ceci est une séquence très classique d'établissement de connexion TCP. Ensuite le serveur envoie ou non une bannière d'accueil ; cela dépend de la configuration du serveur. Tout ceci se passe sur le port de contrôle FTP (TCP 21).

Étape 2 : Transfert des données

Num	Source ad	Port	Destination ad	Port	Flags	Long	Données
5	192.168.0.204	1387	195.98.246.30	21	ACK PUSH	16	Client commande USER
6	195.98.246.30	21	192.168.0.204	1387	ACK PUSH	70	Serveur accepte client

7	192.168.0.204	1387	195.98.246.30	21	ACK PUSH	5	Client commande PWD
8	195.98.246.30	21	192.168.0.204	387	ACK PUSH	34	Serveur donne répertoire accueil

Le client envoie la commande USER (paquet 5) avec son nom de connexion, ou anonymous pour une connexion anonyme (ce qui est le cas dans cet exemple). Le serveur répond tout de suite (paquet 6), car il accepte la commande USER seule. D'autres serveurs requièrent l'envoi d'un mot de passe après la commande USER même pour le compte anonyme. Simplement, le mot de passe envoyé n'a pas d'importance. Tout est une question de configuration et de type de serveur. Ici le client envoie tout de même la commande mot de passe (paquet 7) et le serveur répond en indiquant quel est le répertoire courant (paquet 8)

Etape 3 : Quitter la session FTP

La session FTP se termine avec la commande QUIT (paquet 46) suivie des paquets de fin de connexion TCP sur le port de contrôle FTP (TCP 21).

46	192.168.0.204	1387	195.98.246.30	21	ACK PUSH	6	Client QUIT	requête
47	195.98.246.30	21	192.168.0.204	1387	ACK PUSH	99	Serveur	Ok 221-Goodbye e. You uploaded 0 and downloaded 1 kbytes...221 Logout - CPU time spent: 0.00 seconds..

4. Les modes de FTP

FTP présente deux modes de transfert différent selon qui des clients ou serveur qui initie le transfert des données. Il s'agit d'un FTP en mode actif ou en mode passif.

4.1 Mode ACTIF

Historiquement, le mode actif du protocole FTP est le premier à avoir existé. Il n'y avait pas encore de distinction entre les deux modes puisque c'était le seul à être utilisé. C'est à l'apparition du FTP dit "passif" que le mode historique a été nommé "actif" par opposition.

Dans ce mode, le client initie (comme pour toute connexion dans une architecture client/serveur) la connexion sur le serveur FTP (port 21) en partant lui même d'un port $N > 1024$. Le client se met alors à écouter sur le port $N+1$ et transmet ce port au serveur FTP en utilisant la commande PORT. Cette commande se présente comme suit : PORT $n_1, n_2, n_3, n_4, p_1, p_2$ dans laquelle n_1, n_2, n_3, n_4 représentent les quatre octets de l'adresse IP, et $(p_1 * 256 + p_2)$ représente le port de

la deuxième connexion. Le serveur peut alors se connecter sur ce dernier en partant de son port local de données (port 20).

Ainsi, le fonctionnement de FTP du côté du serveur est vu comme suit :

- Il utilise deux ports : le 1^{er} port pour les commandes et contrôles divers (TCP 21). Le 2^{ème} port FTP-DATA pour le transfert des données proprement dites (TCP 20)
- Le serveur FTP écoute sur le port 21 ; l'initiative de la connexion est au client (terminologie C/S) ;
- Le serveur FTP répond depuis son port 21 vers le port client $N > 1024$ (connexion de contrôle) ;
- Le serveur FTP initie la connexion de données depuis son port 20 vers le port client $N+1$.

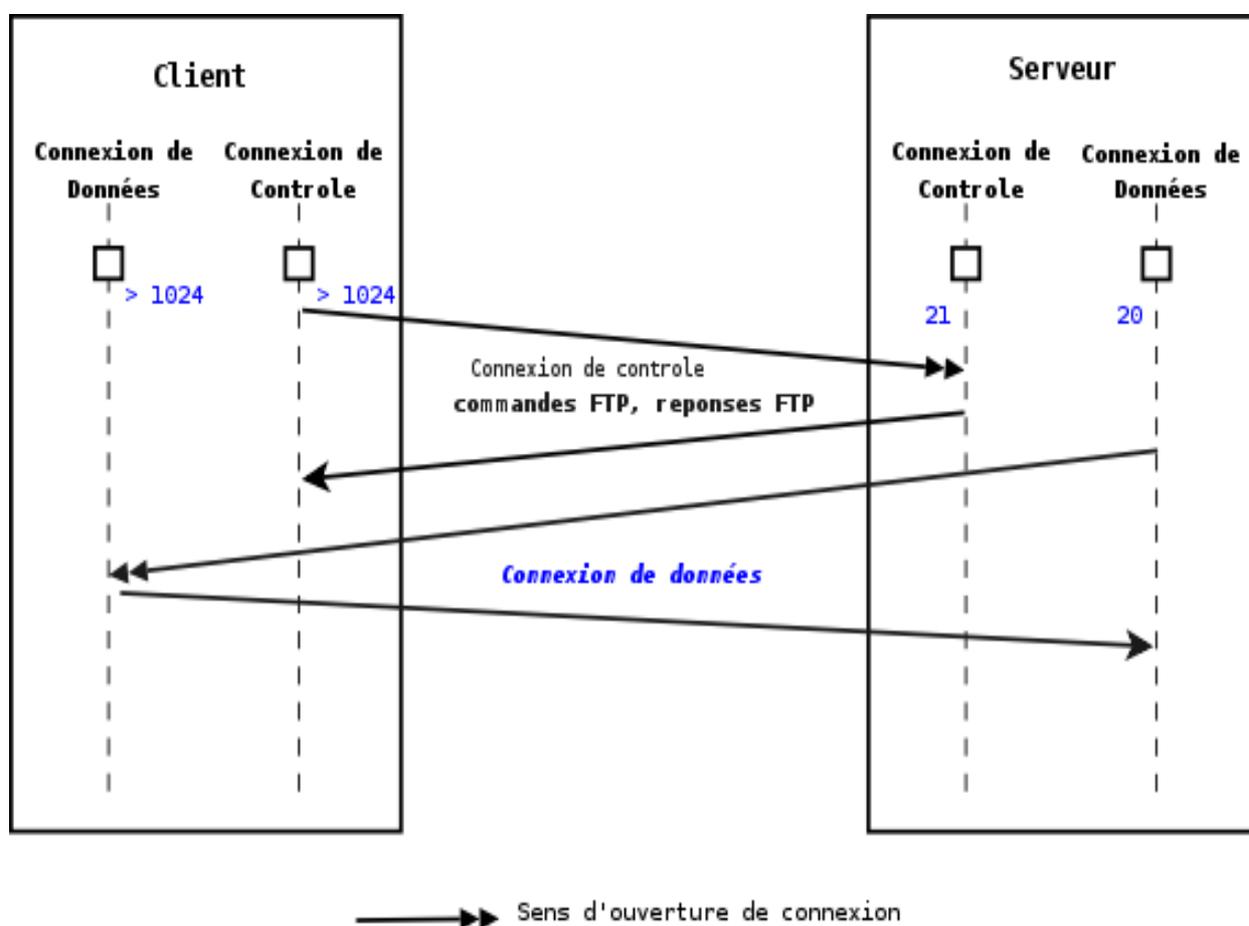


Figure 2 : Enchaînement du mode actif de FTP [2]

4.2 Mode PASSIF

Le fait que sur la connexion de données se soit le serveur qui ait l'initiative, rend le mode actif incompatible avec toutes les organisations qui se protègent (leurs utilisateurs) derrière un firewall. En effet, ce dernier n'autorise le trafic entrant qu'en tant qu'il est réponse à des connexions initiées depuis l'intérieur de l'organisation. Le mode FTP passif représente une solution à ce problème.

Dans ce mode, le client prend l'initiative sur les deux types de connexions (contrôle et données), ce qui résout le problème lié aux firewalls. A l'ouverture de la connexion FTP le client sélectionne deux ports non privilégiés ($N > 1024$ et $N+1$), le premier est utilisé pour la connexion de contrôle (vers le port 21 du serveur FTP), et au lieu d'utiliser la commande **PORT**, le client envoie la commande **PASV**. Le serveur sélectionne alors un port non privilégié ($P > 1024$) et en informe le client, ce dernier initie alors la deuxième connexion (de données) depuis son port $N+1$ vers le port P du serveur.

Du point de vue du serveur FTP, le mode passif suit la séquence suivante :

- Le serveur FTP écoute sur le port 21 ; l'initiative de la connexion est au client (terminologie C/S) ;
- Le serveur FTP répond depuis son port 21 vers le port client $N > 1024$ (connexion de contrôle) ;
- Le serveur FTP accepte les ouvertures de connexions (de données) depuis le port $N+1$ du client vers un port $P > 1024$;
- Le serveur FTP répond depuis son port $P > 1024$ vers le port $N+1$ du client.

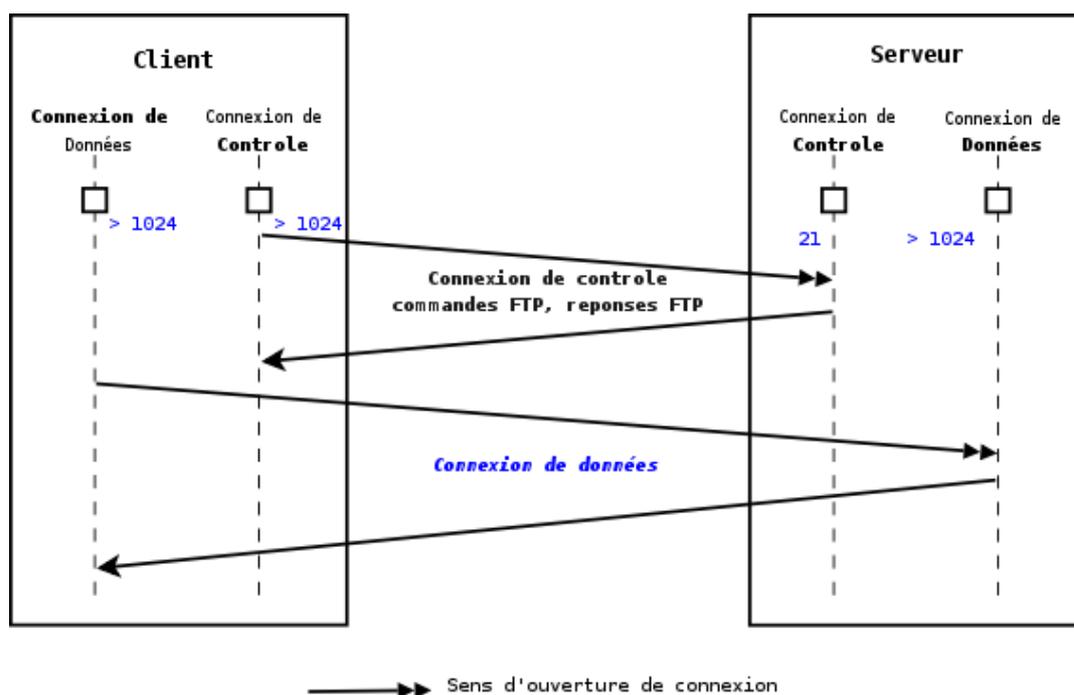


Figure 3: Enchaînement du mode passif de FTP [2]

4.3 Connexion FTP entre serveurs

Il est important de remarquer que, les ports de contrôle et de données étant des canaux séparés, il est possible d'envoyer les commandes à partir d'une machine et de recevoir les données sur une autre. Ainsi, il est par exemple possible de transférer des données entre deux serveurs FTP en passant par un client pour envoyer les instructions de contrôle et en transférant les informations entre deux processus serveurs connectés sur le bon port.

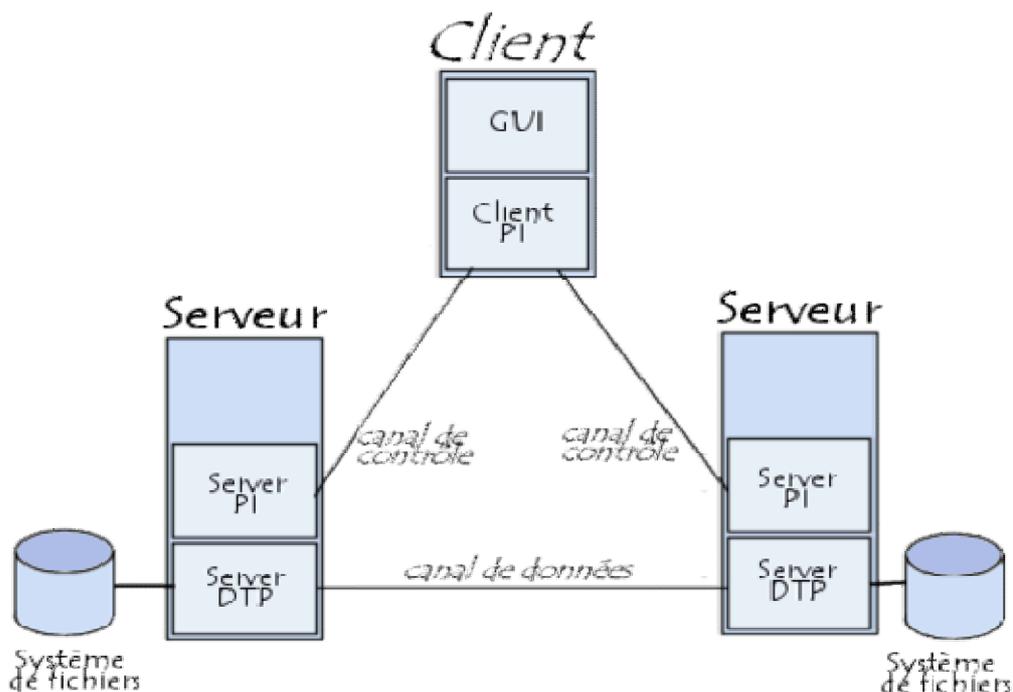


Figure 4 : Model d'usage FTP (cas de 2 serveurs) [3]

Dans cette configuration, le protocole impose que les canaux de contrôle restent ouverts pendant tout le transfert de données. Ainsi un serveur peut arrêter une transmission si le canal de contrôle est coupé lors de la transmission.

5. Les commandes FTP (utilisateur)

Pour accéder au serveur FTP, l'utilisateur sur son poste client, sous Windows comme sous Linux, saisie la ligne de commande suivante :

`ftpadresse_du_serveur` dans une console.

Ou dans les interfaces graphique, comme les navigateurs Web, la forme usuelle est utilisée, à savoir : `ftp://adresse_du_serveur`.

Une fois connecté, l'utilisateur utilise six commandes/fonctions de base, communes à tous les programmes FTP. Il s'agit de : OPEN, CLOSE, CD, DIR, GET, PUT.

Mais il existe d'autres commandes comme : `use`, `mget`, `mput`, `delete`, `mkdir`, `rename`.

La commande	Description
OPEN	La première opération est d'ouvrir (OPEN) un canal de communication

	<p>entre votre ordinateur et le serveur FTP désigné.</p> <p>Exemple</p> <pre>open 87.123.56.92 open 87.123.56.92 5000 (5000 représente le numéro de port)</pre>
CLOSE	Cette commande indique simplement à l'ordinateur éloigné que vous avez terminé et que vous désirez couper le canal de communication.
CD	Commande de changement de répertoire. Comme la plupart des ordinateurs, les serveurs FTP et leur contenu sont conservés dans des répertoires. La commande CD offre le moyen de naviguer dans ces répertoires.
DIR	Cette commande fonctionne de la même manière que dans la version MSDOS. Elle permet d'obtenir la liste du répertoire actuellement sélectionné.
GET	<p>C'est la commande utilisé pour extraire un fichier de l'ordinateur éloigné et le stocker dans votre propre ordinateur. Certains programmes FTP avec fenêtres permettent d'utiliser une commande « GET » implicite en permettant de cliquer-traîner un fichier d'une fenêtre à l'autre. D'autres programmes FTP exigent de taper la commande directement.</p> <p>Exemple :</p> <pre>get stuff.txt get /pub/b/stuff.txt get /pub/c/programs.obsolete/stuff.txt</pre> <p>Il est important de noter que le fichier sera conservé dans le répertoire actuel, quel qu'il soit. C'est pourquoi, l'idéal est de lancer le programme FTP à partir d'un répertoire de téléchargement. Il est également possible de naviguer à travers une interface graphique, entre les répertoires locaux comme pour le gestionnaire de fichiers.</p>
PUT	Commande inverse de « GET », cette commande permet d'envoyer un ou plusieurs fichiers de ordinateur vers un ordinateur éloigné. Cette commande nécessite d'avoir au préalable la permission du propriétaire du système avant de transférer les fichiers

QUIT	cette commande ferme la connexion de contrôle avec le serveur FTP et permet de sortir de l'application du client FTP.
Use	Utiliser la commande « use » seule pour afficher la liste des commandes ou ajouter le -? comme indicateur avec n'importe quelle commande pour obtenir de l'aide sur la syntaxe et l'utilisation de la commande * mget - permet le téléchargement de fichiers multiples – cette commande et celles qui précèdent peuvent être exécutées avec des permissions de lecture
Mput	Permet de télécharger plusieurs fichiers
Delete	Utilisé sous la forme delete<nom de fichier> pour supprimer un répertoire
Mkdir	utilisé sous la forme mkdir<nom du répertoire> pour ajouter un répertoire
Rename	utilisé sous la forme rename<vieux nom de fichier> pour renommer un répertoire

6. Les commandes FTP interne

On distingue trois types de commandes FTP :

- Les commandes de contrôle d'accès ce sont des commandes du system pour vérifier l'accès d'un utilisateur
- Les commandes du paramétrage de transfert
- Les commandes de service FTP

Commande de contrôle d'accès

Commande	Description
----------	-------------

USER	Chaîne de caractères permettant d'identifier l'utilisateur. L'identification de l'utilisateur est nécessaire pour établir une communication sur le canal de données
PASS	Chaîne de caractères spécifiant le mot de passe de l'utilisateur. Cette commande doit être immédiatement précédée de la commande <i>USER</i> . Il revient au client de masquer l'affichage de cette commande pour des raisons de sécurité
ACCT	Chaîne de caractères représentant le compte (account) de l'utilisateur. Cette commande n'est généralement pas nécessaire. Lors de la réponse à l'acceptation du mot de passe, si la réponse est 230 cette phase n'est pas nécessaire, si la réponse est 332, elle l'est
CWD	<i>Change Working Directory</i> : cette commande permet de changer le répertoire courant. Cette commande nécessite le chemin d'accès au répertoire à atteindre comme argument
CDUP	<i>Change to Parent Directory</i> : cette commande permet de remonter au répertoire parent. Elle a été introduite pour remédier aux problèmes de nommage de répertoire parent selon les système (généralement "..")
SMNT	<i>Structure Mount</i> :
REIN	<i>Reinitialize</i> :
QUIT	Commande permettant de terminer la session en cours. Le serveur attend de finir le transfert en cours le cas échéant, puis de fournir une réponse avant de fermer la connexion

Commande de paramètres de transfert

Commande	Description
PORT	Chaîne de caractères permettant de préciser le numéro de port à utiliser
PASV	Commande permettant d'indiquer au serveur DTP de se mettre en attente une connexion sur un port spécifique choisi aléatoirement parmi les ports disponibles. La réponse à cette commande est l'adresse IP de la machine et le port.

TYPE	Cette commande permet de préciser le type de format dans lequel les données seront envoyées
STRU	Caractère Telnet précisant la structure du fichier (F pour <i>File</i> , R pour <i>Record</i> , P pour <i>Page</i>)
MODE	Caractère Telnet précisant le mode de transfert des données (S pour <i>Stream</i> , B pour <i>Block</i> , C pour <i>Compressed</i>)

Commande de service FTP

Commande	Description
RETR	Cette commande (<i>RETRIEVE</i>) demande au serveur DTP une copie du fichier dont le chemin d'accès est passé en paramètre.
STOR	Cette commande (<i>store</i>) demande au serveur DTP d'accepter les données envoyées sur le canal de données et de les stocker dans le fichier portant le nom passé en paramètre. Si le fichier n'existe pas, le serveur le crée, sinon il l'écrase
STOU	Cette commande est identique à la précédente, si ce n'est qu'elle demande au serveur de créer un fichier dont le nom est unique. Le nom du fichier est retourné dans la réponse
APPE	Grâce à cette commande (<i>append</i>) les données envoyées sont concaténées dans le fichier portant le nom passé en paramètre s'il existe déjà, dans le cas contraire il est créé
ALLO	Cette commande (<i>allocate</i>) demande au serveur de prévoir un espace de stockage suffisant pour contenir le fichier dont le nom est passé en argument.
REST	Cette commande (<i>restart</i>) permet de reprendre un transfert là où il s'était arrêté. Pour cela cette commande envoie en paramètre le marqueur représentant la position dans le fichier à laquelle le transfert avait été interrompu. Cette commande doit être immédiatement suivie d'une commande de transfert.
RNFR	Cette commande (<i>renamefrom</i>) permet de renommer un fichier. Elle indique en paramètre le nom du fichier à renommer et doit être immédiatement suivie de la

	commande <i>RNTO</i>
RNTO	Cette commande (<i>rename to</i>) permet de renommer un fichier. Elle indique en paramètre le nom du fichier à renommer et doit être immédiatement précédée de la commande <i>RNFR</i>
ABOR	Cette commande (<i>abort</i>) indique au serveur DTP d'abandonner tous les transferts associés à la commande précédente. Si aucune connexion de données n'est ouverte, le serveur DTP ne fait rien, sinon il la ferme. Le canal de contrôle reste par contre ouvert.
DELE	Cette commande (<i>delete</i>) permet de supprimer le fichier dont le nom est passé en paramètre. Cette commande est irrémédiable, seule une confirmation au niveau du client peut être faite.
RMD	Cette commande (<i>remove directory</i>) permet de supprimer un répertoire. Elle indique en paramètre le nom du répertoire à supprimer
MKD	Cette commande (<i>make directory</i>) permet de créer un répertoire. Elle indique en paramètre le nom du répertoire à créer
PWD	Cette commande (<i>printworking directory</i>) permet de renvoyer le chemin complet du répertoire courant
LIST	Cette commande permet de renvoyer la liste des fichiers et répertoires présents dans le répertoire courant. Cette liste est envoyée sur le DTP passif. Il est possible de passer en paramètre de cette commande un nom de répertoire, le serveur DTP enverra la liste des fichiers dans le répertoire passé en paramètre
NLST	Cette commande (<i>name liste</i>) permet d'envoyer la liste des fichiers et répertoires dans le répertoire courant
SITE	Cette commande (<i>site parameters</i>) permet au serveur de proposer des services spécifiques, non définis dans le protocole FTP
SYST	Cette commande (<i>system</i>) permet d'envoyer des informations sur le serveur distant
STAT	Cette commande (<i>status</i>) permet d'émettre l'état du serveur, par exemple pour connaître la progression d'un transfert en cours. Cette commande accepte en argument un chemin d'accès, elle retourne alors les mêmes informations que LIST

	mais sur le canal de contrôle
HELP	Cette commande permet de connaître l'ensemble des commandes comprises par le serveur. Les informations sont retournées sur le canal de contrôle
NOOP	Cette commande (<i>no operations</i>) sert uniquement à obtenir une commande OK du serveur. Elle peut servir uniquement pour ne pas être déconnecté après un temps d'inactivité trop élevé

Exemples

1

```
C:\>ftp -d 62.114.95.161
```

```
Connected to 62.114.95.161.
```

```
220 Serv-U FTP Server v4.0 for WinSockready...
```

```
User (62.114.95.161:(none)): tt
```

```
---> USER tt
```

```
331 User nameokay, needpassword.
```

```
Password:
```

```
---> PASS tt
```

```
230 User logged in, proceed.
```

```
ftp>dir
```

```
---> PORT 212,93,164,155,6,65
```

```
200 PORT Command successful.
```

```
---> LIST
```

```
150 Opening ASCII mode data connection for /bin/ls.
```

```
drw-rw-rw-  1 user  group    0 Jan  1 06:31 .
```

```
drw-rw-rw-  1 user  group    0 Jan  1 06:31 .
```

```
-rw-rw-rw-  1 user  group  28827 Oct  1 1994 BOOKMARK.HTM
```

```
drw-rw-rw-  1 user  group    0 Jan 17 20:17 Fav
```

```
drw-rw-rw-  1 user  group    0 Jan  1 06:31 Hacking
```

```
-r--r--r--  1 user  group  38912 Oct 17 1994Scrap.shs
```

```
226 Transfer complete.
```

```
ftp: 376 bytes received in 0.12Seconds 3.11Kbytes/sec.
```

```
ftp> bye
```

```
---> QUIT
```

```
221 Goodbye!
```

```
*****
```

2

```
pass in quick on $int_if proto tcp to port 21 divert-to 127.0.0.1 port 8021
```

3

```
pass in on $ext_if proto tcp to port > 49151
```

4

```
ftpproxy_flags="-R 10.10.10.1 -p 21 -b 192.168.0.1"
```

5

```
ext_ip          = "192.168.0.1"
ftp_ip          = "10.10.10.1"
```

```
match out on $ext_if inet from $int_if nat-to ($ext_if)
```

```
anchor "ftp-proxy/*"
pass in on $ext_if inet proto tcp to $ext_ip port 21
pass out on $int_if inet proto tcp to $ftp_ip port 21 user proxy
```

6

```
match out on $ext_if from $int_if nat-to ($ext_if)
anchor "tftp-proxy/*"
pass in quick on $int_if inet proto udp from $int_if to port tftp \
  divert-to 127.0.0.1 port 6969
```

```
anchor "tftp-proxy/*"
```

7

```
127.0.0.1:6969 dgram udp wait root /usr/libexec/tftp-proxy tftp-proxy
```

8

```
List /
226 Transfer complete
```

Remarque

* verbose - cette commande peut être utilisée pour activer ou désactiver l'affichage de tous les messages reçus d'un serveur FTP. Le mode verbose est actif par défaut.

* system - cette commande peut être utilisée pour afficher le répertoire o/system du serveur FTP.

7. Les réponses FTP

Les réponses FTP permettent d'assurer la synchronisation entre client et serveur FTP. Ainsi à chaque commande envoyée par le client, le serveur effectuera éventuellement une action et renverra systématiquement une réponse.

Les réponses sont constituées d'un code à 3 chiffres indiquant la façon suivant laquelle la commande envoyée par le client a été traitée. Toutefois, ce code à 3 chiffres étant difficilement lisible par un humain, il est accompagné d'un texte (chaîne de caractères Telnet séparée du code numérique par un espace).

Les codes de réponse sont constitués de 3 chiffres dont voici les significations :

- Le premier chiffre indique le statut de la réponse (succès ou échec)
- Le second chiffre indique ce à quoi la réponse fait référence
- Le troisième chiffre donne une signification plus spécifique (relative à chaque deuxième chiffre)

Premier chiffre		
Chiffre	Signification	Description
1yz	Réponse préliminaire positive	L'action demandée est en cours de réalisation, une seconde réponse doit être obtenue avant d'envoyer une deuxième commande
2yz	Réponse positive de réalisation	L'action demandée a été réalisée, une nouvelle commande peut être envoyée
3yz	Réponse intermédiaire positive	L'action demandée est temporairement suspendue. Des informations supplémentaires sont attendues de la part du client
4yz	Réponse négative de réalisation	L'action demandée n'a pas eu lieu car la commande n'a temporairement pas été acceptée. Le client est prié de réessayer ultérieurement
5yz	Réponse négative permanente	L'action demandée n'a pas eu lieu car la commande n'a pas été acceptée. Le client est prié de formuler une requête différente

Second chiffre		
Chiffre	Signification	Description
x0z	Syntaxe	L'action possède une erreur de syntaxe, ou bien il s'agit d'une commande non comprise par le serveur
x1z	Information	Il s'agit d'une réponse renvoyant des informations (par exemple pour une réponse à une commande STAT)
x2z	Connexions	La réponse concerne le canal de données
x3z	Authentification et comptes	La réponse concerne le login (USER/PASS) ou la demande de changement de compte (CPT)
x4z	Non utilisé par le protocole FTP	
x5z	Système de fichiers	La réponse concerne le système de fichiers distant

8. paramètres de transfert des fichiers

La spécification du protocole FTP fournit de nombreux choix dans la façon de transférer et de stocker les fichiers. Essentiellement, le choix peut se faire dans un espace à 4 dimensions :

1. Le type de fichier :
 - a) ASCII ...
 - b) EBCDIC ...
 - c) binaire ...
 - d) local ...
2. Le format de contrôle (limité aux types de fichier ASCII et EBCDIC) :
 - a) Nonprint
 - b) Contrôle de format TELNET
 - c) Contrôle de transport FORTRAN
3. La structure :
 - a) Structure de fichier
 - b) Structure d'enregistrement
 - c) Structure de page

4. Le mode de transmission (spécifiant comme le fichier est transféré sur la connexion de données) :
- a) Mode flux [stream mode]
 - b) b. Mode bloc [block mode]
 - c) Mode compressé [compressed mode]

9. conclusion

Dans ce chapitre, nous avons représenté le protocole FTP, avec son fonctionnement, les commandes et les réponses pour établir la connexion entre le utilisateur et le serveur ce qui nous a permis de distinguer deux modes de fonctionnement importants en FTP qui sont le mode actif et le mode passif à travers des canaux de contrôle et de données.

Chapitre 2

Installation et configuration de fileZilla server

Ce chapitre est élaboré a partir de [4]

1. Présentation

Il existe de nombreux logiciels de serveurs FTP disponibles pour windows et les autres systèmes d'exploitation. Nous avons choisi un serveur FTP nommé Filezilla serveur. Filezilla serveur est un logiciel libre, souvent mise à jour, simple d'emploi et très complet. Son installation est parfaitement simple et classique pour un logiciel windows.

2. Installation de FileZilla Server

Tout d'abord, on récupère le fichier d'installation sur le site officiel :

<https://filezilla-project.org/download.php?type=server>

Une fois téléchargé, on passe à l'installation du logiciel.



Figure 5 : l'icône FileZilla

En premier lieu, la licence d'utilisation s'affiche une interface graphique. On clique sur le bouton I Agree. Par défaut, l'installation se fait en Standard, on peut éventuellement, faire une installation complète (Full), installer uniquement le service (Service only), l'interface uniquement (Interface only) ou l'installation personnalisée (Custom).

Dans le cas présent, on va faire une installation standard puis on clique sur le bouton Next. Le répertoire d'installation par défaut (sur un windows 7 en 64 bits) :

On clique sur le bouton Next

Par défaut, l'installation du service ainsi que le démarrage du service au démarrage de Windows. La deuxième option disponible est l'installation du service mais avec un démarrage manuel. La troisième option n'installe pas de service, par contre le serveur démarre automatique.

Dans le cas présent, on choisit l'option n°2 (installation du service et démarrage manuel).

On peut éventuellement décocher le démarrage du serveur après l'installation (Start Server after setup completes). Le port par défaut (14147) pour l'interface d'administration du serveur FTP peut être personnalisé. Une fois les options définies, on clique sur le bouton Next.

Ceci concerne le démarrage de l'interface d'administration. L'option par défaut démarre lorsque n'importe quel utilisateur se connecte à une session, la deuxième option permet de démarrer l'interface d'administration si seulement c'est l'utilisateur

courant qui se connecte à la machine. La troisième option permet de démarrer l'interface d'administrateur manuellement.

Dans le cas présent, nous sélectionnons la troisième option (Start manually) puis on clique sur le bouton Install.

On peut décocher l'option Start Interface after setup completes si l'on souhaite ou non démarrer l'interface d'administration et cliquer sur le bouton Install pour démarrer l'installation du logiciel.

Une fois l'installation terminée, on clique sur le bouton Close.

Si l'installation s'est bien déroulée, on doit avoir le programme **FileZilla Server** dans le menu démarrer.

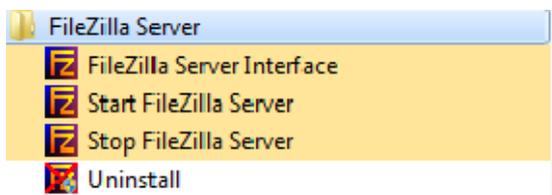


Figure 6 : le menu démarrer

Pour démarrer le serveur FTP, on clique **Start FileZilla Server** en faisant un clic droit « **Exécuter en tant qu'administrateur** » sinon le service ne démarrera pas.

Pour arrêter le serveur FTP, on clique sur **Stop FileZilla Server** en faisant un clic droit « **Exécuter en tant qu'administrateur** » sinon le service ne s'arrêtera pas.

Pour lancer l'interface d'administration, on clique sur **FileZilla Server Interface**

Pour vérifier si le service **FileZilla Server** est démarré, on va dans le menu **Gérer** (on clic droit sur **Ordinateur**) :

Ensuite, on va dans **Services et Applications** puis **Services** :

Normalement, on trouve dans la liste le service **FileZilla Server FTP server** :

Le service présente un état **Démarré**, c'est-à-dire que le service FTP est lancé.



Figure 7 : FileZilla server FTP server

Si le service **n'est pas démarré**, cela se présente comme ci-dessous :



Figure 8 : FileZilla server FTP server

On peut accéder également aux services en tapant **services.msc** dans le menu rechercher les programmes et fichiers.

3. Interface d'administration

On clique sur **FileZilla Server Interface** dans le menu Démarrer,

Dans le cas présent, nous avons l'adresse du serveur (**Server Address**), le port d'administration (**14147** par défaut) ainsi que le mot de passe administrateur (**Administration password**, par défaut, il n'est pas défini).

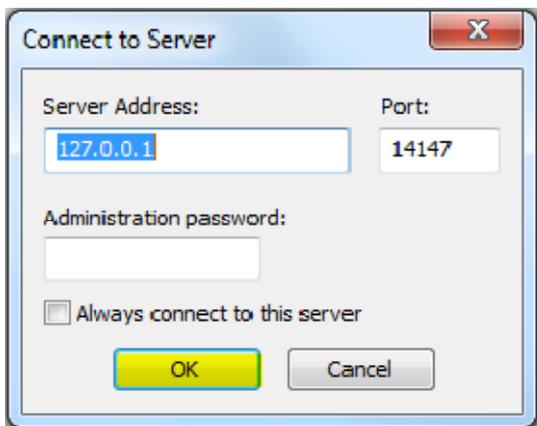


Figure 9 : l'interface d'information de serveur

Always connect to this server permet de se connecter toujours à ce serveur.

On clique ok

La fenêtre suivante s'affiche, elle présente les éléments d'interface suivants :

- Le menu de l'interface d'administration :

- La barre des icônes qui donne l'accès aux différentes fonctions du serveur FTP
- un message d'accueil

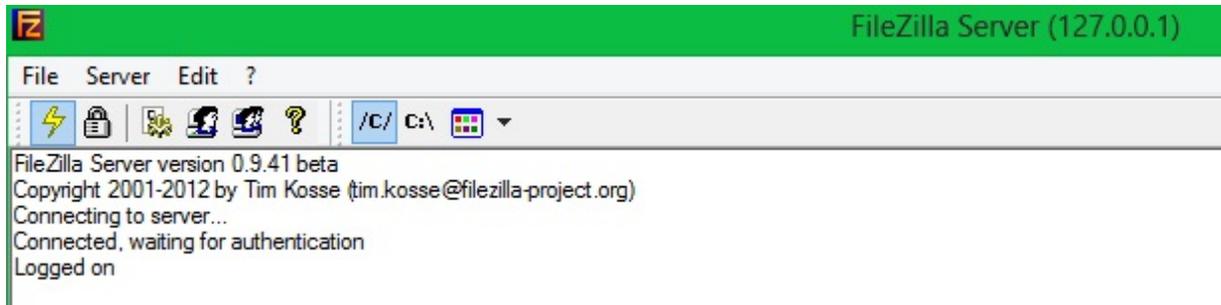


Figure 10 : l'interface d' administration de serveur

 Active/désactive (Online / Offline) le serveur FTP.

 Lock le serveur FTP (Aucune connexion entrante n'est acceptée).

 Paramètres du serveur FTP.

 Gestion des utilisateurs.

 Gestions des groupes.

 Affiche les informations concernant l'auteur du logiciel.

L'accès aux différentes fonctions peut se faire **soit via les icônes** vu précédemment **soit par les menus détaillés** ci-dessous.

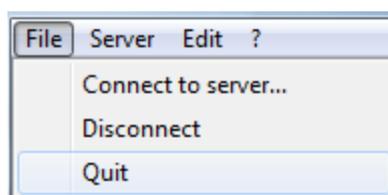


Figure 11 : le menu détaillé

Le menu **File** propose les options suivantes :

Connect to server : permet de se connecter au serveur que l'on souhaite administrer.

Disconnect : permet de se déconnecter du serveur que l'on administre.

Quit : permet de quitter l'interface d'administration.

Le menu **Server** propose les options suivantes :

Active : permet de d'activer ou de désactiver (online/offline) le serveur FTP depuis l'interface d'administration. Un message demande confirmation de l'administrateur pour mettre le serveur FTP hors ligne (**offline**) :

Lock : permet de ne plus accepter de connexion entrante sur le serveur FTP (suite à un message de confirmation).

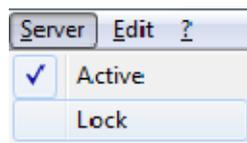


Figure 12 : le menu server

Le menu **Edit** propose les options suivantes :

Settings permet de définir les différents paramétrages.

Users permet de gérer les utilisateurs.

Groups permet de gérer les groupes d'utilisateurs.

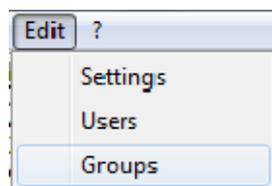


Figure 13 : le menu edit

4. Les paramètres généraux de FileZilla Server

Revenons sur le menu **Edit** qui permet de définir les paramètres (**Settings**), les utilisateurs (**Users**) et les groupes (**Groups**) d'utilisateurs.

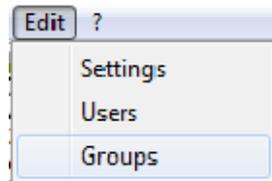


Figure 14 : le menu edit

Tout d'abord, nous allons voir l'option **Settings** puis **General settings** :

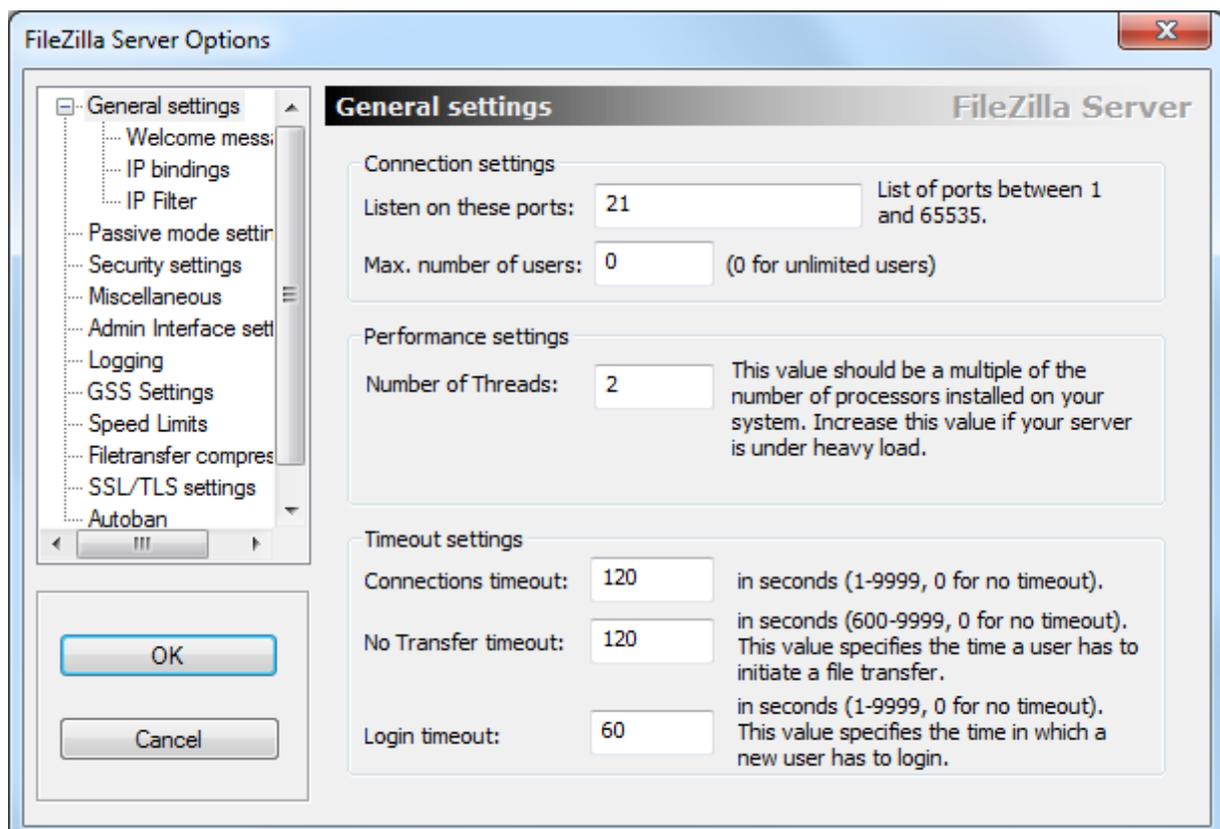


Figure 15 : FileZilla server options

Listen on these ports : port d'écoute du serveur FTP (21 par défaut).

Max. number of users: 0 pour un nombre illimité d'utilisateur connecté.

Number of Threads: permet de régler le nombre de processus utilisé par le serveur FTP.

Connections timeout : laps de temps avant une déconnexion si pas d'action (120 par défaut).

No Transfert timeout : laps de temps avant une déconnexion quand il n'y a pas de transfert de fichier (120 par défaut).

Login timeout : laps de temps pour effectuer la connexion au serveur FTP (60 par défaut).

4.1.General settings

-IP bindings

Cette option permet de définir l'IP qui est utilisée par le serveur FTP pour fonctionner. Par défaut , le serveur écoute sur toutes les IP assignées à une ou plusieurs cartes réseaux installées sur la machine. Cette configuration est représentée par le symbole : *

* : localhost (127.0.0.1) et ip réseau local (exemple : 192.168.1.2)

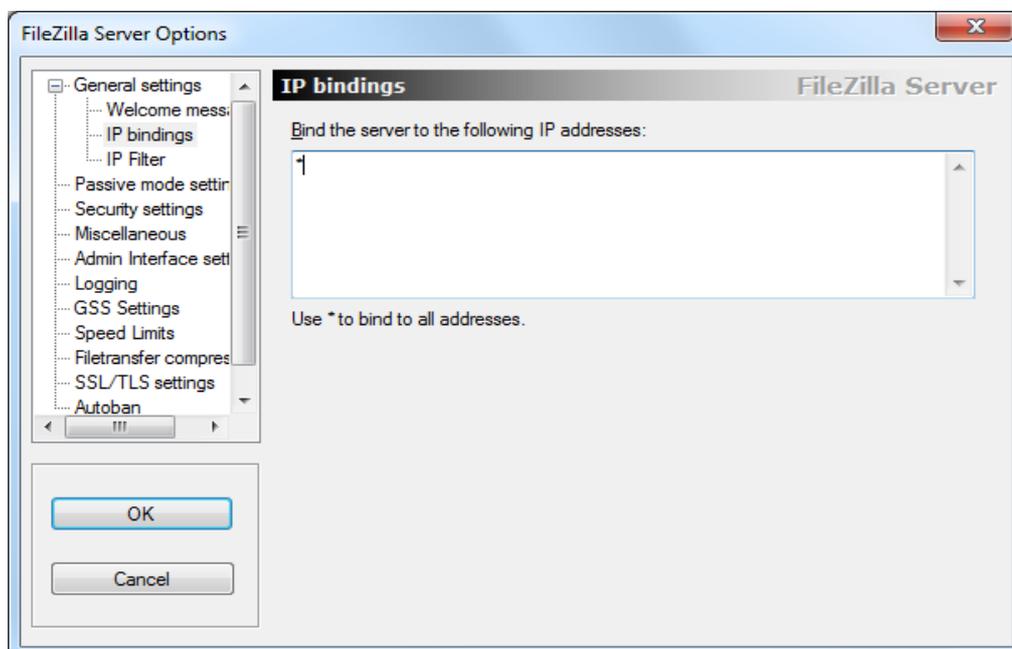


Figure 16 : IP bindings

-IPFilter

Cette option permet d'autoriser ou de refuser des adresses IP sur le serveur FTP.

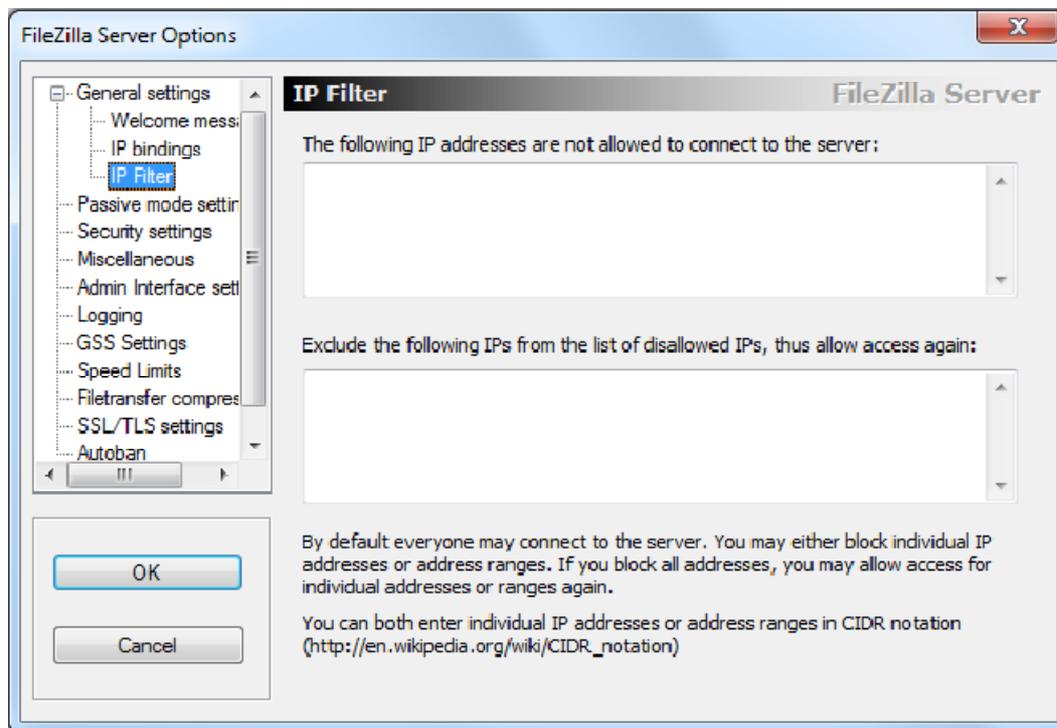


Figure 17 : IP filter

La partie du haut permet d'interdire une IP et la partie du bas permet d'autoriser une IP à se connecter au FTP. On peut autoriser ou d'interdire soit une IP distincte soit un bloc d'IP.

4.1.1 General settings / Passive mode settings

Cette rubrique permet de définir la plage pour les ports PASV (on coche l'option **Use custom port range**) qui seront utiliser par le serveur FTP. *La commande PASV indique au serveur qu'il doit attendre passivement la connexion en écoutant un port TCP*

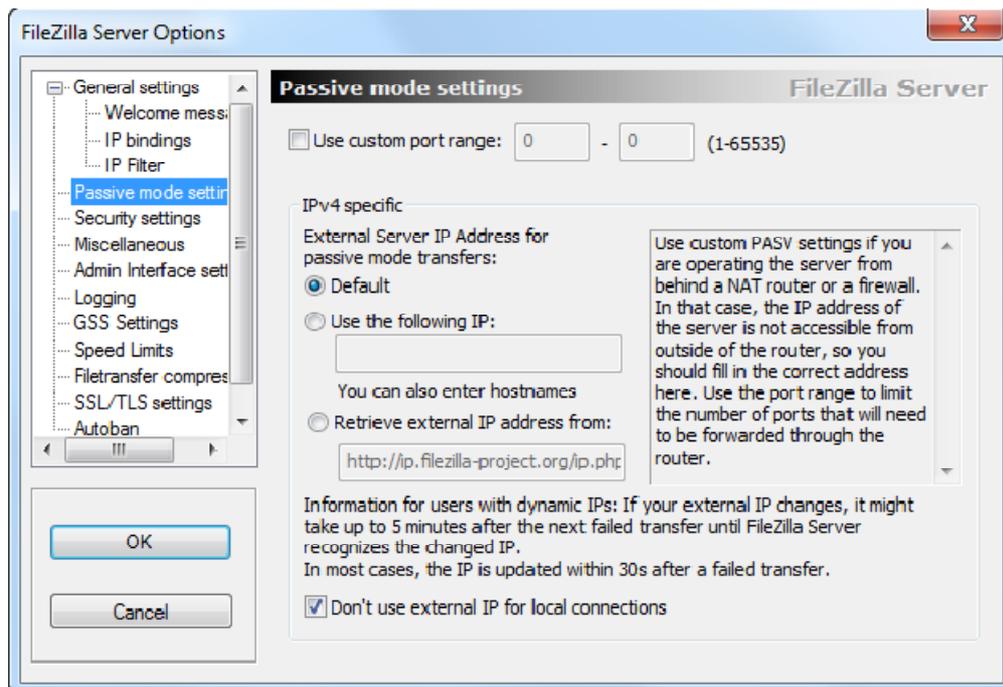


Figure 18: passive mode settings

IPv4 specific permet de définir la façon dont on récupère l'**IP externe** de la connexion internet utilisée pour le **PASV**.

Default : l'adresse IP est celle dont dispose le logiciel par défaut.

Use the following IP : on définit l'adresse IP manuellement que l'on souhaite utiliser.

Retrieve external IP address from: l'adresse IP est récupérée via le site officiel du logiciel.

L'option **Don't use external IP for local connections** dit que l'on n'utilisera pas d'IP externe pour des connexions sur un réseau local.

4.1.2 General settings / Security settings

Cette rubrique permet de définir la sécurité en cas d'attaque de type **bounce attack** ou de transfert de serveur FTP à serveur FTP (**FXP transfers**).

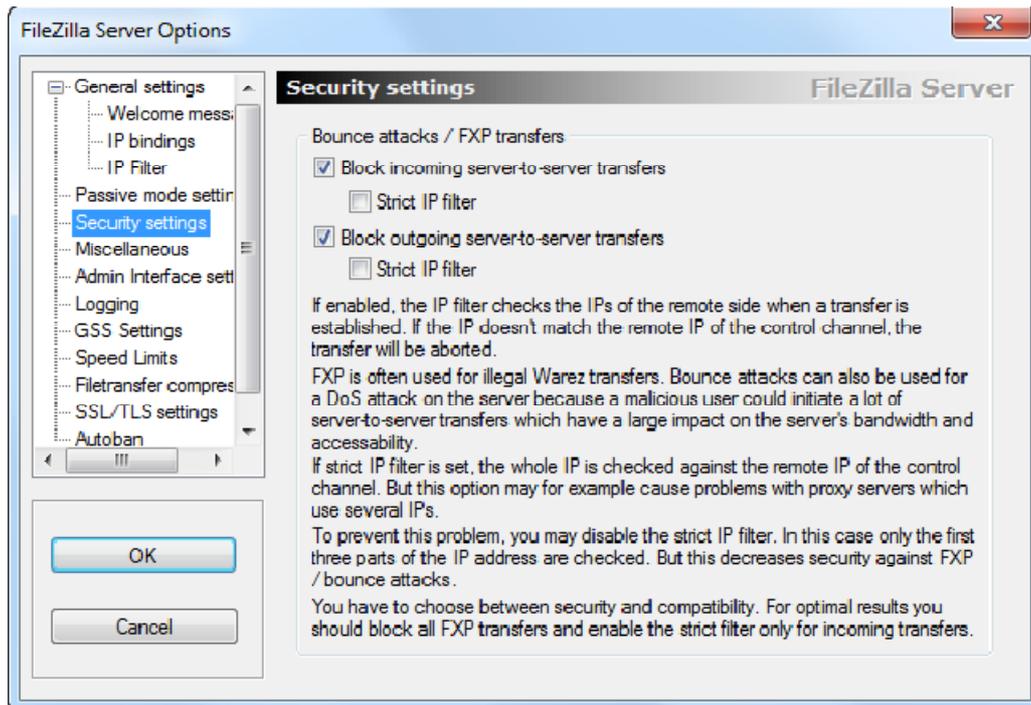
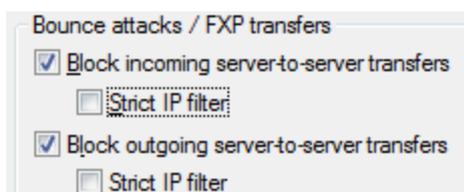


Figure 19 : security settings

Dans les deux cas, ceci est bloqué par défaut : **Block incoming server-to-server transfers** et **Block outgoing server-to-server transfers**

Strict IP filter permet d'activer la vérification de l'IP *attaquante*. Cocher l'option permet d'avoir une sécurité supplémentaire.



4.1.3 General settings / Admin interface settings

Cette rubrique permet de définir les options liées à l'interface d'administration.

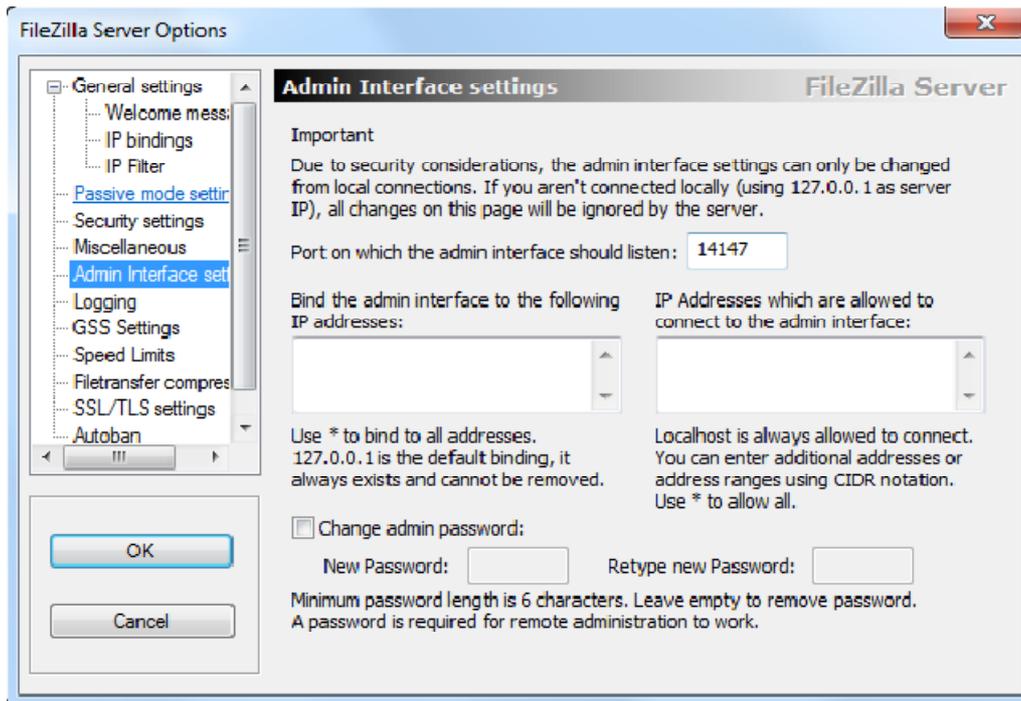


Figure 20 : admin interface settings

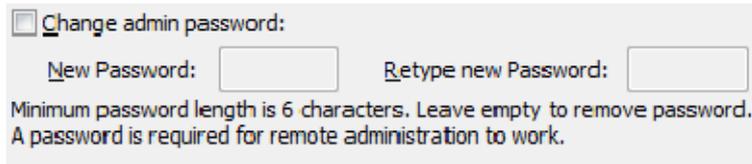
On peut définir le port d'administration (**14147** par défaut). On peut utiliser un port compris entre 0 et 65535 (hormis le 20 et 21 utilisé par le serveur FTP).

Port on which the admin interface should listen:

L'adresse IP sur laquelle écoute l'interface d'administration (à gauche ci-dessous) ainsi que les IP autorisées (à droite ci-dessous) à se connecter à la dite interface.



Change adminpassword permet de définir ou de changer le mot de passe qui est utilisé lors de la connexion à l'interface d'administration. Le mot de passe doit faire **au minimum 6 caractères**, si on n'indique pas de mot de passe cela supprime celui qui aurait pu être défini.

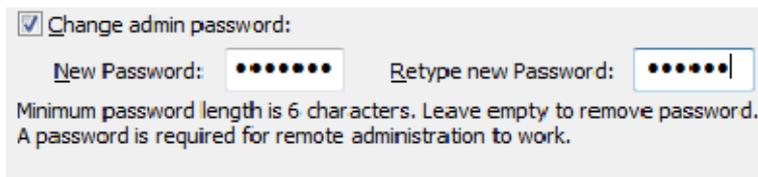


Change admin password:

New Password: Retype new Password:

Minimum password length is 6 characters. Leave empty to remove password.
A password is required for remote administration to work.

Pour définir un mot de passe on coche l'option **Change adminpassword** puis on tape une première fois (**New Password**) le mot de passe puis une deuxième fois (**Retype new Password**) :



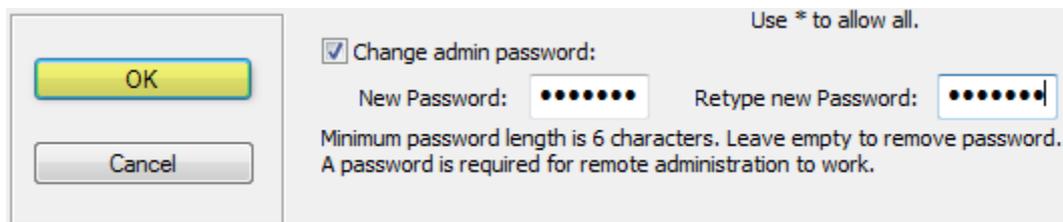
Change admin password:

New Password: Retype new Password:

Minimum password length is 6 characters. Leave empty to remove password.
A password is required for remote administration to work.

Si le mot de passe est trop court, on obtient le message d'erreur .

Pour valider le mot de passe que l'on vient de définir, on clique sur le bouton **OK** (ci-dessous).



Use * to allow all.

Change admin password:

New Password: Retype new Password:

Minimum password length is 6 characters. Leave empty to remove password.
A password is required for remote administration to work.

OK

Cancel

4.1.4 General settings / Logging

Cette rubrique permet d'activer l'option de **LOG** pour avoir un fichier journal des différentes actions qui pourrait avoir lieu sur le serveur FTP. Par défaut, l'option n'est pas activée.

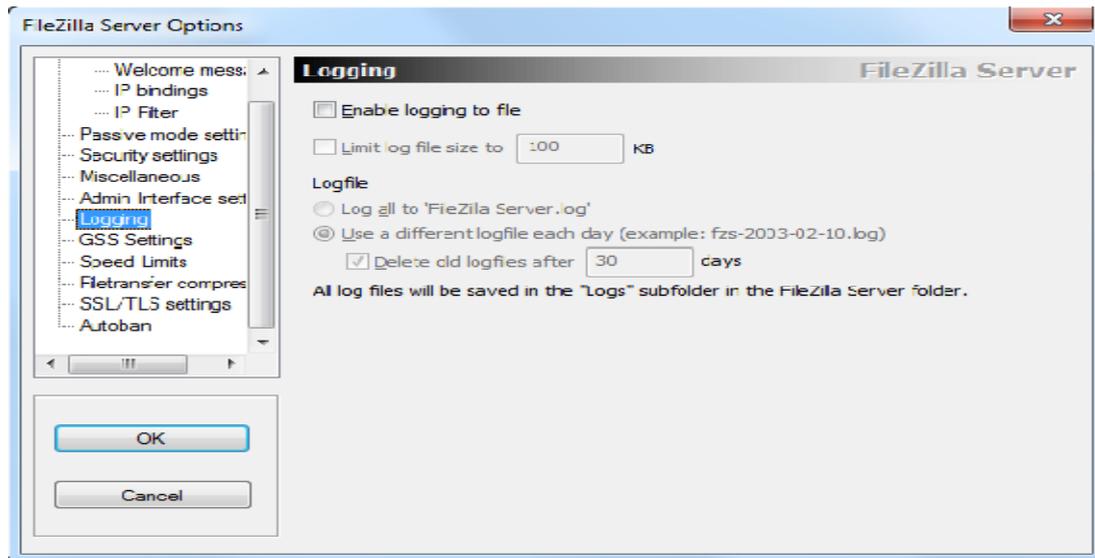
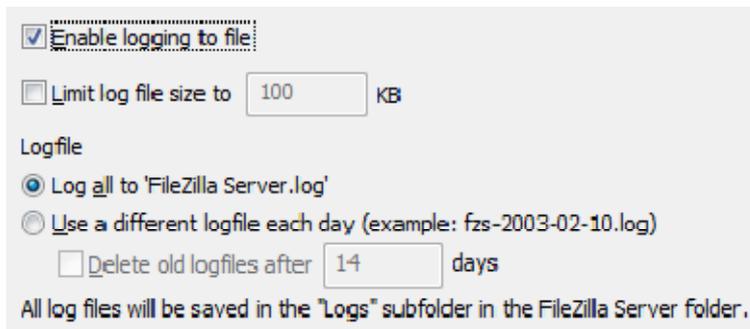


Figure 21 : l'interface logging

Pour se faire, on coche l'option :



Une fois l'option activée :



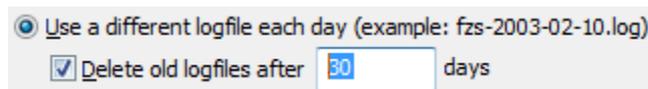
Limit log file size to ... KB permet de définir la taille maximum du fichier journal (log).

Log all to 'FileZilla Server.log' enregistrera dans un seul et unique fichier **FileZilla Server.log**.

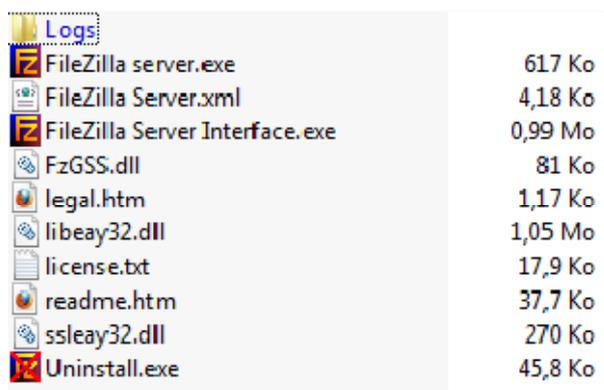
Use a different logfile each day permet de créer un fichier journal par jour sous la forme **fzs-année-mois-jour.log**

Deleteoldlogfilesafter ... days permet de supprimer automatiquement les anciens fichiers journaux après **X** jours.

Exemple : on enregistre **un fichier / jour** et on supprime les anciens fichiers **après 30 jours**.



Tous les fichiers journaux sont enregistrés dans un sous répertoire **Logs** du répertoire d'installation de FileZilla Server à savoir **C:\Program Files (x86)\FileZilla Server\Logs** (Windows 7 en 64 bits).

A screenshot of a Windows Explorer window showing the contents of a directory named 'Logs'. The window displays a list of files and folders with their respective sizes. The files are: FileZilla server.exe (617 Ko), FileZilla Server.xml (4,18 Ko), FileZilla Server Interface.exe (0,99 Mo), FzGSS.dll (81 Ko), legal.htm (1,17 Ko), libeay32.dll (1,05 Mo), license.txt (17,9 Ko), readme.htm (37,7 Ko), sslsleay32.dll (270 Ko), and Uninstall.exe (45,8 Ko).

File Name	Size
FileZilla server.exe	617 Ko
FileZilla Server.xml	4,18 Ko
FileZilla Server Interface.exe	0,99 Mo
FzGSS.dll	81 Ko
legal.htm	1,17 Ko
libeay32.dll	1,05 Mo
license.txt	17,9 Ko
readme.htm	37,7 Ko
sslsleay32.dll	270 Ko
Uninstall.exe	45,8 Ko

Figure 22 :repertoire logs

4.1.5 General settings / GSS Settings

Cette rubrique permet de définir les options pour utiliser un serveur d'authentification **KERBEROS**

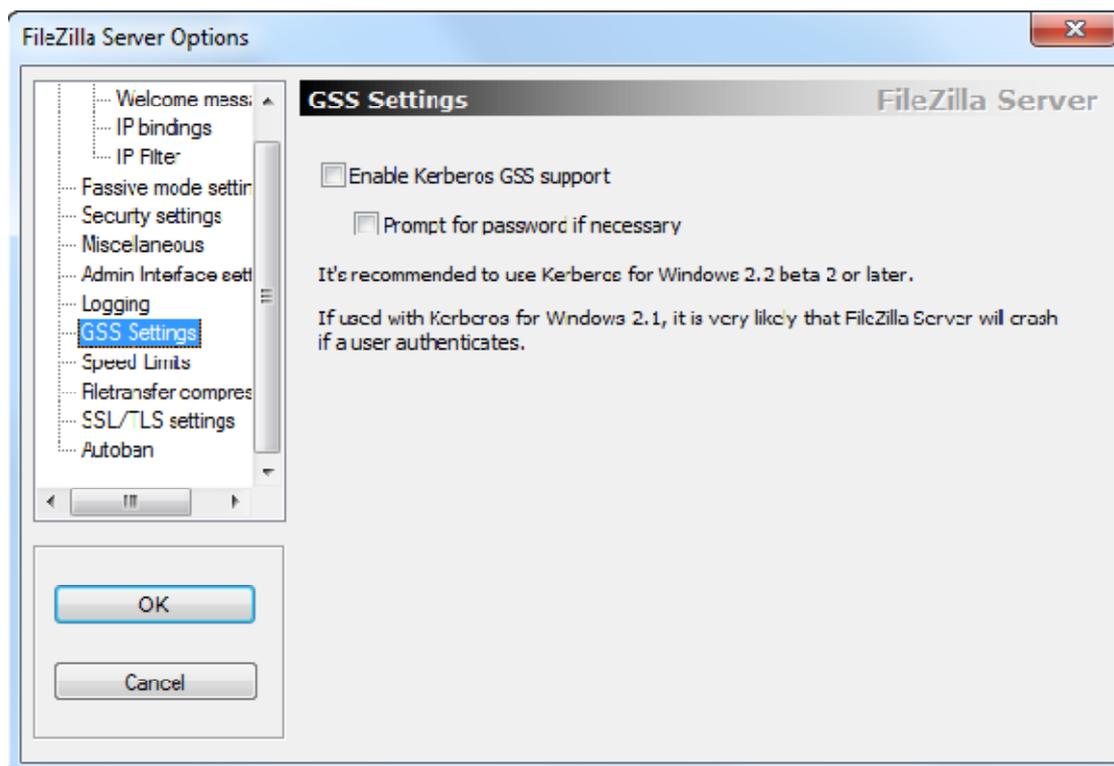


Figure 23 : l'interface GCC Settings

Pour activer le support Kerberos, il suffit de cocher l'option **EnableKerberos GSS support**.

Si on coche la case **Prompt for password if necessary**, le mot de passe sera demandé si nécessaire.

4.1.6 General settings / Speed Limits

Cette rubrique permet de définir les vitesses de téléchargement pour l'envoi ou la réception de fichier.

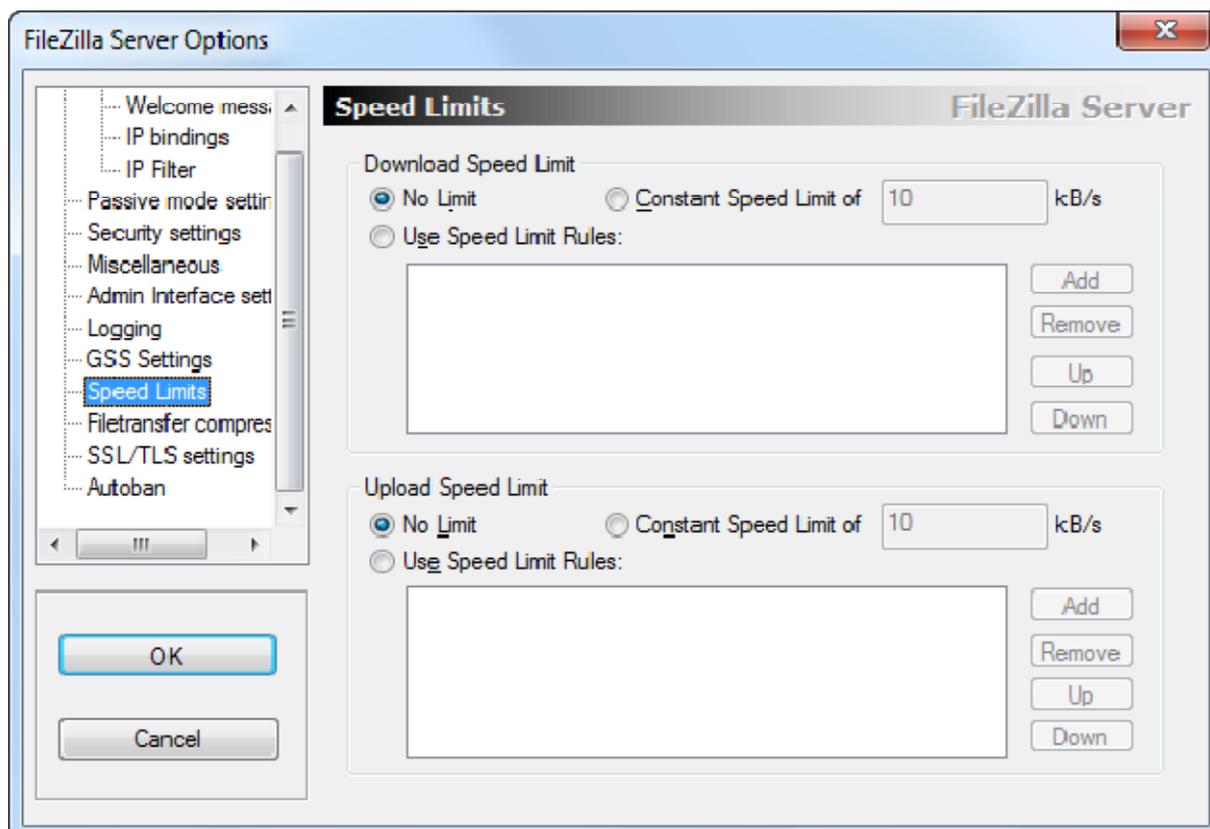


Figure 24 : l'interface speed limits

Par défaut, aucune limite de téléchargement n'est définie (**No Limit**).

Voyons les différentes options de limitation de débit en envoi et en réception.

Download Speed Limit :

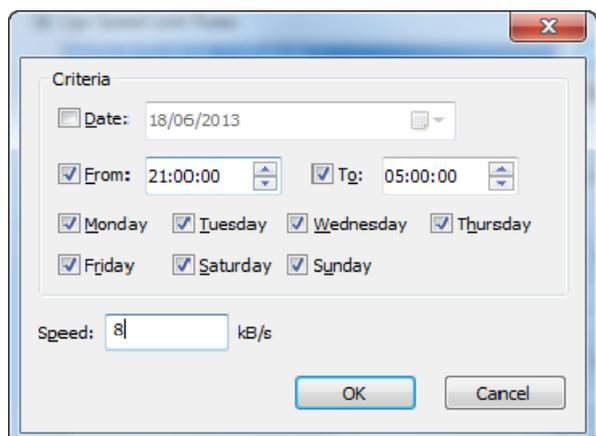
No Limit : pas de limitation.

Constant Speed Limit of ... kB/s : on définit une vitesse qui sera constante.

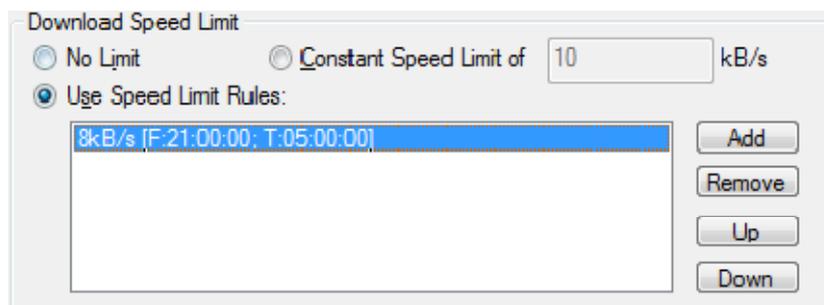
Use Speed LimitRules : des règles sont appliquées selon une date, un créneau horaire, un jour de la semaine, une vitesse de téléchargement.

Si l'on souhaite définir des règles pour la réception de fichier (**Download Speed Limit**), on choisit l'option **Use Speed LimitRules** puis on clique sur le bouton **Add**.

Dans le cas présent, on souhaite mettre en place une règle par rapport à un créneau horaire de 21h à 5h soit (**From 21:00:00 To 05:00:00**), pour tous les jours de la semaine, on définit une valeur de 8 kB/s et on clique sur le bouton **OK** pour confirmer.



Une fois la règle validée, elle apparaît comme ceci :



Pour la modifier, il suffit de cliquer dessus. Si l'on a plusieurs règles on peut définir un ordre d'application avec les boutons **Up** et **Down**. On peut mettre en place le même type de règle pour **Upload Speed Limit**.

4.1.7 General settings / Filetransfer compression

Cette rubrique permet d'activer la compression des fichiers à la voler.

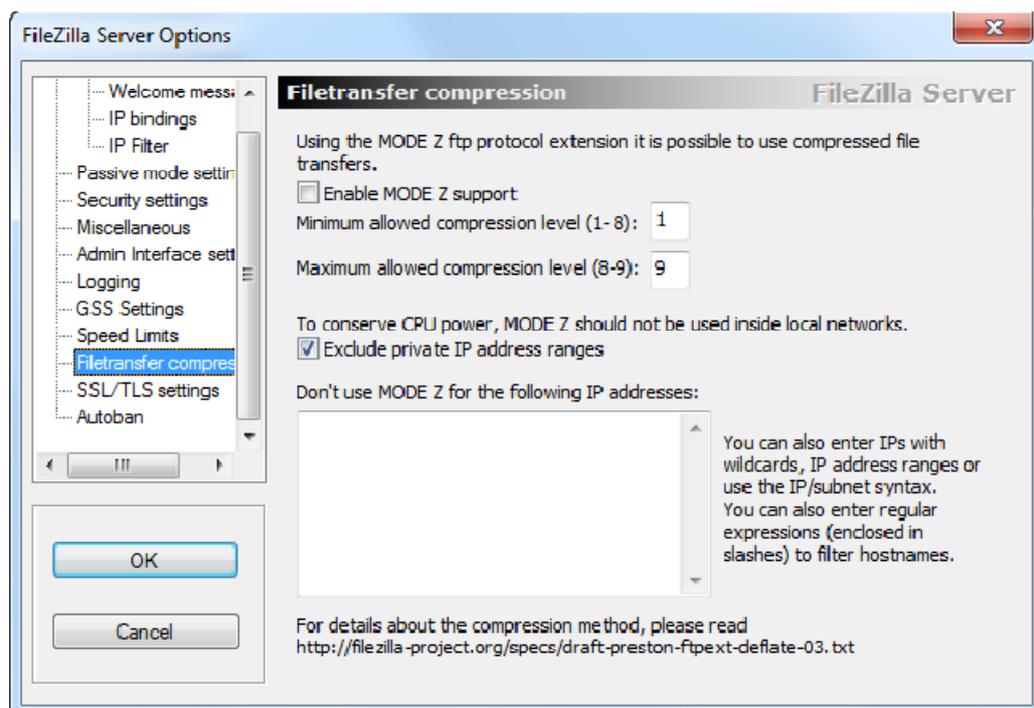


Figure 25 : l'interface filetransfer compression

Pour ce faire, on coche l'option **Enable MODE Z support** et on définit le niveau de compression.

Minimum allowed compression level : Niveau de compression minimale autorisé, entre 1 et 8.

Maximum allowed compression level : Niveau de compression maximale autorisé, entre 8 et 9.

L'option **Exclude private IP address ranges** exclut les plages d'adresses IP privées (réseau local).

L'option **Don't use MODE Z for the following IP addresses** permet de définir la ou les IP pour lesquelles on n'utilisera pas la compression de fichier.

4.1.8 General settings / SSL/TLS settings

Cette rubrique permet d'utiliser les fonctions de chiffrement de communication.

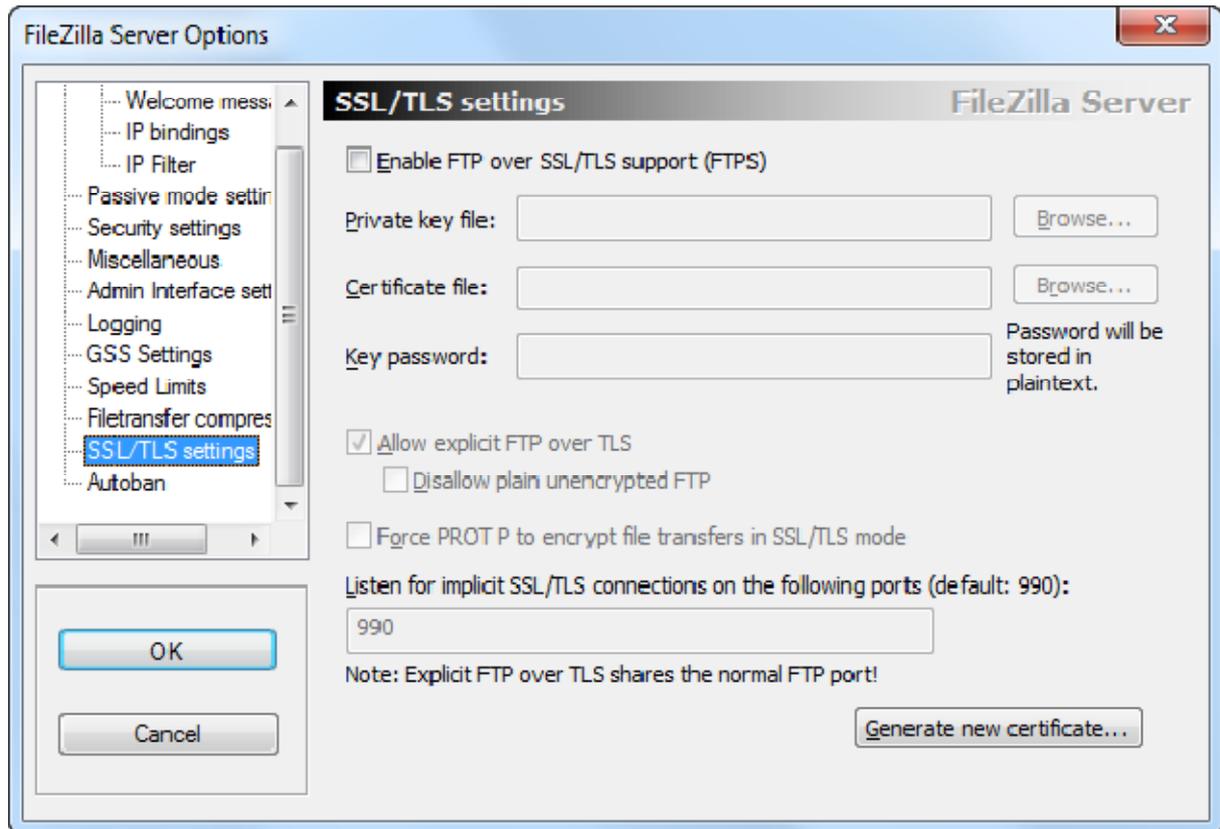


Figure 26 : l'interface SSL/TLS settings

Pour ce faire, on active l'option **Enable FTP over SSL/TLS support (FTPS)** qui va sécuriser le FTP et devenir un FTP Sécurisé.

Les champs **Private key file** et **Certificate file** font référence au fichier contenant la **clef privée** ainsi que le fichier de **certificat**. La première chose à faire étant de générer un nouveau certificat (**Generate new certificate**).

Enable FTP over SSL/TLS support (FTPS)

Private key file:

Certificate file:

Key password: Password will be stored in plaintext.

Allow explicit FTP over TLS
 Disallow plain unencrypted FTP

Force PROT P to encrypt file transfers in SSL/TLS mode

Listen for implicit SSL/TLS connections on the following ports (default: 990):

Note: Explicit FTP over TLS shares the normal FTP port!

Pour cela, on clique sur le bouton et on renseigne les différents champs ainsi que la taille de la clé (**Key size, par défaut 1024**), si on souhaite avoir une sécurité accrue, on peut choisir **une taille de clé de 2048 bit**.

This dialog will help you to create a new private key and a self-signed certificate, needed by FileZilla Server to accept SSL/TLS connections.
Please fill out the required information. Wrong or missing information may confuse clients.

Key size: 1024 bit 2048 bit 4096 bit

2-Digit country code:

Full state or province:

Locality (City):

Organization:

Organization unit:

Contact E-Mail:

Common name (Server address):

Save key and certificate to this file:

Generating the certificate may take some time depending on the key size.

Figure 27 : l'interface generatecertificate

Une fois les informations rentrées, on clique sur le bouton **Browse** pour choisir l'emplacement où seront enregistrés la clé et le certificat et pour finir on clique sur le bouton **Generate certificate** pour générer le nouveau certificat.

Une fois le certificat généré, les champs **Private key file** et **Certificate file** sont renseignés automatiquement. Puis on rentre un mot de passe clé (**Key password**).

L'option **Allow explicit FTP over TLS** autorise la connexion FTP sur TLS. C'est-à-dire que la connexion sera sécurisée.

L'option **Disallow plain unencrypted FTP** interdit la connexion sans TLS. C'est-à-dire qu'une connexion en clair ne sera pas autorisée.

L'option **Force PROT P to encrypt file transfers in SSL/TLS mode** va crypter les données qui seront transférées en mode FTP sécurisé.

Le port par défaut pour le **FTPS** est le **990**. Il peut être défini sur un autre port. Si l'on utilise le **FTPS**, le port de données est le **989**.

4.1.9 General settings / Autoban

Cette rubrique permet d'activer l'option pour bannir les 'utilisateurs illégitimes'.

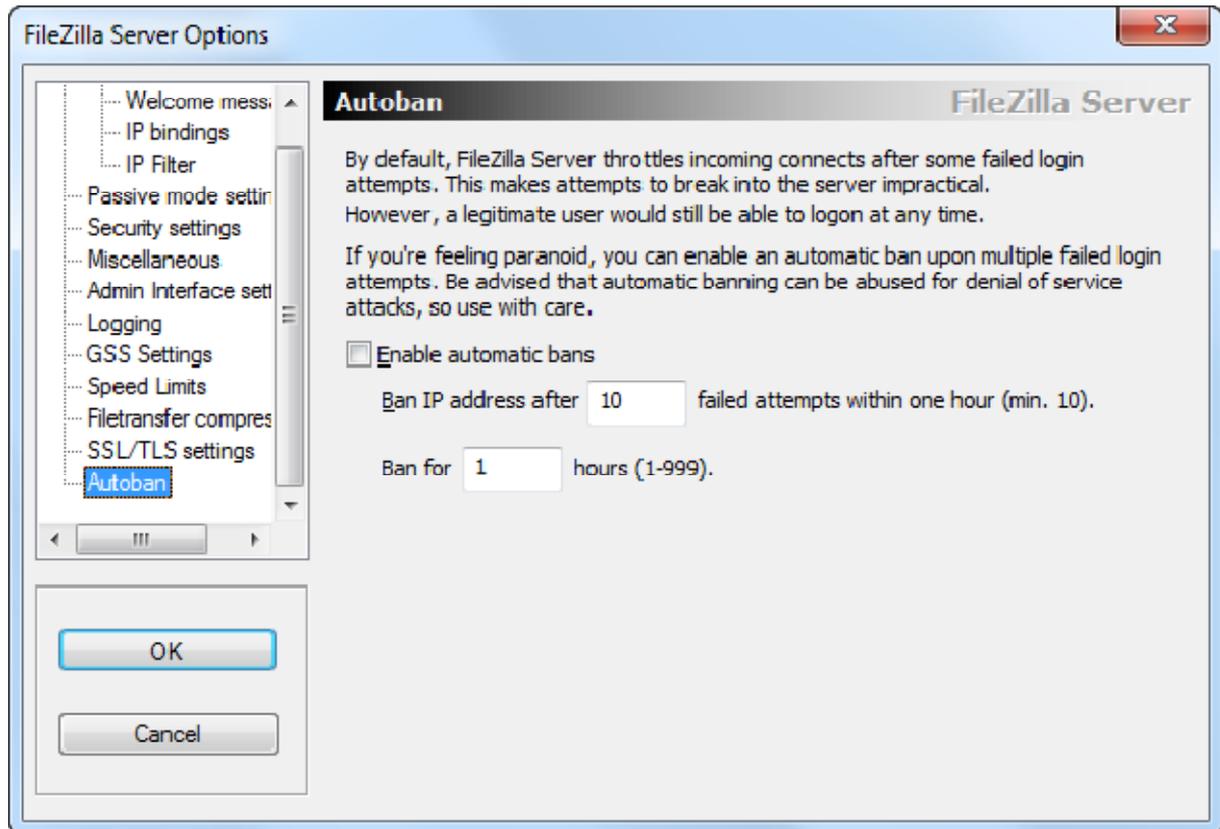


Figure 28 : l'interface Autoban

Pour ce faire on coche l'option **Enable automatic bans** qui aura pour effet de bannir un utilisateur qui essaierait de se connecter au serveur FTP avec un login/pass incorrect de manière répétée.

Par défaut, si l'utilisateur essaye **de s'authentifier 10 fois en 1 heure** alors l'utilisateur sera **banni pour 1 heure**.

5. La première connexion FTP

Du côté du serveur FTP via l'interface d'administration :

```

FileZilla Server (127.0.0.1)
File Server Edit ?
FileZilla Server version 0.9.41 beta
Copyright 2001-2012 by Tim Kosse (tim.kosse@filezilla-project.org)
Connecting to server...
Connected, waiting for authentication
Logged on
(000014)10/05/2014 18:36:44 - mino (127.0.0.1)> QUIT
(000014)10/05/2014 18:36:44 - mino (127.0.0.1)> 221 Goodbye
(000014)10/05/2014 18:36:44 - mino (127.0.0.1)> disconnected.
(000015)10/05/2014 18:37:45 - (not logged in) (127.0.0.1)> Connected, sending welcome message...
(000015)10/05/2014 18:37:45 - (not logged in) (127.0.0.1)> 220-FileZilla Server version 0.9.41 beta
(000015)10/05/2014 18:37:45 - (not logged in) (127.0.0.1)> 220-written by Tim Kosse (Tim.Kosse@gmx.de)
(000015)10/05/2014 18:37:45 - (not logged in) (127.0.0.1)> 220 Please visit http://sourceforge.net/projects/filezilla/
(000015)10/05/2014 18:37:45 - (not logged in) (127.0.0.1)> USER mino
(000015)10/05/2014 18:37:45 - (not logged in) (127.0.0.1)> 331 Password required for mino
(000015)10/05/2014 18:37:45 - (not logged in) (127.0.0.1)> PASS *****
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> 230 Logged on
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> FEAT
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> 211-Features:
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> MDTM
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> REST STREAM
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> SIZE
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> MLST type*size*modify*
(000015)10/05/2014 18:37:45 - mino (127.0.0.1)> MLSD

```

Figure 29 : l'interface d'administration du serveur FTP

On voit que l'utilisateur **minosait** connecté au serveur FTP et qu'il est actuellement dans **la racine /soit le répertoire HOME**.

Dans le bas de la fenêtre du serveur FTP, on peut voir l'activité de ce dernier (utilisateur connecté, IP, transfert en cours, progression, vitesse) :

ID	Account	IP	Transfer	Progress	Speed
←000015	mino	127.0.0.1			

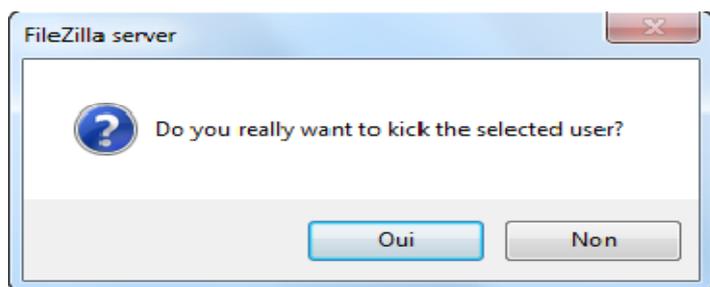
Si on fait un clic droit sur la connexion en cours, on peut déconnecter l'utilisateur (**Kick user**) ou le bannir (**Ban user's IP address**) :

ID	Account	IP	Transfer	Progress	Speed
←000005	mino	127.0.0.1			

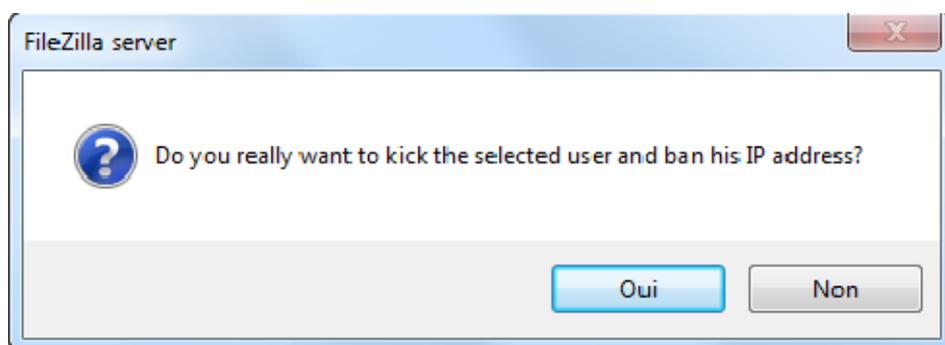
Kick user

Ban user's IP address

Message de confirmation, si l'on souhaite déconnecter (**Kick user**) l'utilisateur (ci-contre).



Message de confirmation si l'on souhaite déconnecter et bannir l'utilisateur (**Ban user's IP address**) (ci-contre).



6. CONFIGURATION D'UN SERVEUR FTP

La majorité des progiciels de serveur FTP permettent de configurer un certain nombre de paramètres, par exemple :

* Nombre de connexions simultanées permises (également désignées sous le nom de « régulation »). Habituellement, les serveurs FTP sont configurés pour utiliser un nombre de connexions inférieur à celles d'un serveur Web puisqu'une session FTP moyenne est plus longue et permet de transférer beaucoup plus de données. Cette valeur peut être réduite si les demandes du client FTP imposent une charge excessive au serveur.

* Le port TCP que surveille le serveur FTP pour les demandes de connexion de client. Par défaut, il s'agit du port 21; toutefois, l'utilisation d'un port différent permet d'obtenir une meilleure protection du serveur. Les clients qui se connectent au serveur doivent connaître le port à utiliser, sinon il pourrait être difficile d'établir une connexion. L'adresse URL peut être modifiée et inclure le numéro de port utilisé par FTP. Par exemple : `ftp://ftp.mycompany.com:22`
Il n'est pas nécessaire d'inclure le numéro de port dans l'adresse URL lorsque l'on utilise le port implicite.

* On peut configurer trois messages différents dans un serveur FTP :

- message de bienvenue - s'affiche lors de la première connexion de l'utilisatrice.

- message de sortie - s'affiche lors de la déconnexion de l'utilisatrice.
- message de connexion maximale - s'affiche lorsque le nombre de connexions maximales simultanées du paramètre a été atteint.

Les répertoires FTP peuvent être configurés de manière à prendre en charge un accès en mode lecture et (ou) écriture. L'autorisation d'écriture permet aux utilisatrices de produire la liste des fichiers puis de les copier du serveur vers le client. Les autorisations d'écriture permettent aux utilisatrices de télécharger des fichiers vers le serveur.

Il est recommandé de prévoir un répertoire spécialisé pour les fichiers téléchargés afin de s'assurer que les utilisatrices n'écrasent pas, par inadvertance, des fichiers portant le même nom.

Les serveurs FTP peuvent prendre en charge les connexions. Le niveau de détail peut être configuré aux fins de sécurité et de vérification.

7. SÉCURITÉ DES SERVEURS FTP

Les serveurs FTP offrent normalement deux types d'accès : authentifié et anonyme.

Accès authentifié avec nom d'utilisatrice et mot de passe

Le serveur FTP authentifie l'utilisatrice en s'assurant que celui-ci fournit un nom et un mot de passe valables avant de permettre l'accès à ses ressources.

Comme c'est le cas pour les serveurs de fichiers, le nom d'utilisatrice et les groupes auxquels elle appartient peuvent déterminer les ressources qui lui sont accessibles ainsi que le niveau d'accès associé à chaque ressource.

Les autorisations d'accès aux ressources peuvent être configurées dans un répertoire ou au niveau du fichier.

8. Conclusion

Dans ce chapitre nous avons expliqué les étapes à suivre pour installer FileZilla server et pour configurer le serveur et les différents paramètres du serveur.

Chapitre 3

Implémentation de client ftp

Ce chapitre est élaboré a partir de document JAVA [5]

1. Introduction

Dans ce chapitre, nous allons présenter notre application, qu'est un client FTP sous Windows compatible avec FileZilla Server.

2. Environnement de développement

Au cours du processus de développement, on a utilisé les moyens matériels et logiciels suivants :

2.1 Moyen matériel

Deux PC (machine server et machine client) dont les caractéristiques sont :

- .Processeur « core i5 »
- .6G de RAM
- .écran 15
- .carte graphique RADEON
- .Windows 7

2.2 Moyen logiciel

L'application a été développée sous l'environnement JAVA avec l'outil (JDK 7). Java est un langage de programmation, un langage dit compiler : c'est-à-dire que le code généré n'est pas directement compréhensible et utilisable par votre ordinateur. Nous devons donc passer par une étape de compilation (étape obscure où le code source est entièrement transformé). En fait, on peut distinguer trois grandes phases dans la vie d'un code Java :

- La phase d'écriture du code source, en langage Java ;
- La phase de compilation du code ;
- La phase d'exécution.

On a utilisé aussi **NetBeans** qui est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

L'IDE Netbeans s'enrichit à l'aide de greffons.

3. Programmation du protocole FTP en JAVA

3.1 Socket et server Socket

3.1.1 Socket

Nous avons utilisé les Sockets dans l'application.

Pour utiliser les Sockets, on fait l'importation de package « net ».

```
(import java.net.* ;)
```

Pour établir une connexion au server, il faut créer une instance de classe (objet) de type socket, le constructeur de cette classe contient deux paramètres : l'adresse IP de server et le port (port 21 par défaut en FTP).

```
Socket client = new Socket(server, port) ;
```

3.1.2 Server Socket

Utilisé par le serveur, et nous avons besoin de server socket pour accepter la communication qui est établie par le server.

```
ServerSocket ser = new ServerSocket(port) ;
```

```
Socket client = ser.accept() ;
```

3.2 Les flux d'entrée/sortie

Pour envoyer les commandes, et récupérer les réponses du server il faut utiliser les flux d'entrée/sortie.

3.2.1 Les flux d'entrée

Utilisé pour récupérer les réponses de server

```
InputStream in = clientSocket.getInputStream();
```

3.2.2 Les flux de sortie

```
OutputStream out = clientSocket.getOutputStream();
```

3.3 Fonctionnement du client

L'application est un client FTP qui travaille avec FileZilla server, et qui fonctionne comme suit :

-Le client peut télécharger ou envoyer un fichier directement.

- Le client peut accéder et gérer son disque local : supprimer ou renommer un fichier, créer ou ouvrir un dossier.

-Le client peut télécharger ou upload des fichier.

-Après l'établissement de la connexion l'utilisateur peut faire les opération suivantes :

*Suppression : Si l'élément à supprimer est un fichier

```
public boolean deleteFile(String fileName) throws IOException
```

```
{
```

```
int response = executeCommand("dele " + fileName);
```

```
return isPositiveCompleteResponse(response);
```

```
}
```

Si l'élément est un dossier

```
public boolean removeDirectory(String directory) throws IOException
```

```
{
```

```
int response = executeCommand("rmd " + directory);
```

```
return isPositiveCompleteResponse(response);
```

```
}
```

*Création d'un dossier

```
public boolean makeDirectory(String directory) throws IOException
{
    int response = executeCommand("mkd " + directory);
    return isPositiveCompleteResponse(response);
}
```

*Changement du répertoire de travail

```
public boolean changeDirectory(String directory) throws IOException
{
    int response = executeCommand("cwd " + directory);
    return isPositiveCompleteResponse(response);
}
```

*Changement du répertoire père

```
public boolean parentDirectory() throws IOException
{
    int response = executeCommand("cdup");
    return isPositiveCompleteResponse(response);
}
```

4 Présentation du client FTP

Lors d'ouverture de notre client, la fenêtre suivante s'ouvre

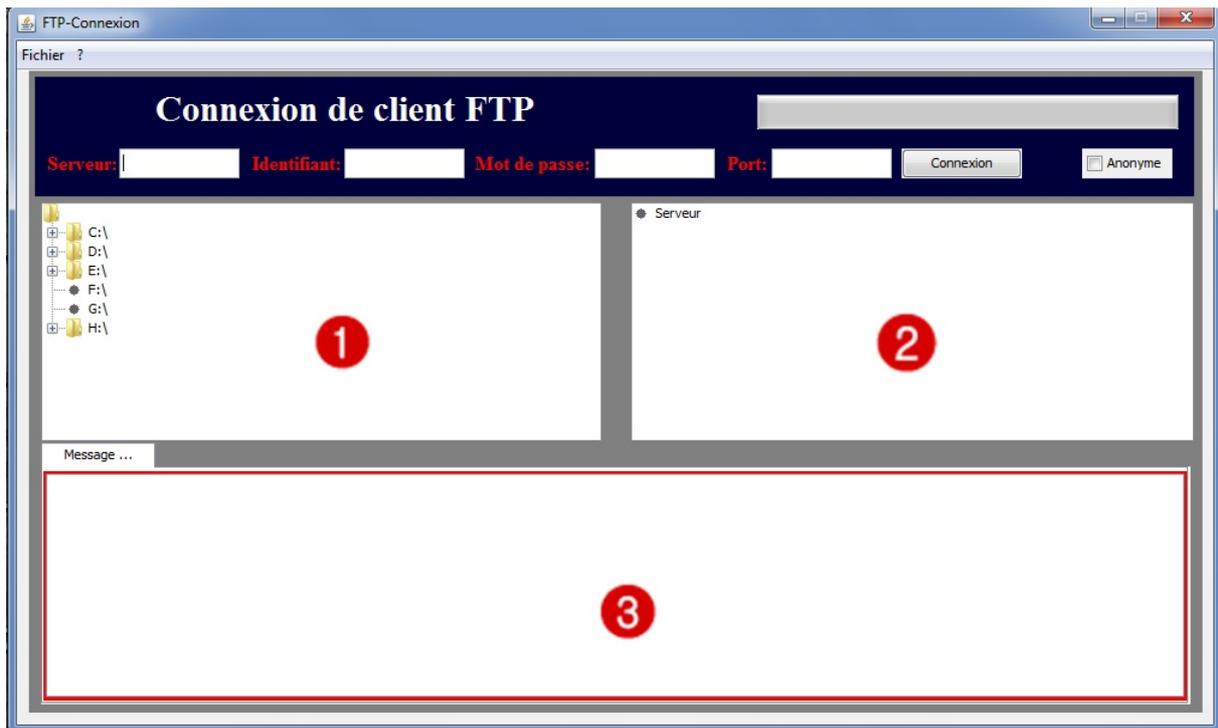


Figure 30 : interface client

Cette interface (Figure1) contient trois parties qui sont :

4.1. Barre de menu :

Un menu Fichier : il nous permet de quitter l'application.

4.2. Barre de connexion

Est un outil pour se connecter au serveur FileZilla server. Elle contient trois champs :

- * Server : pour entrer l'adresse du serveur
- * Identifiant : pour entrer le nom d'utilisateur pour le serveur
- * Mot de passe : pour entrer le mot de passe de l'utilisateur si il existe

4.3. Les panneaux

Dans ce client FTP, on distingue trois panneaux :

Le panneau 1 : ce panneau représente les répertoires et les fichiers de disque dur local (machine client).

Le panneau 2 : ce panneau représente les répertoires et les fichiers distants (FileZilla server).

Le panneau 3 : ce panneau représente les messages de confirmation ou d'erreur.

4.4 Le bouton connexion

Pour connecter à serveur FileZilla

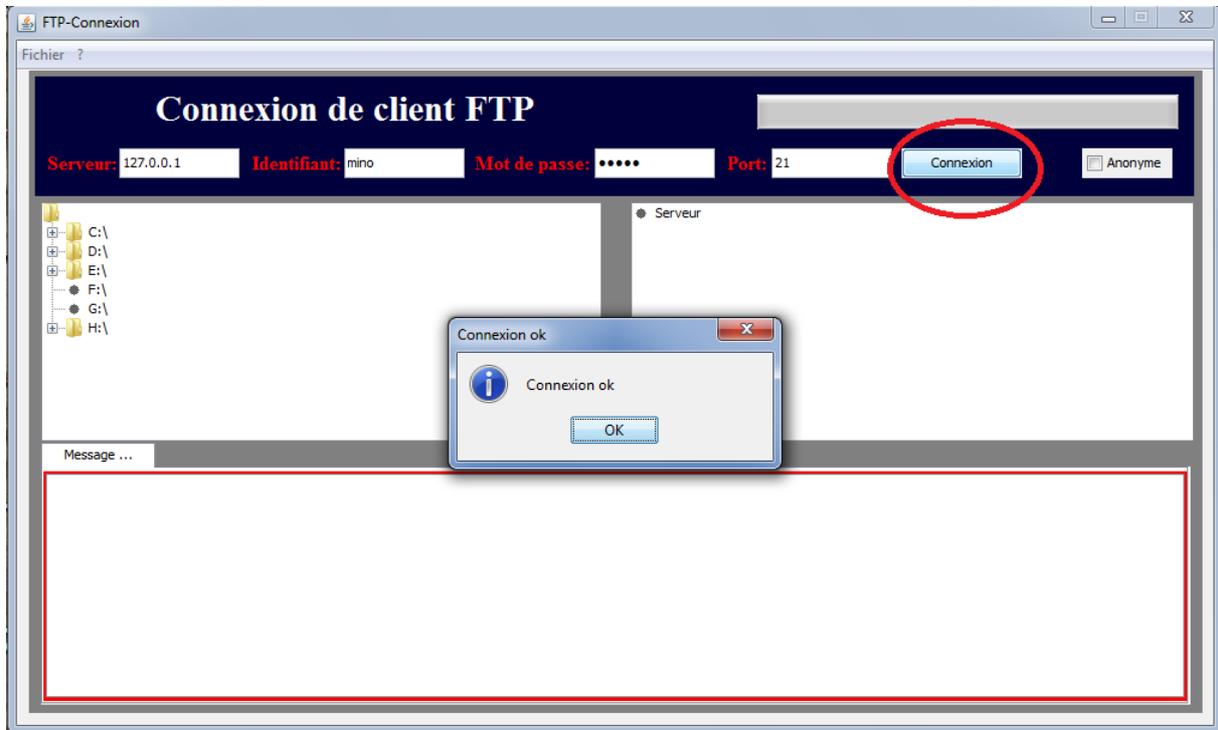


Figure 31 : connexion au serveur

Si les informations sont incorrectes un message d'information a été affiché

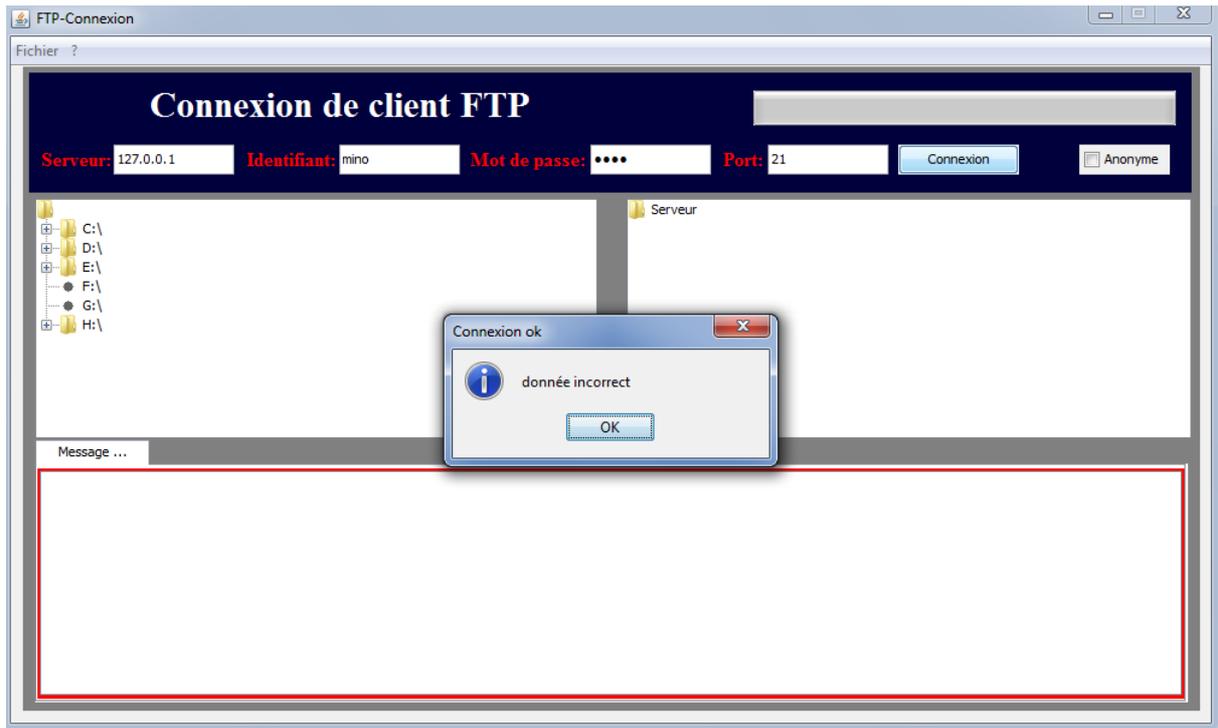


Figure 32 : message d'information

4.5 Le button déconnexion

Pour déconnecter du serveur.

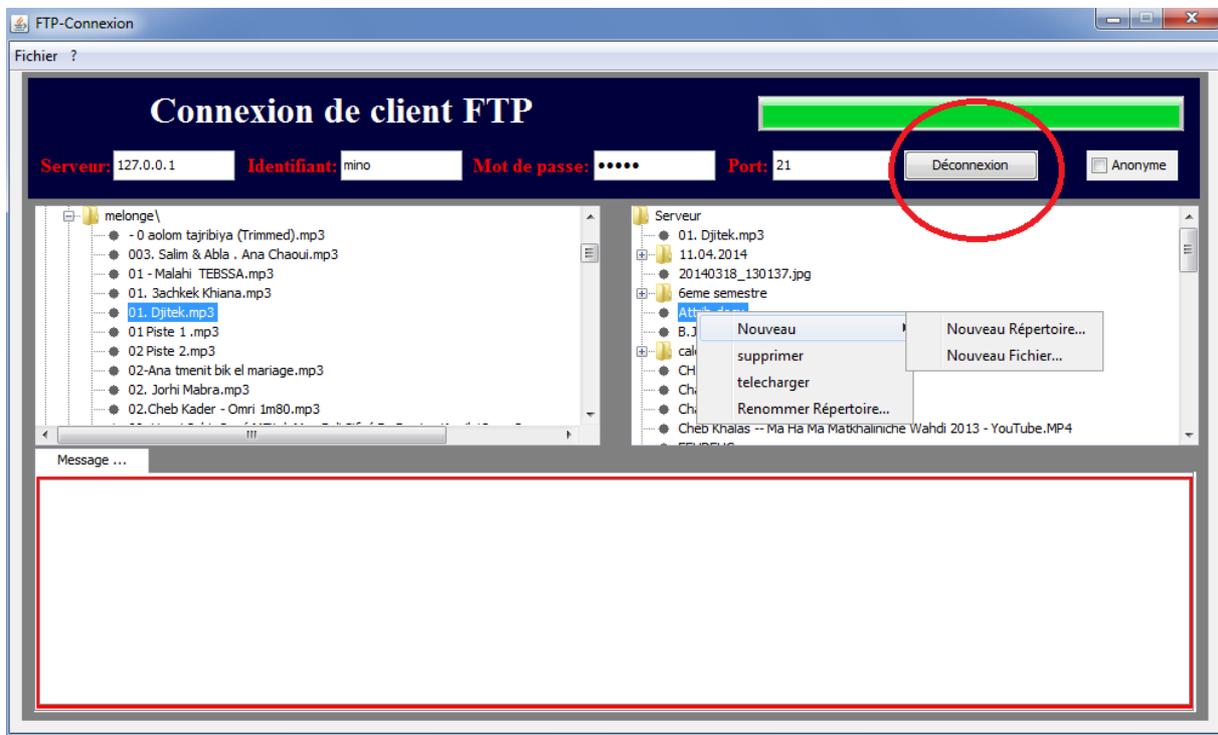


Figure 33 : déconnexion du serveur

5. Les opérations de notre client

5.1 Création de dossier

Pour créer un dossier, on fait un clic droit sur le homdir. Il apparaît un menu contextuel, il suffit de cliquer sur le sous menu Créer un dossier.

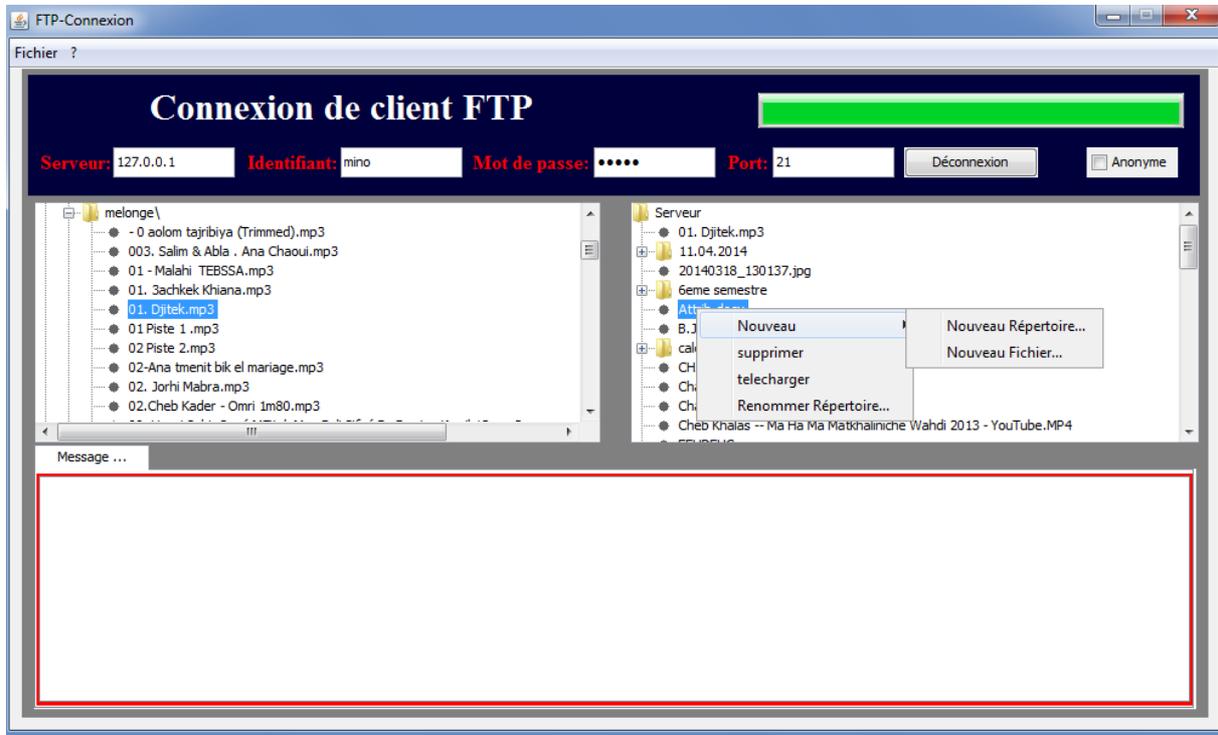


Figure 34 : créer un dossier

Une sous fenêtre affichera dans laquelle on peut saisir le nom de dossier qui nous allons le créer.

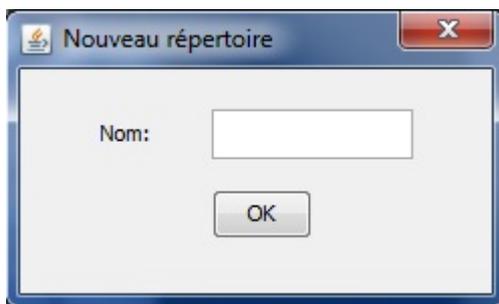


Figure 35 : entrer le nom de dossier

Les autres sous menu sont :

* Supprimer : pour supprimer un fichier ou un répertoire.

* Renommer : pour renommer un fichier ou un répertoire.

5.2 Téléchargement/Upload

5.2.1 Téléchargement

Pour télécharger un fichier il suffit de cliquer droit a ce dernier dans le home dir. Un menu contextuel s'affiche, alors on clique sur sous menu télécharger, une boite de dialogue apparait :

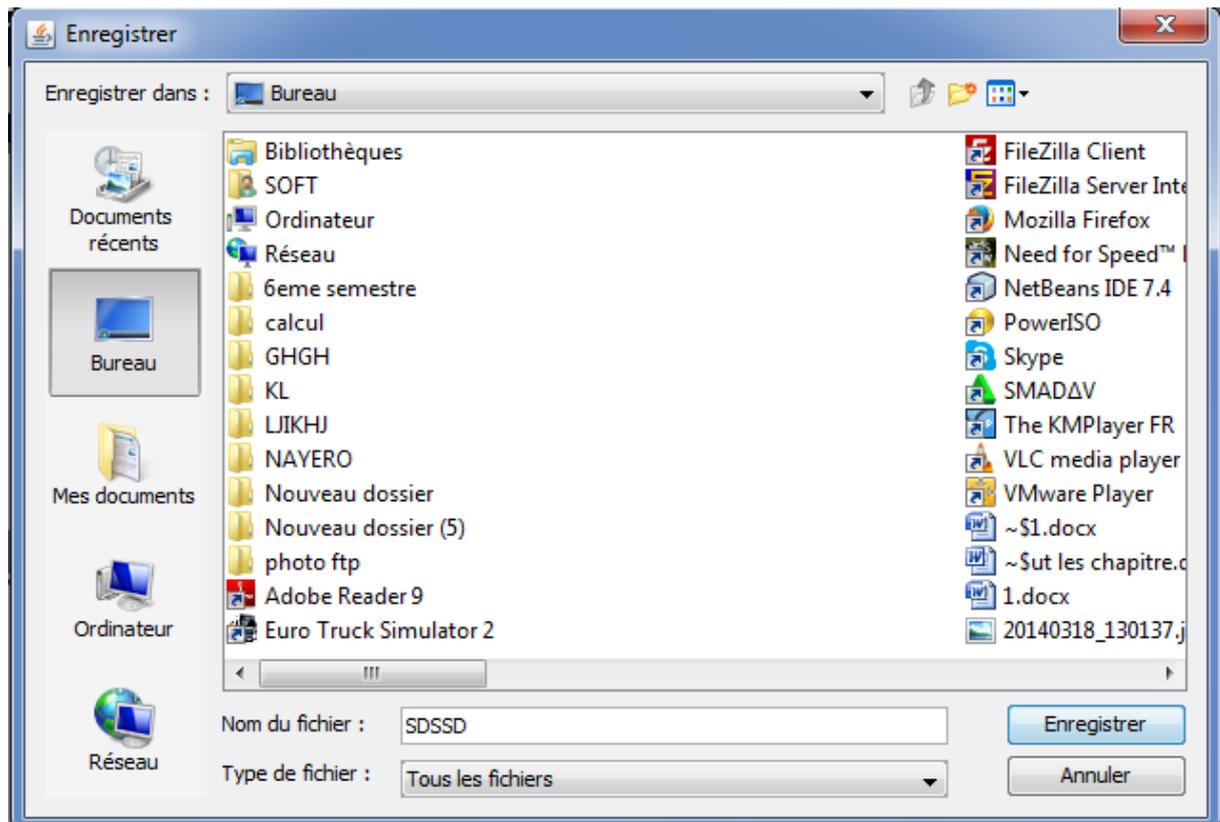


Figure 36 : boite de dialogue

5.2.2 Upload

Pour envoyer un fichier vers le serveur depuis notre disque local, on clique droit sur le fichier et on valide le sous menu Chareger un fichier.

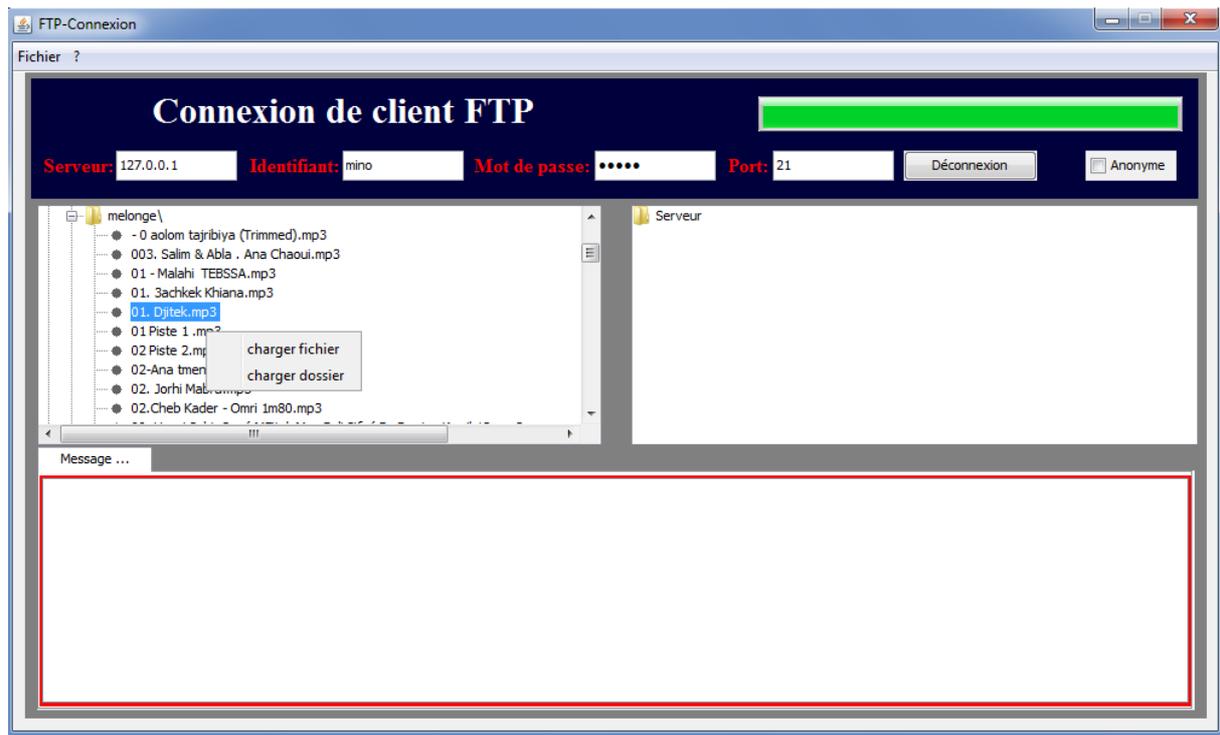


Figure 37 : envoyer le fichier

6. Conclusion

On a développé un client FTP compatible avec FileZillaserver ,qui fait la plupart des opérations connues du protocole FTP, et parmi ces opération on distingue :La suppression, renommée , téléchargement, l'upload....

Conclusion générale

Le but du projet est de réaliser un client FTP. Pour ce faire nous avons étudié le standard du protocole FTP et ses différents paramètres. Il est également question de comprendre les paramètres de configuration du serveur File Zilla utilisé.

Le client que nous avons réalisé assure les fonctionnalités de base de téléchargement dans les deux sens entre le serveur et le logiciel client de fichiers ou de répertoire. Il opère dans les deux modes étudiés : mode actif et mode passif.

Le client est développé en JAVA-NETBEANS IDE 7.4. Mais il est difficile de prétendre atteindre la perfection du client FILEZILLA et le détrôner. Nous proposons comme perspective à notre travail d'ajouter les fonctionnalités manquante dans un objectif purement pédagogique.

Bibliographie

Les référence :

[1] : <http://Forum.Eléments théoriques.fr>

[2] : <http://Comment ca marche .net>

[3] : <http://www.wistEE.net>

[4] : <http://www.filezilla server.net>

[5] : à partire de l'application

[6] : <http://www.wikipedia.com>