

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

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: - Filière informatique général

# Conception et Réalisation d'un Client FTP

Préparé par

Loubna barkat

Wahida chekhemoum

Encadré par :

TALAI .m

Année universitaire : 2013/2014



## *Remerciement*

*C'est avec l'aide de DIEU tout puissant que ce modeste mémoire a pu être réalisé,*

*DIEU qui nous a donné foi, raison et lucidité.*

*Nous tenons à remercier notre encadreur, Mme Talai M.*

*Nous tenons à remercier et à saluer tous les professeurs du département science et technologie.*

*Un grand merci à, Kaïss, Me Amri, pour leur aide et leur soutien qu'ils nous ont accordé.*

*Enfin, Nous tenons remercier tous ceux qui ont participé de près à la loi à la réalisation de ce modeste travail.*

*Merci à tous.*



## *Dédicaces*

*Je dédie ce modeste travail à :*

*Mes très chers parents, pour leur patience, leur soutien et leur confiance.*

*À mon père, Bouzide qui m'a donné le courage tout le temps.*

*À ma très chère maman Akila qui m'a fait protéger pendant*

*Toute ma vie*

*À tous mes frères et sœurs.*

*À tout ma famille.*

*À mon fiancé Omar.*

*À mon binôme Loubna et tout sa famille.*

*À tous mes amis surtout : Meriem, Salwa, Sabah, Samira,  
mouna Amira.*

*À tous mes enseignants sans exception.*

*Wahida*



## *Dédicaces*

*A chaque fois qu'on achève une étape importante dans notre vie,  
On fait une pose pour regarder en arrière et se rappeler toutes ces  
personnes Qui ont partagé avec nous tous les bons moments de notre  
existence, mais surtout les mauvais.*

*Ces personnes qui nous ont aidés sans le leur dire, soutenus sans  
réserve, aimé sans compter, Ces personnes à qui notre bonheur  
devient directement le leur, Se transforme en pleur.*

*Je dédie ce modeste travail en signe de reconnaissance et de respect*

*A ma très chère mère Ounassa le plus chère personne pour moi la  
Meilleure femme du monde qu'elle a été comme une bougie  
brule Et brûlé pour ses fils et sa famille, merci ma mère.*

*A mon cher père Tayeb qui m'a donné tous ces pouvoir et tout  
Son temps à moi et a tous mon parcours d'étude il m'a donnée  
Le Courage, la volonté, la confiance à moi-même, merci beaucoup  
Mon père.*

*A toutes mes sœurs et mes frères*

*A tous mes amis*

*A toute la famille Barkat*

*A tous mes enseignant sans exception*

*Loubna*

# *Table des matières*

## *Introduction Général*

### *Chapitre I le protocole FTP*

Introduction .....	1
I. Présentation du protocole FTP.....	1
II. Caractéristiques fonctionnelles de FTP.....	2
II.1. Canaux de transmission de FTP.....	2
II.2. Processus de transmission de FTP.....	3
III.1. Les modes de transfert des données.....	5
II.4. Les commandes internes de FTP.....	6
II.4.1. Les commandes de contrôle d'accès.....	6
II.4.2. Les commandes paramètre de transfère.....	7
II.4.3. Les commandes de service FTP.....	7
III. Déploiement de FTP au niveau utilisateur .....	9
III.1. Démare une session FTP.....	9
III.2 Les commandes FTP externe (utilisateur).....	10
III.3. les réponses FTP.....	10
Conclusion.....	12

# *Chapitre II installation et configuration du Serveur FileZilla Server*

Introduction.....	13
I. Installation de FileZilla Server.....	13
II. Accès à l'interface d'administrateur.....	17
III. Les rubriques de configuration de FileZilla Server.....	19
III.1. Rubrique de configuration générique.....	20
III.2. Rubrique de configuration de mode passif.....	21
III.3. Rubrique Miscellaneous (divers paramètres).....	23
III.4. Rubrique Admin interface setting.....	23
III.5. Rubrique Logging.....	24
III.6. Rubrique GSS Setting.....	24
III.7. Rubrique Speed limits.....	25
III.8. Rubrique File transfert compression.....	26
III.9. Rubrique SSL /TLS Setting.....	26
III.10. Rubrique Autoban.....	28
IV. Gestion d'utilisateur.....	29
IV. Rubrique Général.....	29
IV. Rubrique Shared Folders.....	32
IV. Rubrique compte utilisateur Speed Limits.....	33
Conclusion.....	33

## *Chapitre III Implémentation*

Introduction.....	35
I .L'environnement de programmation DELPHI 7.....	35
I.1 Choix de l'outil.....	35
I.2. L'interface de développement de Delphi.....	36
I.3 Pourquoi utiliser Delphi ?.....	37
II. L'Interface graphique .....	37
II.1. Interface authentification.....	37
II.2. La Fenêtre principal client FTP .....	38
II.3. Fonctionnalité du client FTP.....	39
Conclusion.....	45

### *Conclusion Générale*

### *Bibliographie*

# *Liste des Figure*

Figure I.1 modèle Client/ Serveur du FTP.....	1
figure I.2 Transfert de fichiers dans deux sens via FTP.....	2
Figure I.3 Canaux et processus de FTP.....	4
Figure I.4 FTP en mode actif.....	5
Figure I.5 FTP en mode Passif.....	6
Figure II.1 La première étape de l'installation.....	13
Figure II.2 Le choix d'un type d'installation.....	14
Figure II.3 Le choix du type de démarrage du serveur FTP.....	15
Figure II.4 le choix du type de démarrage de l'interface d'administration.....	16
Figure II.5 Interface administrateur du serveur.....	17
Figure II.6 Interface pour spécifier les informations de connexion au serveur.....	17
Figure II.7 L'interface d'administration.....	18
Figure II.8 La barre des icônes.....	18
Figure II.9 Le menu FILE.....	19
Figure II.10 Le menu Server .....	19
Figure II.11 Le menu Edit.....	19
Figure II.12 Fenêtre de configuration du serveur FTP.....	20
Figure II.13 Les options génériques de configuration du serveur.....	21
Figure II.14 Fenêtre de paramétrage du mode passif.....	22
Figure II.15 paramètres de la rubrique Miscellaneous.....	23
Figure II.16 La rubrique Admin interface settings.....	23

Figure II.17 la rubrique Logging.....	24
Figure II.18 déploiement de l'authentification via KERBEROS.....	24
Figure II.19 Paramétrage des débits de transfert des fichiers.....	25
Figure II.20 L'utilisation des règles pour Téléchargement des fichiers.....	25
Figure II.21 paramétrage de la compression des fichiers.....	26
Figure II.22 paramétrage du chiffrement des transmissions.....	27
Figure II.23 Création d'un nouveau certificat.....	27
Figure II.24 Paramètres pour la connexion FTP sur TLS.....	28
Figure II.25 paramètres pour bannir quelques utilisateurs.....	28
Figure II.26 Fenêtre principale de la gestion des Utilisateurs.....	29
Figure II.27 L'ajout d'un compte utilisateur.....	30
Figure II.28 Définition du mot de passe du nouvel utilisateur.....	30
Figure II.29 Paramétrage des restrictions sur le serveur FTP.....	31
Figure II.30 Description textuelle optionnelle du compte utilisateur.....	31
Figure II.31 paramétrage des répertoires et droits d'accès.....	32
Figure II.32 Les droits d'accès pour les fichiers et les répertoires.....	32
Figure II.33 Créer un nouveau répertoire.....	33
Figure II.34 Définir les limites de réception.....	34
Figure II.35 Définir des critères.....	34
Figure III.1 L'interface de Delphi.....	36
Figure III.2 Interface authentification.....	38
Figure III.3 Boite de dialogue.....	38
Figure III.4 La Fenêtre principal client FTP.....	39
Figure III.5L'option créée dossier.....	39
Figure III.6L'option renommée dossier ou fichier.....	40
Figure III.7L'option supprimée dossier.....	40

Figure III.8Message confirmation de la suppression.....	41
Figure III.9L'option supprimée fichier.....	41
Figure III.10Confirmation de la suppression du fichier.....	42
Figure III.11L'option Télécharger.....	42
Figure III.12L'option Télécharger.....	43
Figure III.13L'option envoyée .....	44
Figure III.14L'option envoyée.....	44
Conclusion.....	45

# *Liste des tableaux*

Tableau I.1 code réponse premier chiffre .....11

Tableau I.2 code réponse second chiffre .....12

# *Introduction générale*

De nombreuses communications sur un réseau informatique sont basées sur le modèle de client/serveur. Des machines faisant partie du réseau dites client contactent un serveur, une autre machine généralement très puissante en terme de capacités d'entrée-sortie. Dans ce modèle, le client demande des services toujours fournis par le serveur. Ces services concernent majoritairement le transfert de données telles que des fichiers, ou des services de connexion, ...etc.

Ces services sont exploités par des programmes, appelés programmes client, s'exécutant sur les machines clients. On parle ainsi de client lorsque l'on désigne un programme tournant sur une machine client capable de traiter des informations qu'il récupère auprès d'un serveur.

FTP est un protocole qui obéit au modèle client /serveur. Le service fourni concerne le transfert de fichiers via des fonctionnalités bien distinguées.

Les spécifications du modèle FTP sont publiées depuis le RFC114 (Request For Comments) et ont connues depuis une évolution constante.

Notre projet consiste à réaliser un client FTP compatible au serveur FileZilla Server en vue de bien comprendre son mécanisme et de l'implémenter via l'environnement de développement offert par Delphi.

Notre travail est synthétisé sur le présent document, organisé en trois chapitres. Le premier chapitre est consacré au protocole FTP et les commandes et services offerts par ce protocole. Le deuxième chapitre est consacré à l'installation et la configuration du serveur FileZilla Server. Quant au troisième chapitre, il est consacré à la phase d'implémentation dans laquelle on présentera notre client et l'environnement de programmation Delphi7 utilisé pour l'implémentation. Nous terminons enfin le mémoire par une conclusion générale du travail présenté.

*Chapitre I*  
*Le protocole FTP*

## Introduction

De nombreuses communications sur réseau informatique sont basées sur l'architecture de client/serveur. Comme son nom l'indique, cette architecture offre des services, fournis par des machines Serveur, destinés à des machines Client. Pour bénéficier de ces services, un Client envoie des requêtes au Serveur.

Vus d'un niveau applicatifs, les services sont exploités par des programmes, appelés programmes Client, s'exécutant sur les machines Clients et capable de traiter des informations qu'il récupère auprès de machines Serveur. On parle ainsi de client FTP qui traite des fichiers, de client de messagerie qui traite du courrier électronique. De même, un serveur peut être spécialisé en serveur FTP, de messagerie électronique.

## I. Présentation du protocole FTP

Le FTP (FILE Transfer Protocol ou protocole de transfert de fichiers) est un protocole de communication qui obéit au modèle Client/serveur (**figure I.1**), destiné à l'échange de fichiers sur un réseau TCP/IP. Il intervient au niveau de la couche application du modèle OSI (couche n°7) et utilise TCP comme protocole de transport. Il permet de recopier un fichier complet d'un ordinateur vers un autre ordinateur du réseau, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur.

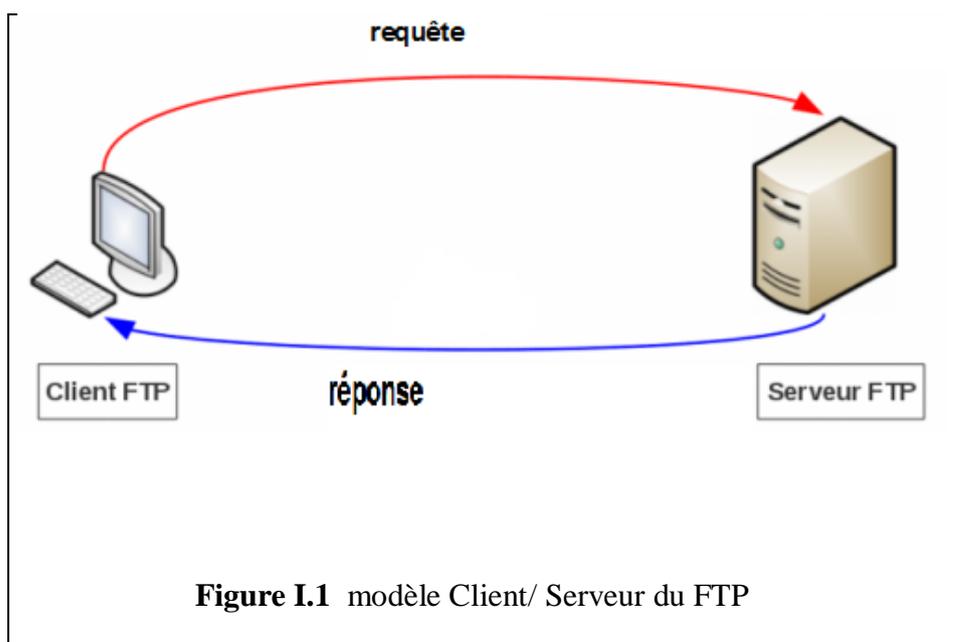
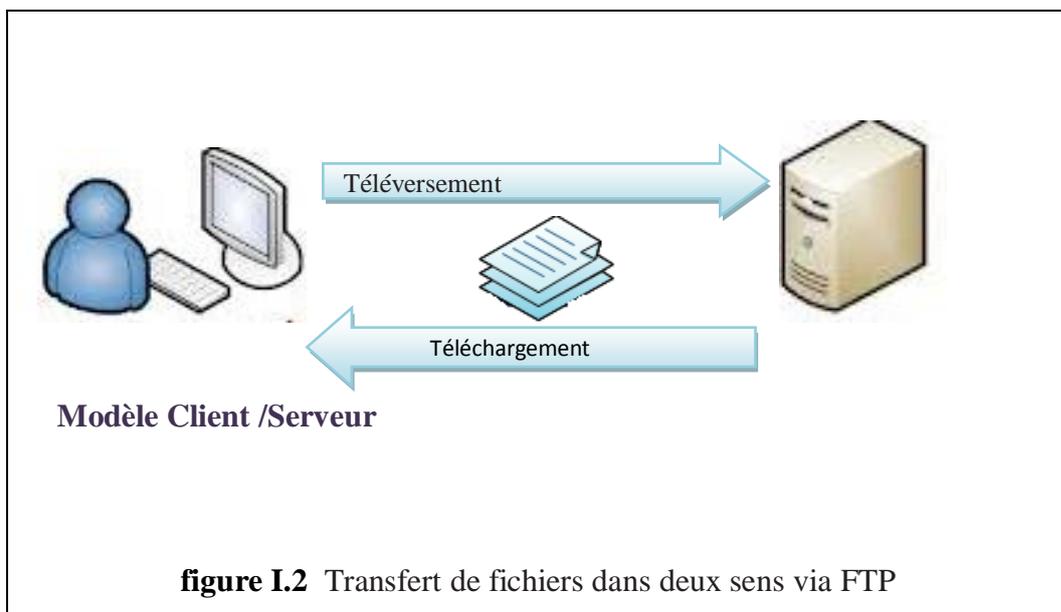


Figure I.1 modèle Client/ Serveur du FTP

Le transfert de fichiers via FTP se fait dans deux sens (**figure I.2**) :

- Download ou Téléchargement, lorsque l'on récupère des informations depuis le serveur.
- Upload ou Téléversement, lorsque l'on dépose des données sur le serveur, par exemple pour la mise à jour des pages web personnelles.



Parmi les principaux apports du protocole FTP:

- Permettre un partage de fichiers entre machines distantes
- Permettre une indépendance aux systèmes de fichiers des machines clientes et serveur
- Permettre de transférer des données de manière efficace [1].

## **II. Caractéristiques fonctionnelles de FTP**

Dans ce qui suit, nous mettons le point sur les fonctionnalités de FTP qui le caractérisent.

### **II.1. Canaux de transmission de FTP**

Le protocole FTP a besoin de deux canaux pour le fonctionnement (**figure I.3**) : un canal de contrôle et un canal pour le transfert des données.

**a) Le canal de contrôle**

Il sert, d'une part, à envoyer les commandes par le client vers le serveur pour accéder et pouvoir manipuler les fichiers sur le serveur (transfert, suppression de fichiers, renommée, liste des fichiers...). D'autre part, il véhicule les réponses du serveur.

Les commandes FTP ainsi que les réponses sont des codes de contrôle suivis de paramètres qui servent à contrôler l'ouverture, la fermeture d'une session FTP et le transfert des données durant la vie d'une session.

**b) Le canal de données**

Ce canal sert à échanger les données (contenu des fichiers, liste de fichiers) entre le serveur et le client.

**II.2. Processus de transmission de FTP**

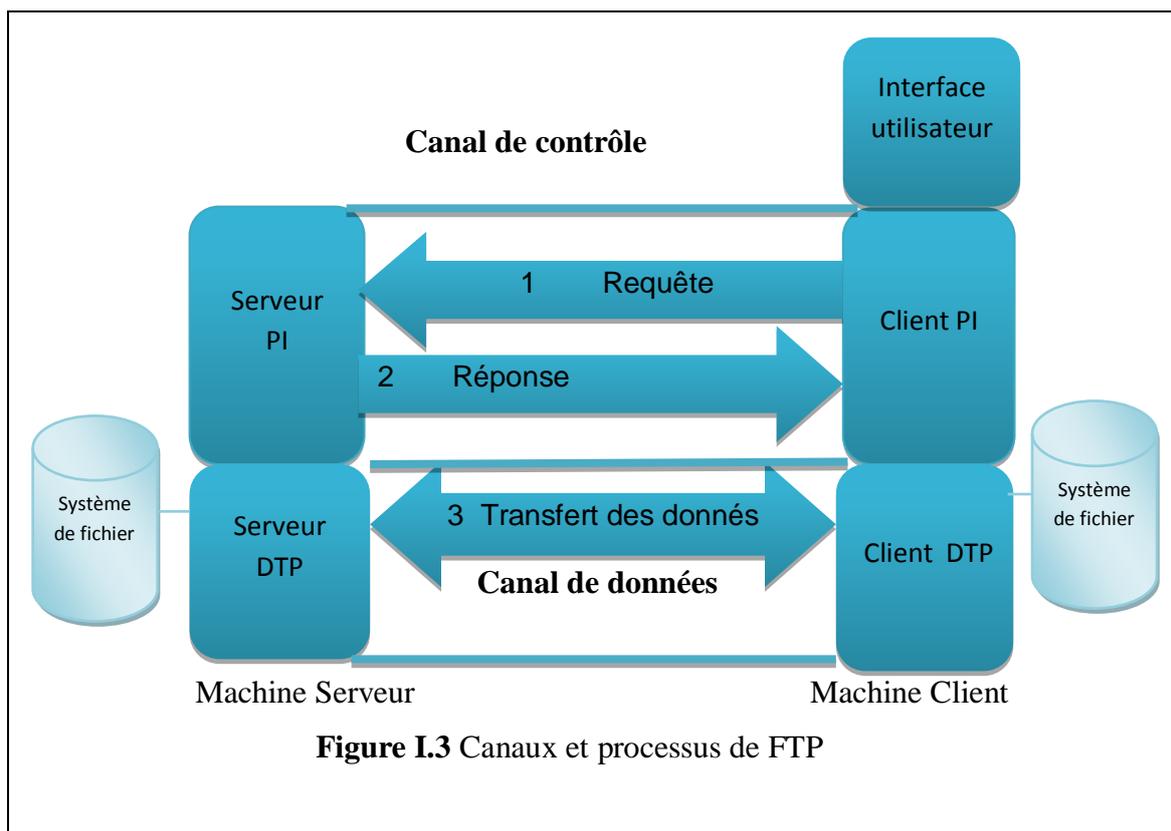
FTP définit pour chaque canal de transmission, un processus permettant de le gérer. Il s'agit des processus DTP et PI.

**a) Le DTP** (*Data Transfer Processus*) est le processus chargé d'établir la connexion et de gérer le canal de données. Le DTP a deux côtés : Côté serveur est appelé *SERVER-DTP*, le DTP côté client est appelé *USER-DTP*.

**b) Le PI** (*Protocol Interpreter*) est l'interpréteur de protocole permettant de commander le DTP à l'aide des commandes reçues sur le canal de contrôle.

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*.

Le *USER-PI* 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 se présente.



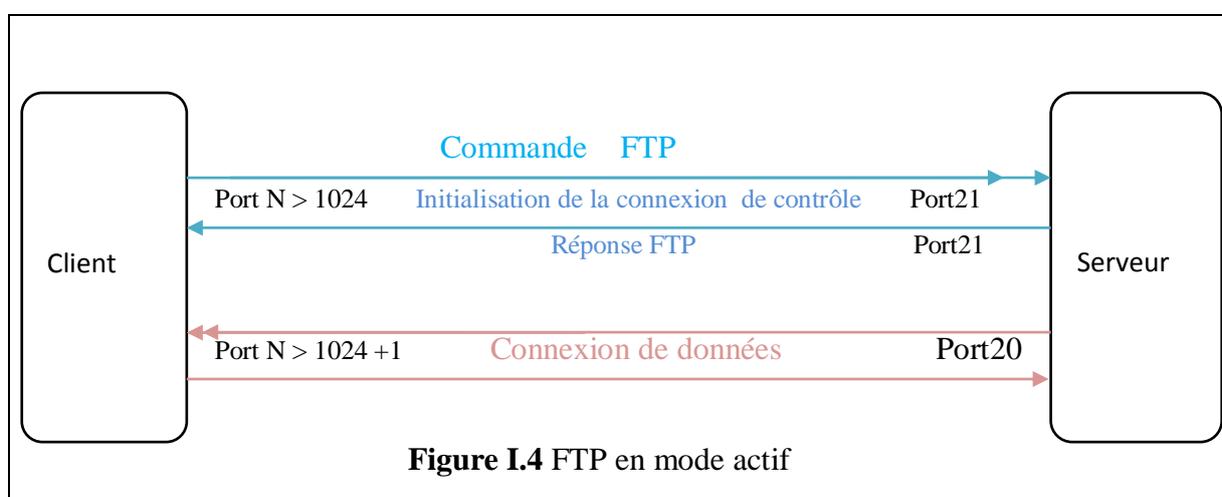
Comme indiqué sur la figure 3, lors de la connexion d'un client FTP à un serveur FTP, l'USER-PI initie la connexion au serveur. Le client envoie des commandes FTP au serveur, et indique le port qu'il utilise. Le serveur interprète ces commandes, pilote son DTP, puis renvoie une réponse standard. Lorsque la connexion est établie. Le client DTP écoute alors sur le port spécifié les données en provenance du serveur. L'échange des données entre le client et le serveur (sur le canal de données) se fait en deux modes différents : mode actif et le mode passif.

Le processus de contrôle FTP reste actif pendant toute la session, en revanche, le processus de transfert n'existe que le temps de l'envoi des données. Cette séparation en 2 circuits garantit qu'un canal restera ouvert pour transmettre les messages d'erreur si le processus de transfert des données est perdu. Le transfert est considéré comme complet quand le processus de transfert des données est fermé. Le résultat est donc que chaque fois qu'un fichier a été envoyé, le processus de transfert se ferme et celui de contrôle en ouvre un autre. Si une erreur provoque la fermeture du transfert, le module de contrôle considérera le fichier comme envoyé et un fragment du fichier restera dans le système de fichiers du destinataire.

### II.3. Les modes de transfert des données

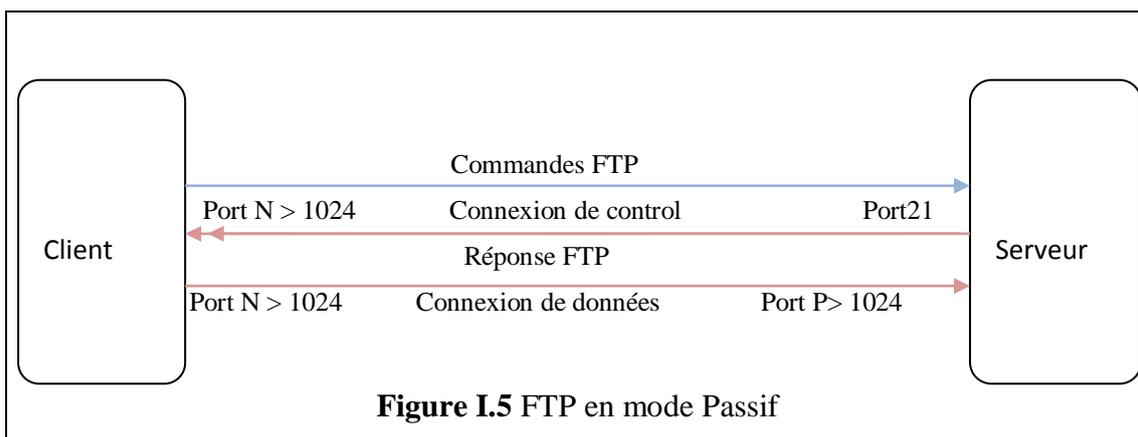
#### a) Mode Actif

Dans ce mode (**figure I.4**), le client choisit un port  $N > 1024$  sur lequel il initie la connexion de contrôle avec le serveur FTP. Ce dernier utilise le port 21 pour ouvrir la connexion de contrôle. Le client transmet le port qu'il utilise pour la connexion des données et **indique également son adresse IP** en utilisant la commande **PORT N+1**. Ainsi, il peut recevoir, sur ce port N+1, les éventuelles données provenant du serveur FTP **sur le port 20**.



#### b) Mode Passif

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). Cependant, 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 [2].



#### II.4 .Les commandes internes de FTP

Une session FTP est enclenchée après un dialogue en alternance entre le client et le serveur et qui véhicule des commandes qui servent à contrôler la session.

Ces commandes ont des codes (en minuscules ou en majuscules) de contrôle suivis de paramètres FTP qui précisent :

- Le port utilisé (21,20, ...)
- Le mode de transfert des données (mode binaire, mode ascii).
- La structure des données (enregistrement, fichier...)
- La nature de l'action à effectuer.

On distingue trois types de commandes FTP :

- Les commandes de contrôle d'accès
- Les commandes du paramétrage de transfert
- Les commandes de service FTP

##### II.4.1 Les commandes de contrôle d'accès

La commande **USER** permet d'identifier l'utilisateur. Cette identification est nécessaire pour établir une communication sur le canal de données.

La commande **PASS** spécifiant le mot de passe de l'utilisateur, elle doit être immédiatement précédée de la commande **USER**. Il revient au client de masquer l'affichage de cette commande pour des raisons de sécurité.

La commande **ACCT** 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.

La commande **CWD** (Change Working Directory) permet de changer le répertoire courant, elle est nécessaire le chemin d'accès au répertoire à atteindre comme argument.

La commande **CDUP** (Change to Parent Directory) permet de remonter au répertoire parent. Elle a été introduite pour remédier aux problèmes de nommage de répertoire parent selon les systèmes.

La commande **QUIT** 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.

#### **II.4.2 Les commande de paramétrage du transfert**

La commande **PORT** permettant de préciser le numéro de port à utiliser.

La commande **PASV** permettant d'indiquer au serveur DTP de se mettre en attente une connexion sur un port spécifique choisis aléatoirement parmi les ports disponible.

La réponse à cette commande est l'adresse IP de la machine et le port.

#### **II.4.3. Les Commandes de service FTP**

La commande **STOR** 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.

La commande **STOU** 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.

La commande **APPE** Grâce à cette commande (*append*) 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

La commande **ALLO** (*allocate*) demande au serveur de prévoir un espace de stockage suffisant pour contenir le fichier dont le nom est passé en argument.

La commande **REST** (*restart*) 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.

La commande **ABOR** (*abort*) 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.

La commande **RNFR** (*renamefrom*) permet de renommer un fichier. Elle indique en paramètre le nom du fichier à renommer et doit être immédiatement suivie de la commande **RNTO**

La commande **RNTO** (*rename to*) permet de renommer un fichier. Elle indique en paramètre le nom du fichier à renommer et doit être immédiatement précédée de la commande **RNFR**

La commande **Delete** Cette commande (*Delete*) 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.

La commande **RMD** Cette commande (*Remove directory*) permet de supprimer un répertoire. Elle indique en paramètre le nom du répertoire à supprimer

La commande **MKD** Cette commande (*make directory*) permet de créer un répertoire. Elle indique en paramètre le nom du répertoire à créer

La commande **PWD** Cette commande (*printworking directory*) permet de renvoyer le chemin complet du répertoire courant

La commande **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

La commande **NLST** Cette commande (*Name liste*) permet d'envoyer la liste des fichiers et répertoires dans le répertoire courant

La commande **SITE** Cette commande (*site paramètres*) permet au serveur de proposer des services spécifiques, non définis dans le protocole FTP

La commande **SYST** Cette commande (*system*) permet d'envoyer des informations sur le serveur distant.

La commande **STAT** (*status*) 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

La commande **NOOP** (*no opérations*) permet de connaître l'ensemble des commandes comprises par le serveur. Les informations sont retournées sur le canal de contrôle Cette commande 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 [3].

### **III. Déploiement de FTP au niveau utilisateurs**

#### **III.1. Démarrer une session FTP**

FTP est fourni en standard sous diverses plateformes, dont UNIX, Windows9x, Windows, Linux...

La commande pour initier une session FTP est généralement la suivante :

**FTP nom\_du\_serveur** où Nom\_du\_serveur représente le nom (ou l'adresse IP) de la machine distante à laquelle on désire se connecter. Il faut bien évidemment que la machine cible soit un serveur FTP.

Le serveur demande à l'utilisateur de saisir le nom d'utilisateur (login ou ID) afin de définir des privilèges d'accès comme le droit d'écriture ou de lecture et sur quel répertoire. Après la validation, une ligne invite à saisir le mot de passe (password). Selon la configuration du serveur, cette ligne peut éventuellement faire apparaître le numéro de réponse 331.

Il se peut que le serveur soit publique, auquel cas le nom d'utilisateur est anonyme, il faudra donc rentrer comme login « anonymous ». Dans ce cas, le mot de passe peut contenir n'importe quel caractère puisqu'il n'a pas d'utilité à différencier les utilisateurs de ce login. De plus, ce type de serveurs anonymes ne permet généralement pas l'envoi de fichiers, mais uniquement la réception.

Dans cette étape d'initialisation de la connexion et lorsque l'utilisateur valide ces données (login et mot de passe), un certain nombre de lignes de texte apparaissent à l'écran : la première ligne signale à l'utilisateur qu'il est connecté à un serveur, les lignes suivantes constituent un message de bienvenue, pouvant indiquer le type de site FTP dont il est question.

Sous FTP, chaque ligne commence par un numéro indiquant un code relatif à un échec ou une réussite. Dans le cas du message de bienvenue, la ligne est par exemple précédée du nombre 220, qui signifie que le service est prêt pour le nouvel utilisateur. Dès que tous les transferts (en mode actif ou passif) sont terminés, l'utilisateur termine sa connexion [1].

### **III.2. Les commandes FTP externe (utilisateur)**

Une fois connectée le site FTP attend des commandes provenant des utilisateurs décrivant les actions à effectuer. Il y a six commandes de base communes à tous les programmes FTP : *OPEN*, *CLOSE*, *CD*, *DIR*, *GET*, *PUT*.

- ✓ La première opération que nous devons effectuer est donc d'ouvrir un canal de communication entre notre ordinateur et le serveur FTP désigné par l'utilisation de la commande *OPEN*. Cette opération est identique à une connexion avec un serveur.
- ✓ La commande *CLOSE* est probablement la plus facile à utiliser de toutes les commandes FTP. Elle indique simplement à l'ordinateur éloigné que nous avons terminé et que nous désirons couper le canal de communication.
- ✓ La Commande *CD* est utilisée pour le 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.
- ✓ Cette commande *DIR* fonctionne de la même manière que dans la version MSDOS. Elle permet d'obtenir la liste du répertoire actuellement sélectionné.
- ✓ *GET* est la commande au cœur du protocole FTP utilisée nécessairement pour extraire un fichier de l'ordinateur distant et le stocker dans l'ordinateur de l'utilisateur. Certains programmes FTP avec fenêtres permettent d'utiliser une commande *GET* implicite en permettant à l'utilisateur de cliquer-tràîner un fichier d'une fenêtre à l'autre.
- ✓ *PUT* est la Commande inverse de *GET*, cette commande permet d'envoyer un ou plusieurs fichiers de notre ordinateur vers un ordinateur éloigné

### **III.3 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)

**a) Premier chiffre**

<b>Chiffre</b>	<b>Signification</b>	<b>Description</b>
<b>1yz</b>	Réponse préliminaire positive	L'action demandée est en cours de réalisation, une seconde réponse doit être obtenue avant d'envoyer une deuxième commande
<b>2yz</b>	Réponse positive de réalisation	L'action demandée a été réalisée, une nouvelle commande peut être envoyée
<b>3yz</b>	Réponse intermédiaire positive	L'action demandée est temporairement suspendue. Des informations supplémentaires sont attendues de la part du client
<b>4yz</b>	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
<b>5yz</b>	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

**Tableau I.1** code réponse premier chiffre

## b) Le 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

Tableau I.2 code réponse second chiffre .

## c) Troisième chiffre

L'interprétation du 3ème chiffre dépend du contexte défini par les deux premiers.

**Par exemple :**

Le chiffre 220 pour une connexion établie avec succès.

Le chiffre 214 réponse à une demande d'information antérieure

Le chiffre 221 acquittement d'une demande de déconnexion.

Le chiffre 530 signifie que le serveur demande une authentification avant de répondre à toute demande [3].

**Conclusion :**

Dans ce chapitre nous avons présenté le protocole FTP, les concepts de base pour établir une connexion entre le client et le serveur ftp, les commandes les plus utilisées dans une session et les réponses FTP.

*Chapitre II Installation et  
Configuration de serveur  
FileZilla Server*

## Introduction

Il existe de nombreux logiciels freeware pour la création d'un serveur FTP, nous avons choisi FileZilla server (0.9.43) car c'est un logiciel serveur rapide et facile avec le support des commande standard de FTP, une interface propre et clair, une architecteur de système de fichier virtuelle, capacité de reprendre le téléchargement interrompu tant en download qu'en upload, Multi langue et beaucoup d'autres fonctions .

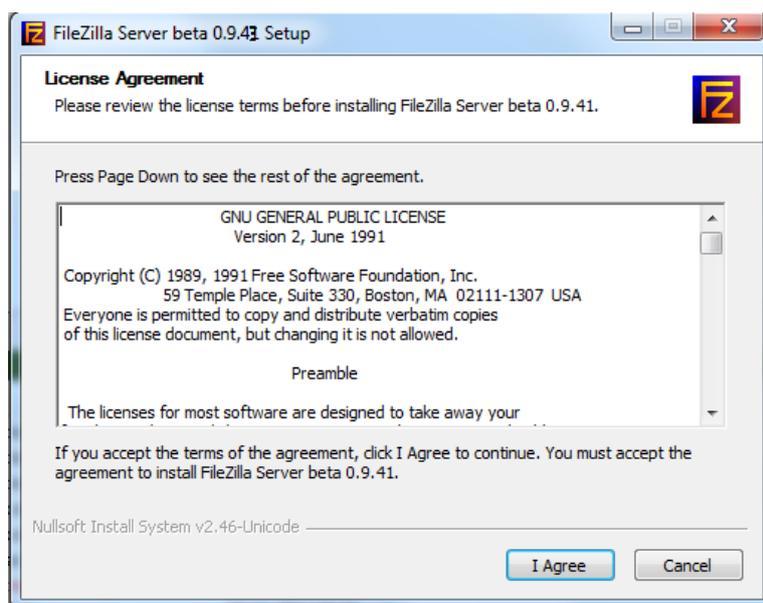
La première étape est évidemment le téléchargement de FileZilla server, ensuite il faut décompresser le fichier FileZilla server.zip et enfin lancer l'installation de FileZilla server .exe.

## I. Installations de FileZilla Server

Une fois le fichier téléchargé sur le site officiel :

<https://filezilla-project.org/download.php?type=server>, on passe à son installation.

En premier lieu quand l'installation est lancée, la licence d'utilisation s'affiche



**Figure II .1** La première étape de l'installation

En acceptant les termes du contrat, l'écran suivant s'affiche pour demander les paramètres d'installation (Type d'installation).

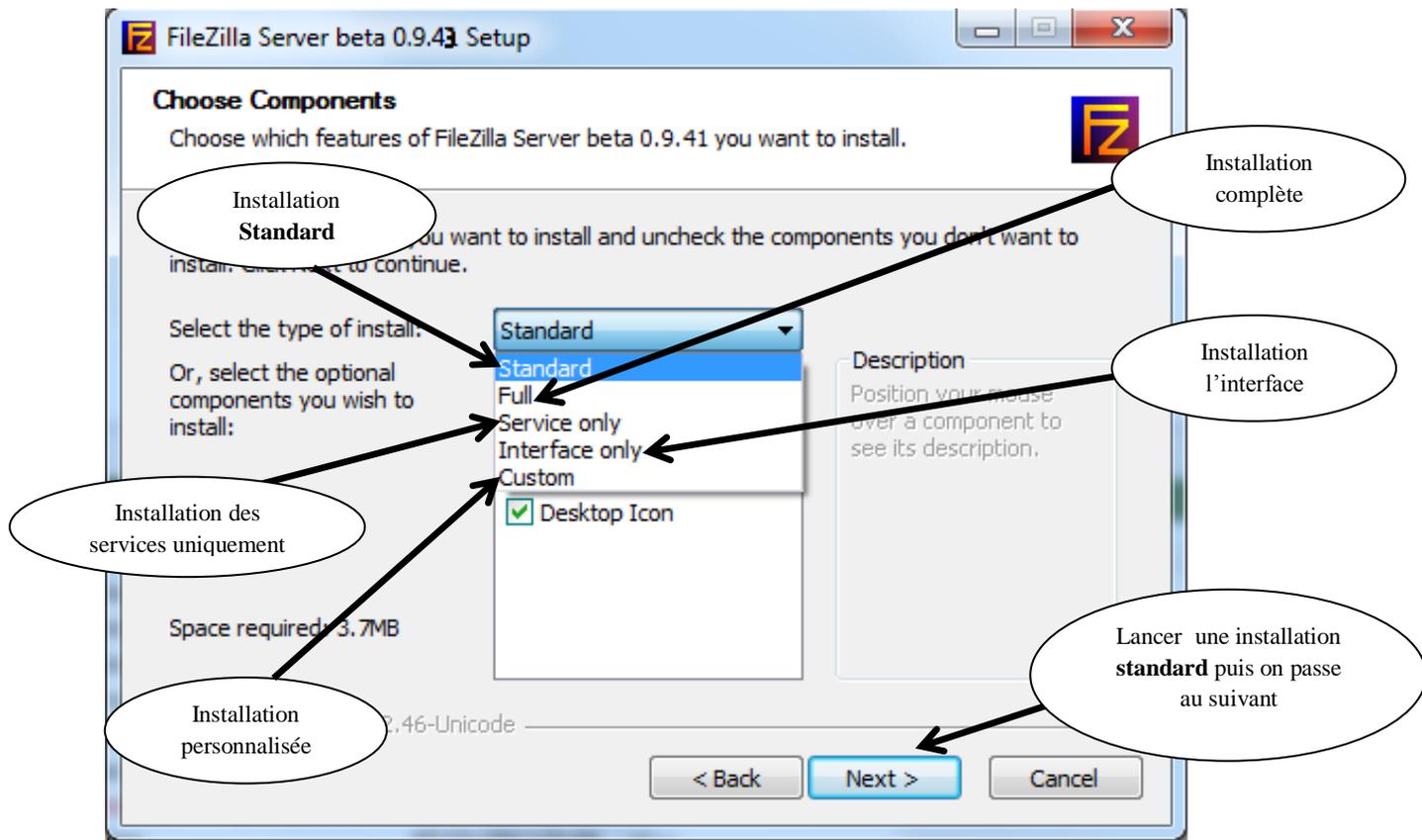
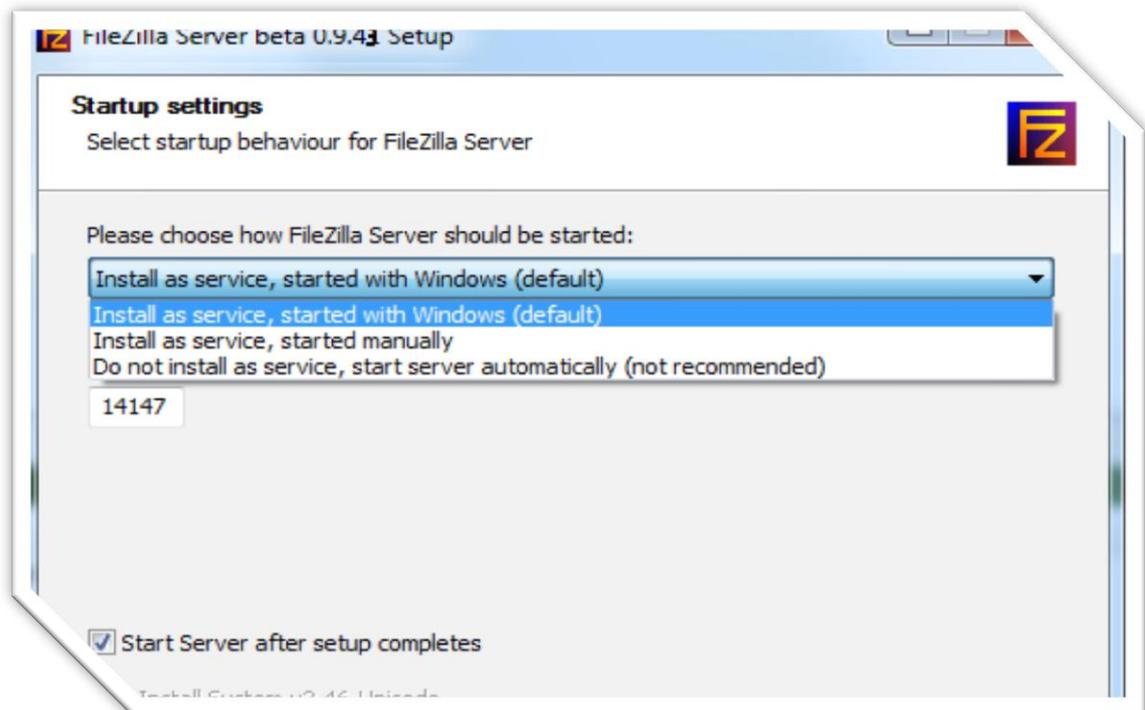


Figure II.2 Le choix d'un type d'installation

Une fois ces paramètres choisis, on devra identifier un répertoire d'installation. On atteint alors l'étape qui permet de choisir si l'on veut installer notre serveur FTP en tant que service de Windows qui démarre automatiquement ou manuellement.



**Figure II. 3** Le choix du type de démarrage du serveur FTP

Nous choisissons un démarrage manuel du serveur en tant que service de Windows, puisque nous ne l'utilisons que dans le cadre du projet et nous ne disposons de serveur physique dédié.

L'étape suivante concerne le démarrage de l'**interface d'administration**

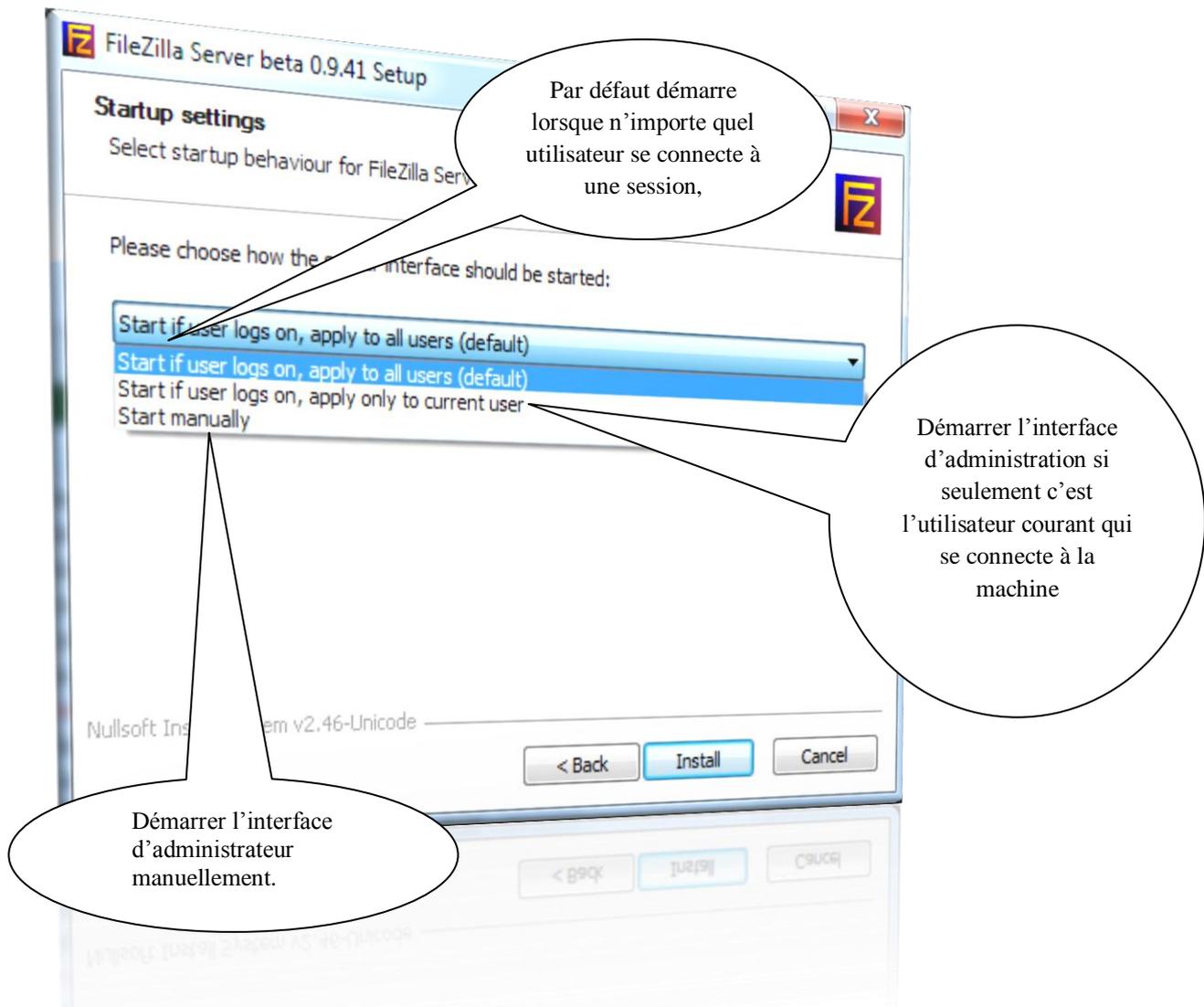
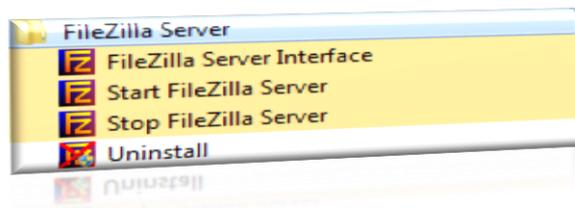


Figure II. 4 le choix du type de démarrage de l'interface d'administration

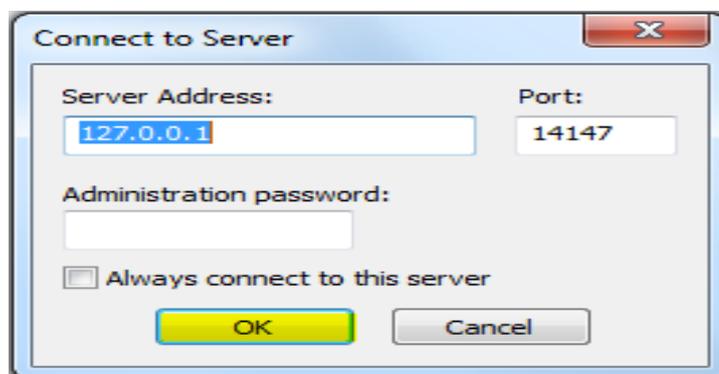
Nous sélectionnons la troisième option pour les mêmes raisons citées ci-dessus. Si l'installation s'est bien déroulée, on doit avoir le programme **FileZilla Server** dans le menu **Démarrer**

## II Accès à l'interface d'administration



**Figure II. 5** Interface administrateur du serveur

Pour pouvoir configurer le serveur FTP, il est nécessaire de lancer l'interface d'administration : FileZilla Server Interface (**figure II.5**) mais surtout de se connecter à l'application serveur via une adresse IP, un numéro de port (**figure II.6**) et en spécifiant le mot de passe.



**Figure II. 6** Interface pour spécifier les informations de connexion au serveur

L'interface d'administration se présente en une fenêtre (**figure II.7**) avec un message qui présente le créateur du logiciel **Tim Kosse** et un message d'authentification au départ, et qui contient l'évolution des différentes interactions entre clients et le serveur. Elle présente également les menus File, Server et Edit et une barre d'icônes dont nous présentons l'utilité dans ce qui suit.

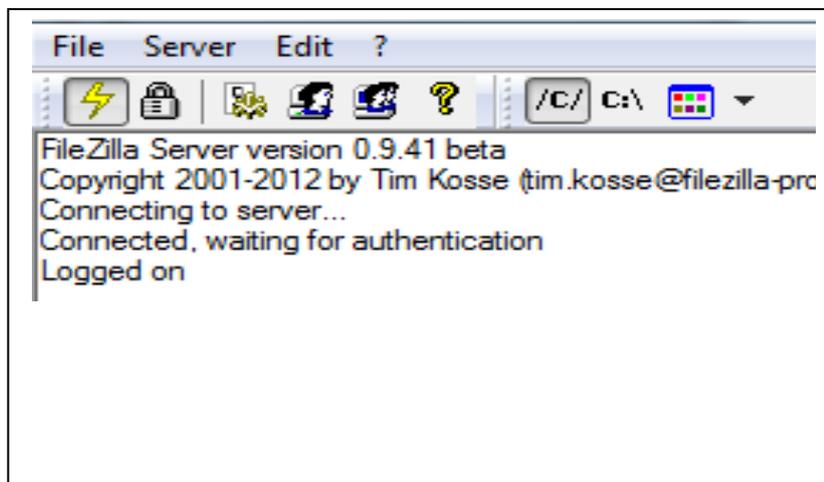


Figure II.7 L'interface d'administration

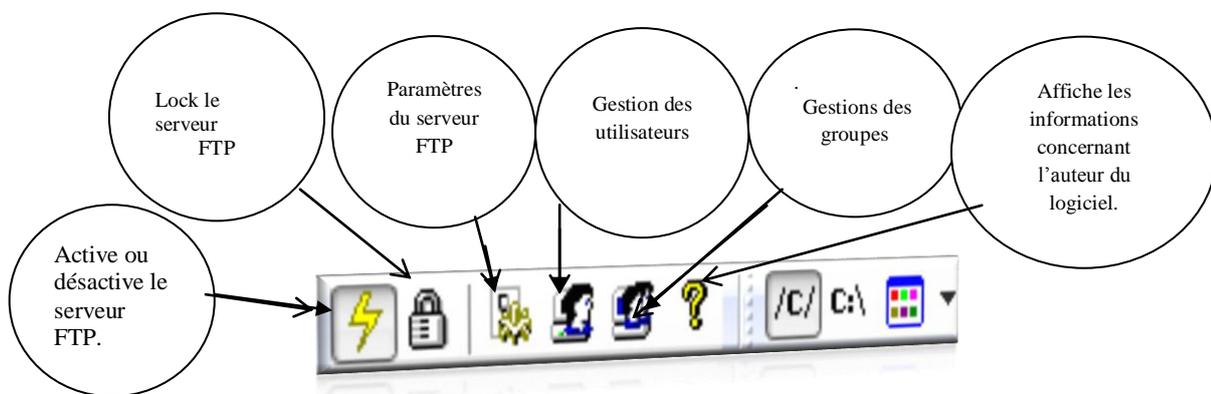


Figure II. 8 La barre des icônes

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

- Le menu FILE (Figure II.9)

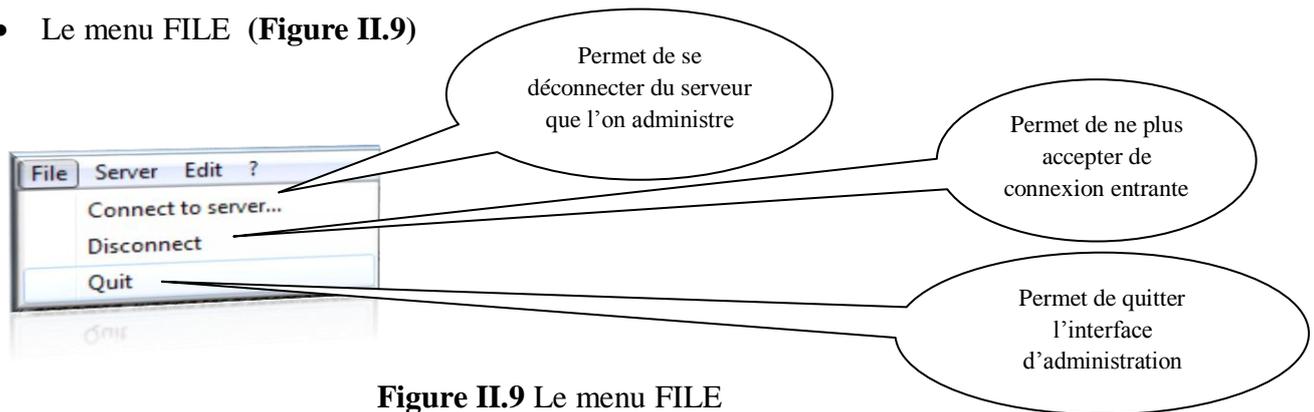


Figure II.9 Le menu FILE

- Le menu SERVER (Figure II.10)

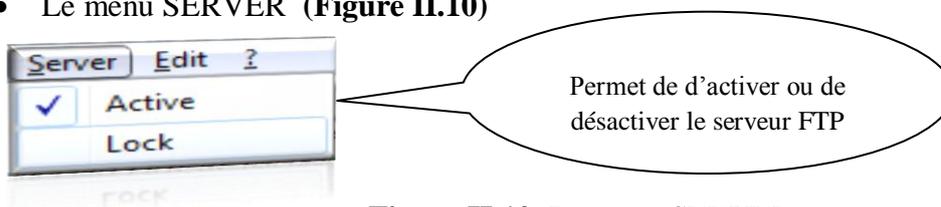


Figure II.10 Le menu SERVER

- Le menu Edit (Figure II.11)

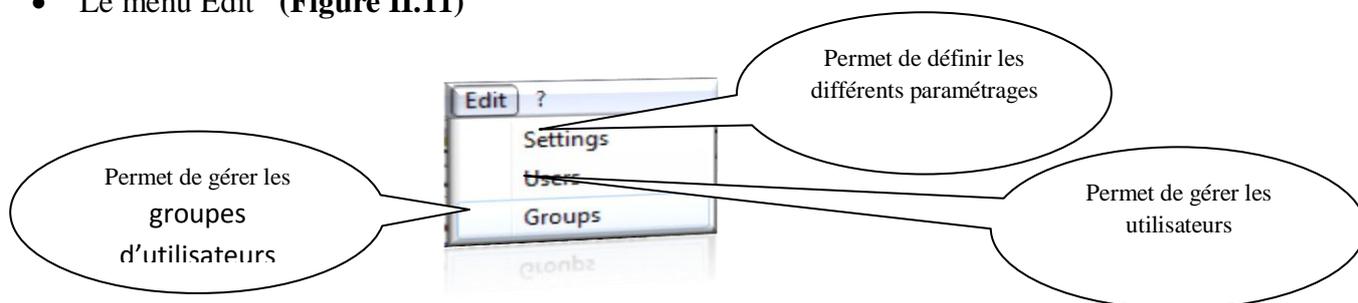


Figure II.11 Le menu Edit

### III. Les rubriques de configuration 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 II.12).

Un clic sur la commande **Settings** renvoie à la fenêtre suivante :

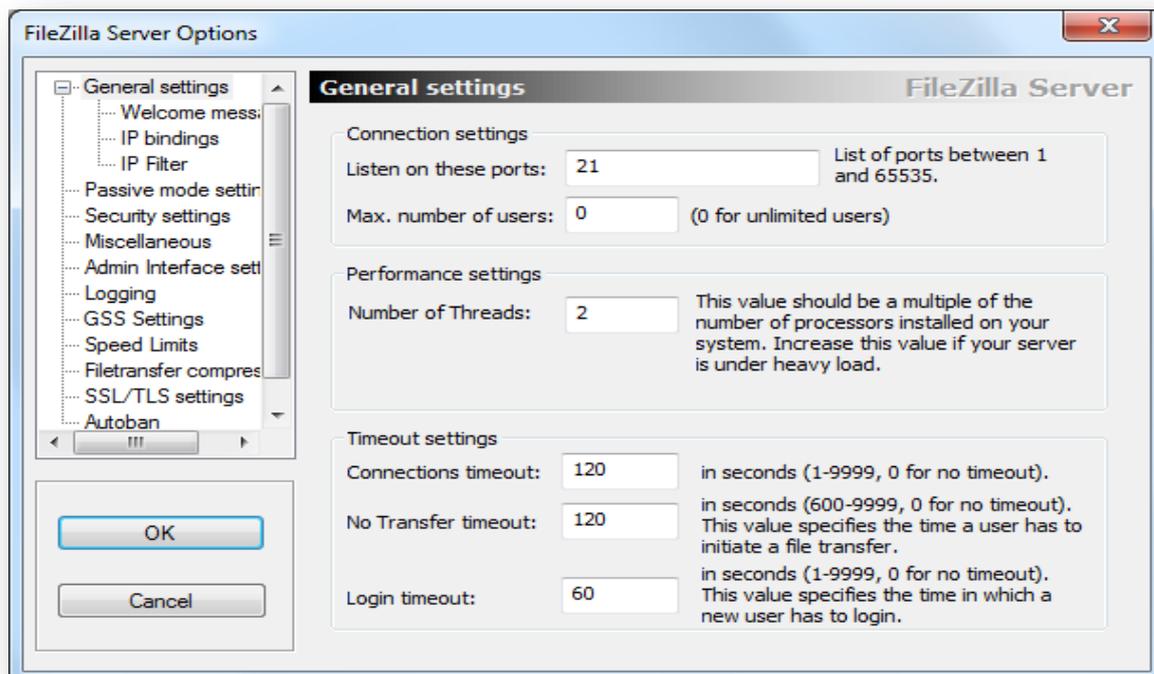


Figure II.12 Fenêtre de configuration du serveur FTP

### III.1. Rubrique de configuration générique

À gauche de cette fenêtre, à travers les commandes numérotées 1, 2 et 3 sur la figure, l'administrateur du serveur peut définir un message d'accueil lorsqu'un utilisateur se connecte au serveur FTP, il peut définir l'adresse IP utilisée par le serveur FTP pour fonctionner ou encore filtrer les adresses non souhaitées.

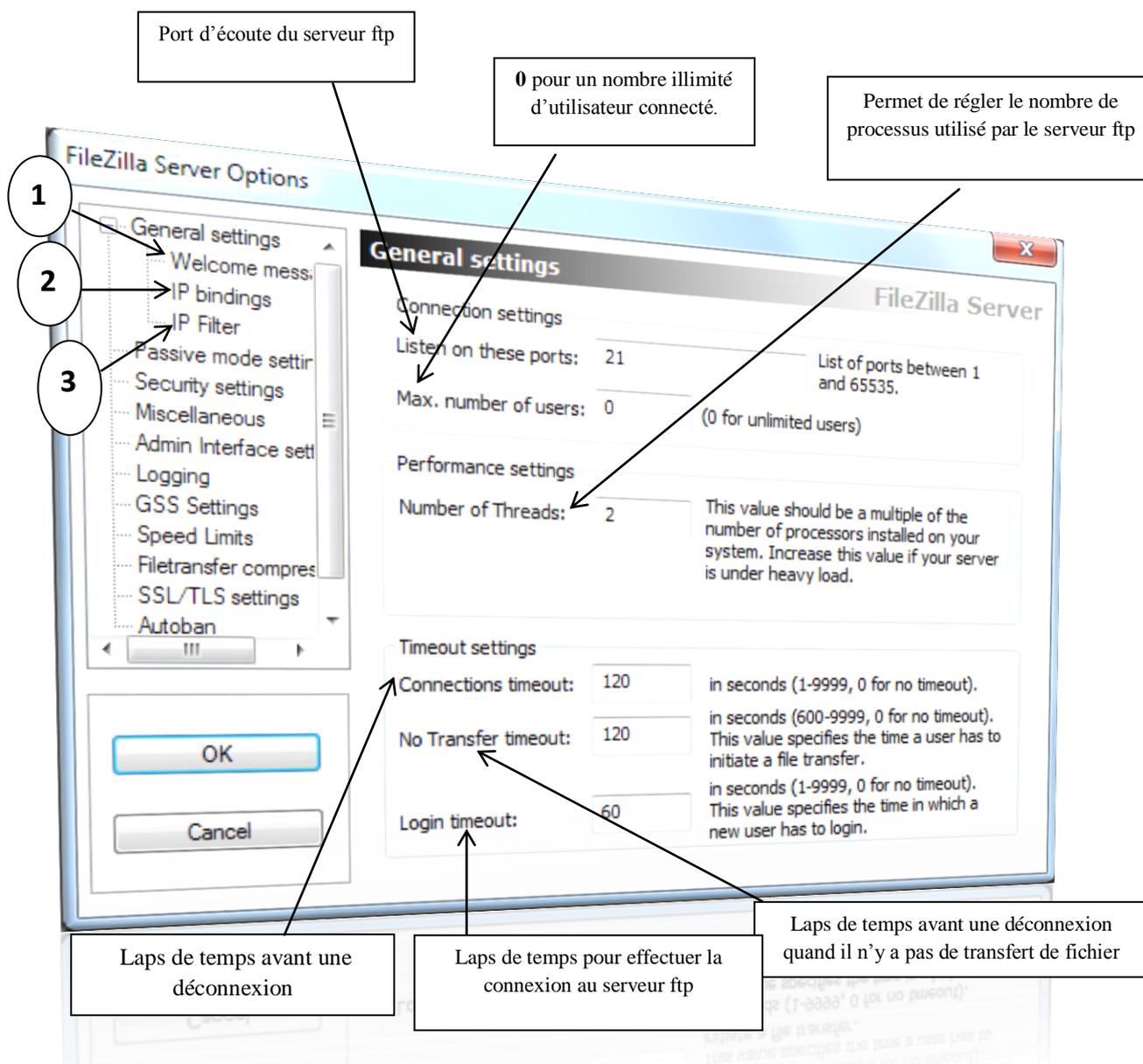


Figure II.13 Les options génériques de configuration du serveur

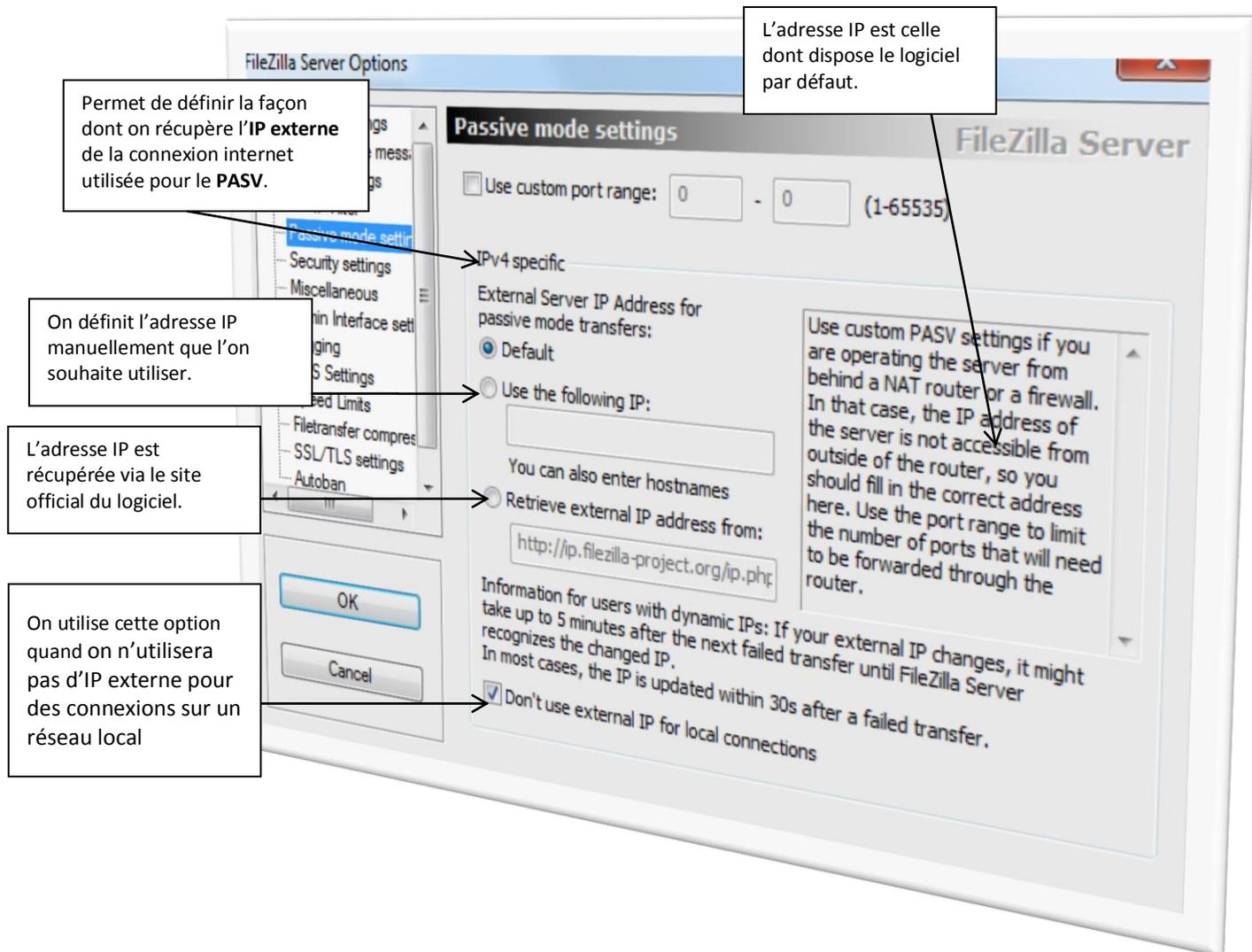
L'option **Hide welcome message in log** permet de ne pas afficher le message d'accueil dans le fichier de log.

Lorsque l'administrateur configure **IP bindings** à la valeur (\*), le serveur écoute sur toutes les IP assignées à une ou plusieurs cartes réseaux installées sur la machine : IP de la machine locale (127.0.0.1) et IP sur réseau (exemple : 192.168.1.2) .Si nous souhaitons personnaliser l'IP sur laquelle le serveur écoute, il suffit de remplacer \* par l'IP que nous souhaitons utiliser.

### III.2 Rubrique de configuration du mode passif

La fenêtre de configuration du serveur FTP (figure II.13) permet de paramétrer le mode passif de FTP à travers la commande **Passive mode settings**. La fenêtre qui

s'ouvre (**figure II.14**) permet de définir la plage pour les ports PASV (on coche l'option **Use custom port range**) qui seront utilisés par le serveur FTP. La commande PASV indique au serveur qu'il doit attendre passivement la connexion en écoutant un port TCP.



**Figure II.14** Fenêtre de paramétrage du mode passif

### III.3 Rubrique Miscellaneous (divers paramètres)

Cette rubrique permet de définir divers paramètres concernant l'affichage du mot de passe, la taille du buffer utilisée et le type d'adresses à considérer lors des transferts

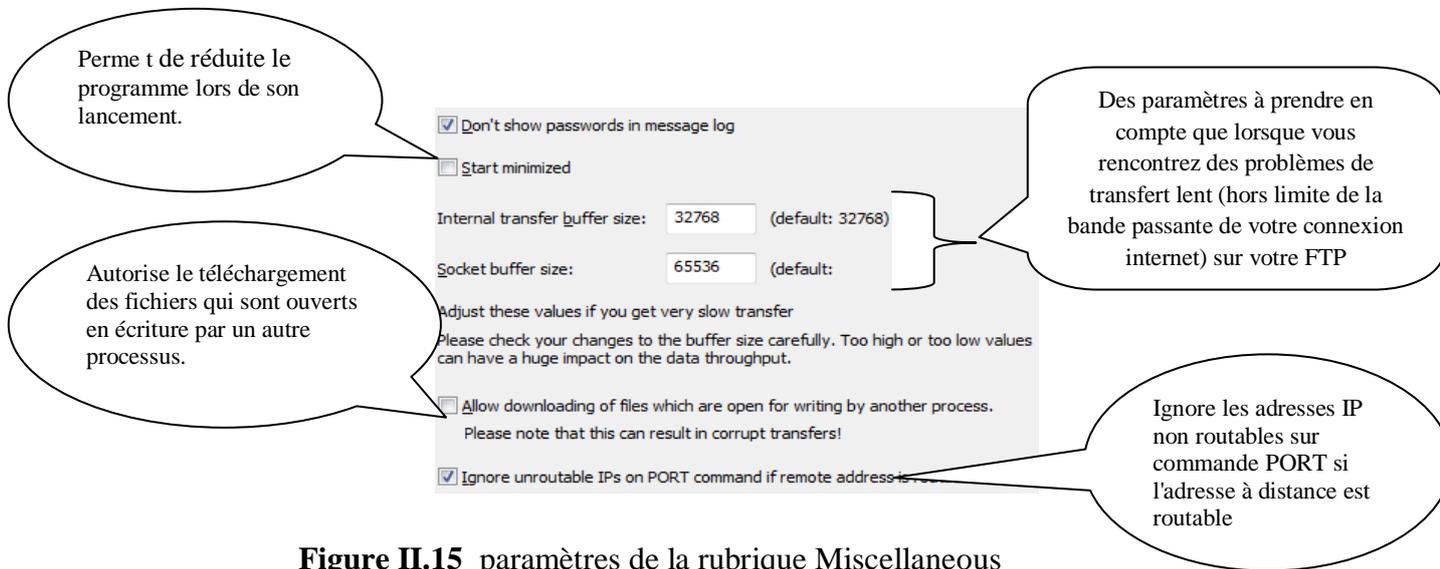


Figure II.15 paramètres de la rubrique Miscellaneous

### III.4. La rubrique Admin interface settings

Cette rubrique permet de définir les options liées à l'interface d'administration. On peut y définir le port d'administration compris entre 0 et 65535 (hormis le 20 et 21 utilisé par le serveur FTP) ainsi que d'autres paramètres (figure II.19).

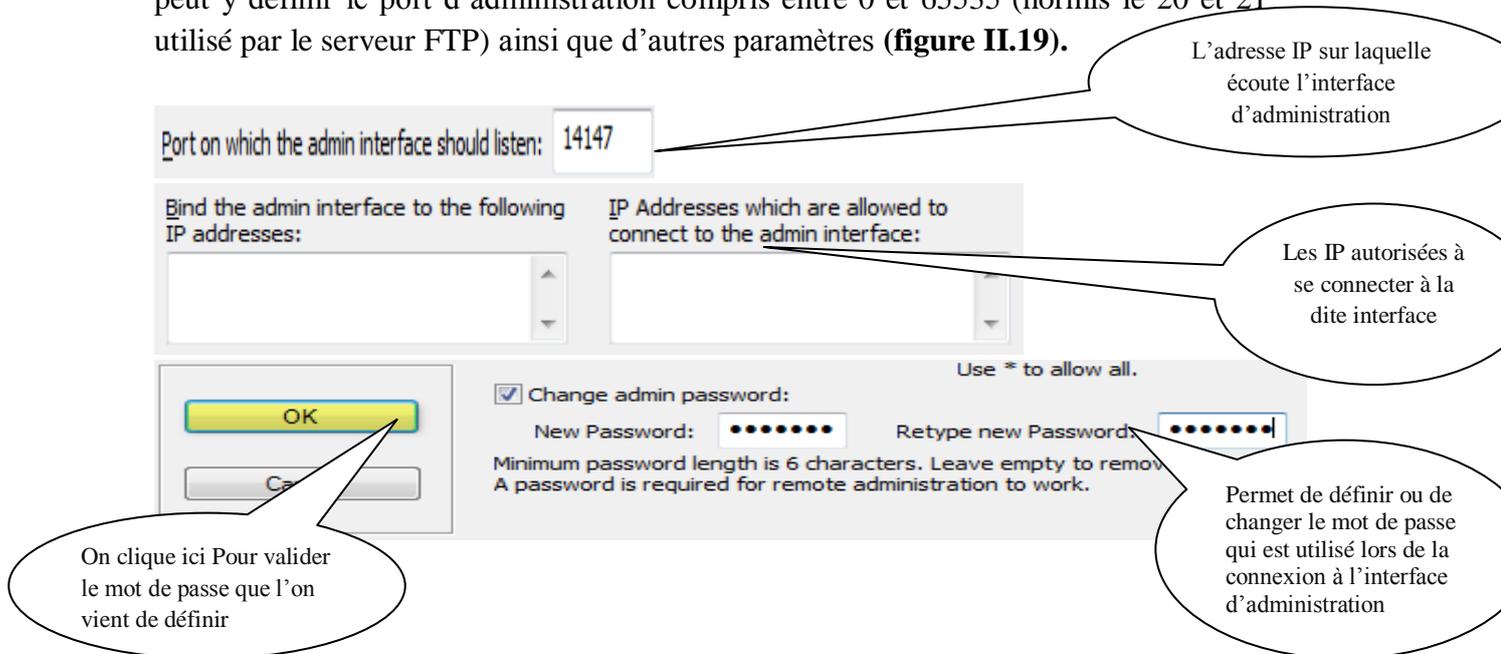


Figure II.16 La rubrique Admin interface settings

III.5. Rubrique 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. Pour ce faire, on coche l'option **Enable Logging to file**.

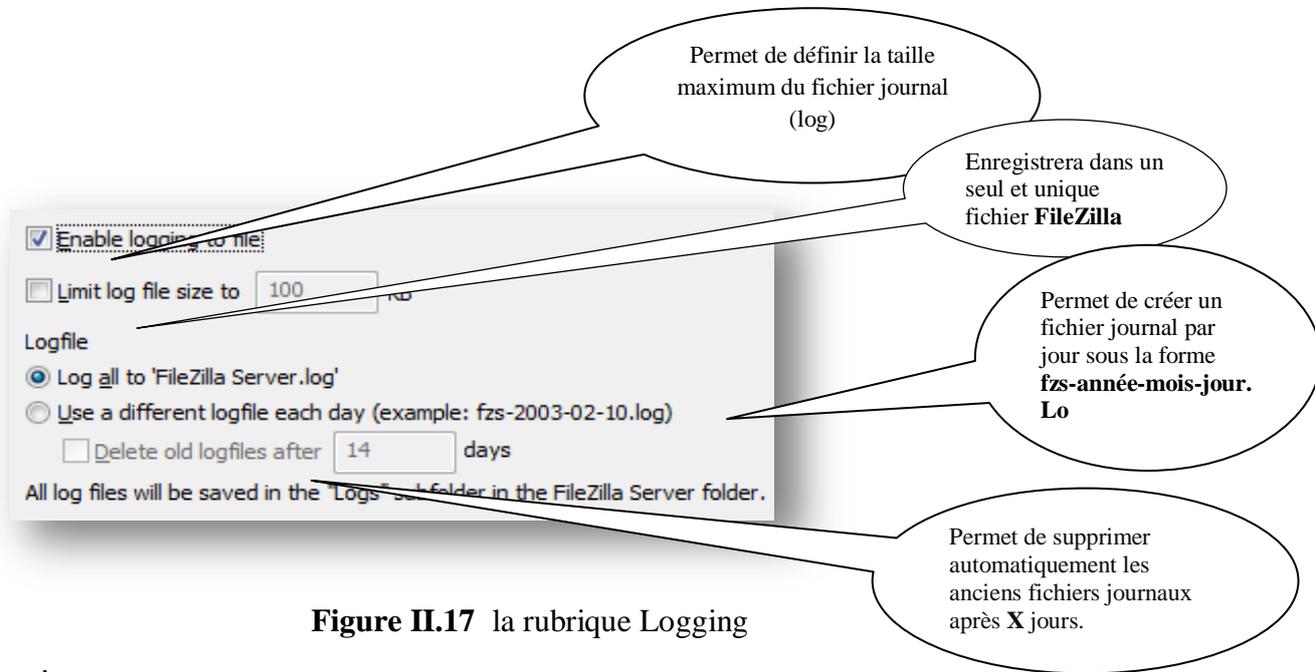


Figure II.17 la rubrique Logging

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**.

II.2.7. Rubrique GSS Settings

Cette rubrique permet de définir les options pour utiliser un serveur d'authentification **KERBEROS** pour assurer la sécurité lors des accès utilisateurs

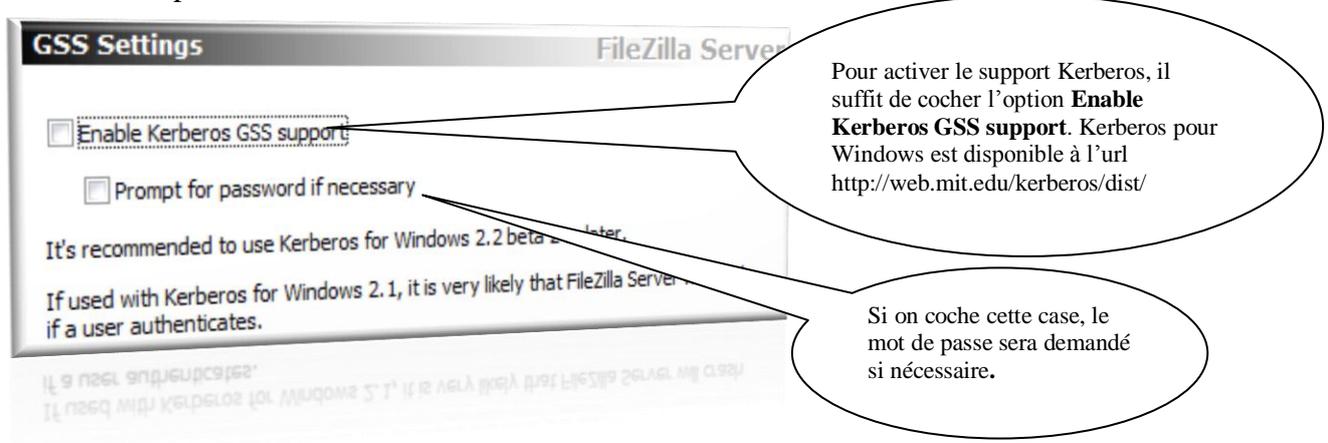


Figure II.18 déploiement de l'authentification via KERBEROS

### III.7 .La rubrique Speed Limits

Cette rubrique permet de définir les débits accordés lors téléchargement pour l’envoi ou la réception de fichier.

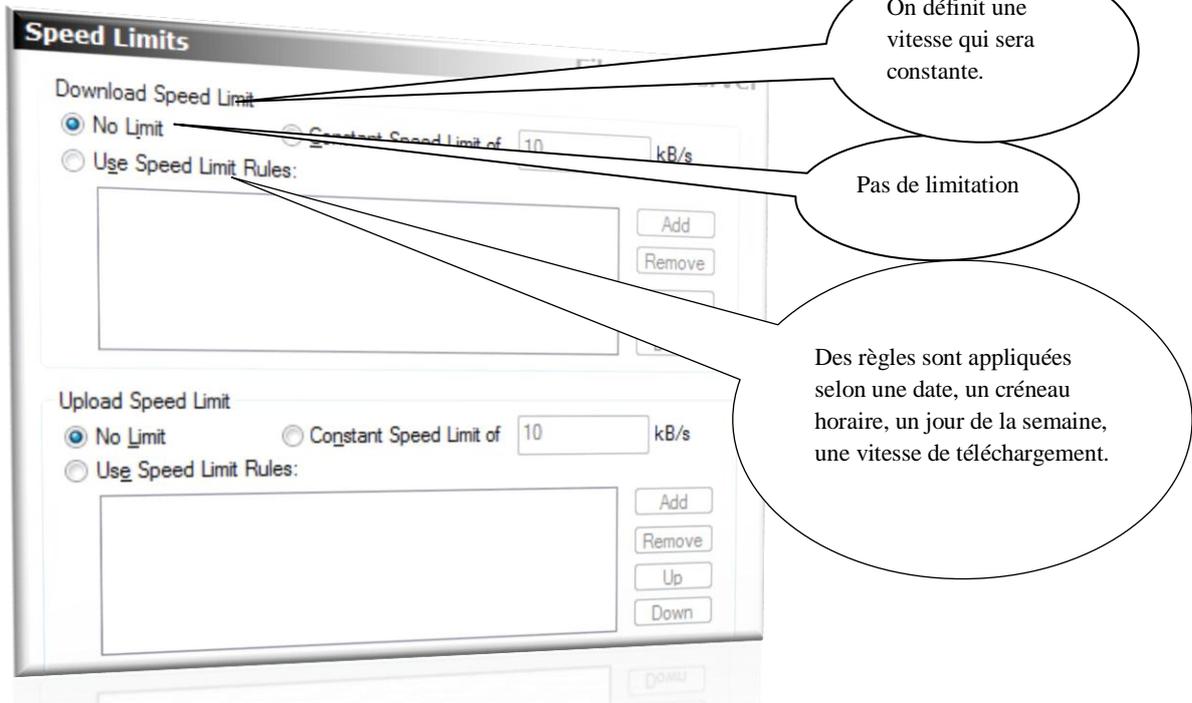


Figure II.19 Paramétrage des débits de transfert des fichiers

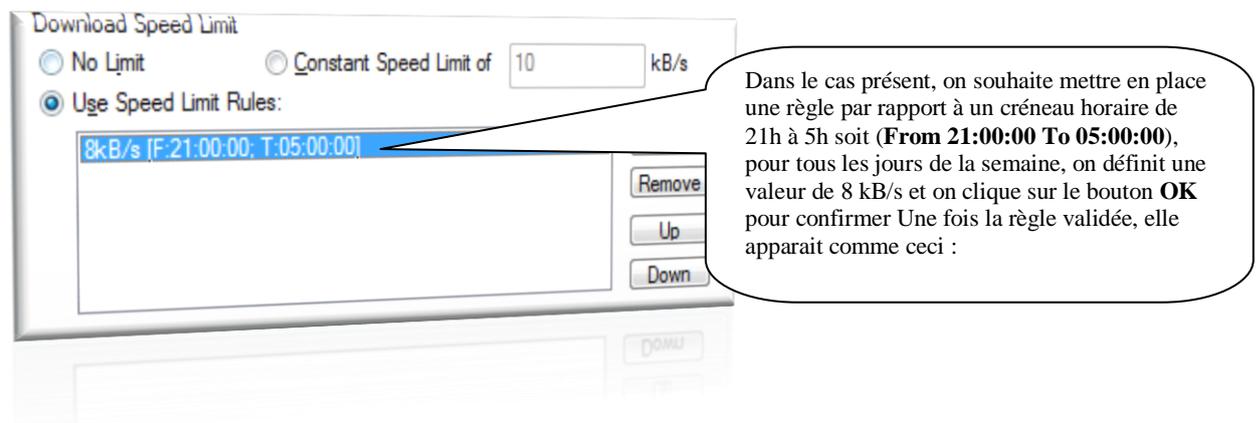


Figure II.20 L’utilisation des règles pour Téléchargement des fichiers

On peut mettre en place le même type de règle pour **Upload Speed Limit**.

### III.8. Rubrique File Transfert compression

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

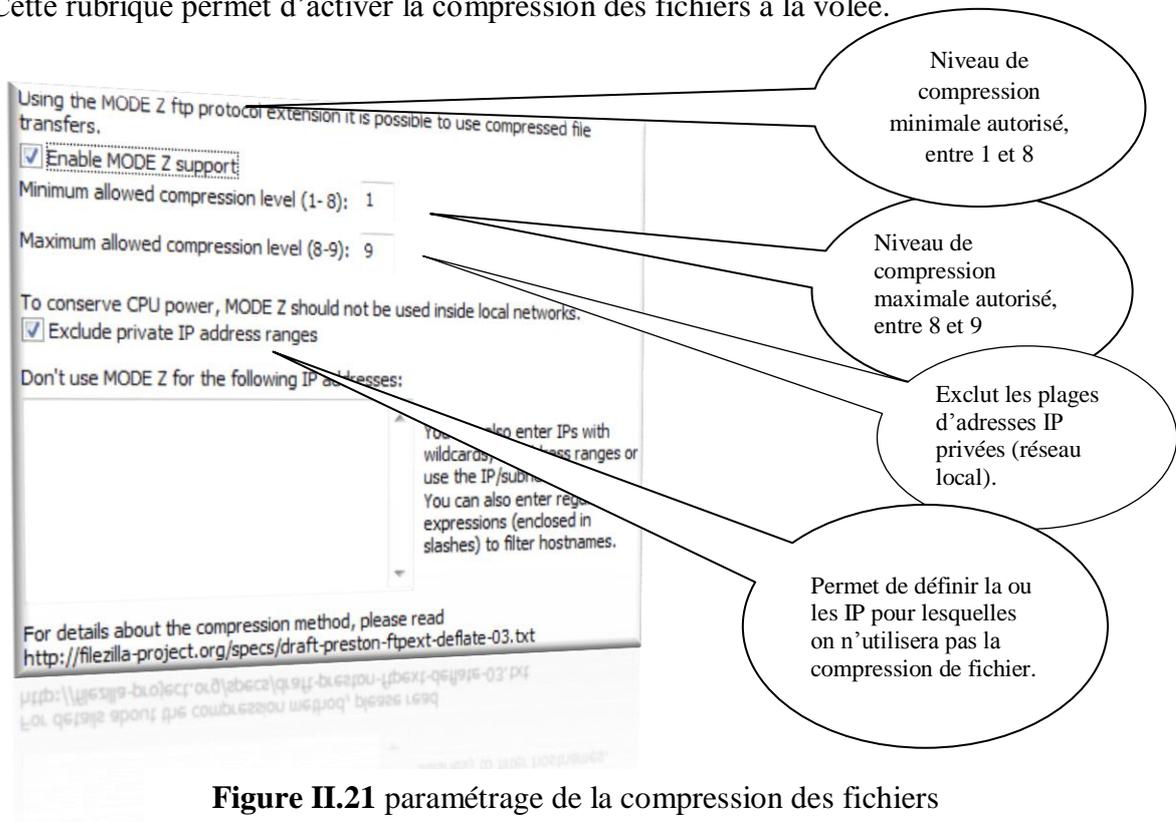
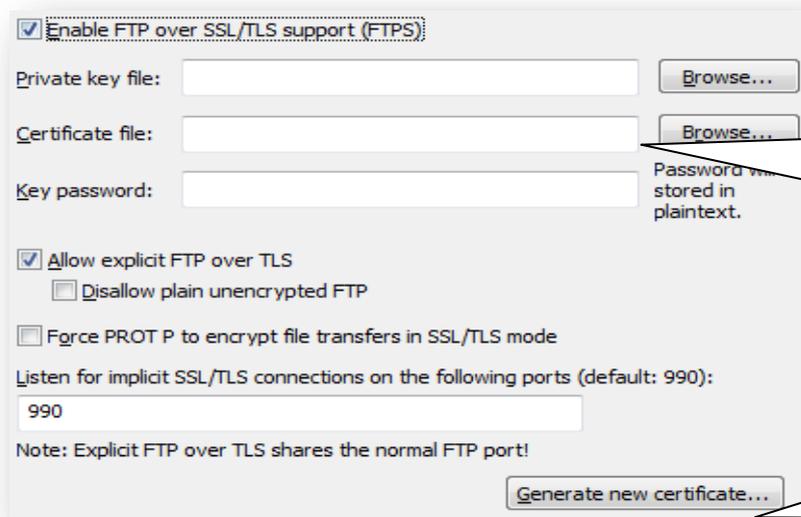


Figure II.21 paramétrage de la compression des fichiers

### III.9. Rubrique SSL/TLS settings

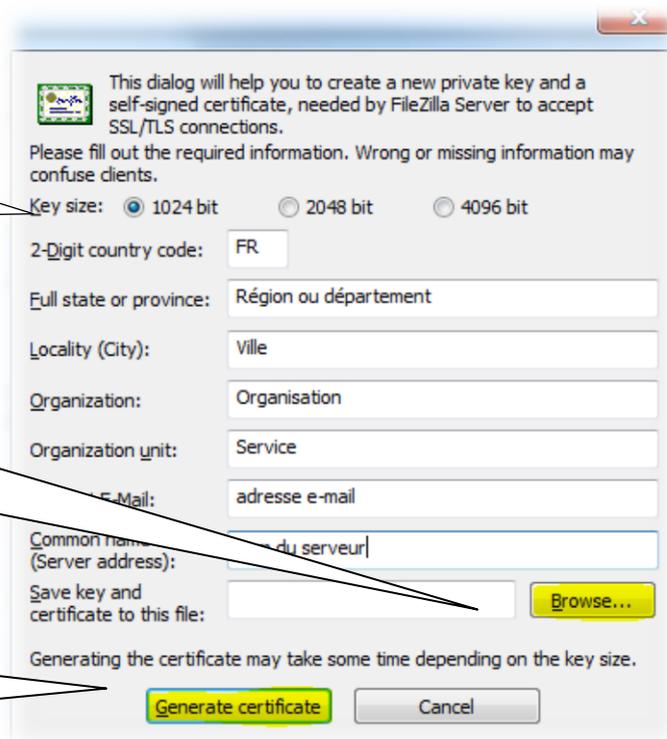
Cette rubrique permet d'utiliser les fonctions de chiffrement des transmissions. Pour ce faire, on active l'option **Enable FTP over SSL/TLS support (FTPS)** qui va sécuriser le FTP.



Les 2 champs 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**.

Pour cela, on clique sur le bouton **Generate new certificate** et on renseigne les différents champs

Figure II.22 paramétrage du chiffrement des transmissions



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**.

Une fois les informations rentrées, on clique ici pour choisir l'emplacement où seront enregistrés la clé et le certificat

Pour finir on clique sur le bouton **Generate certificate** pour générer le nouveau certificat

Figure II.23 Création d'un 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**).

Enable FTP over SSL/TLS support (FTPS)

Private key file:  Browse...

Certificate file:  Browse...

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: 9899)

990

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

Generate certificate...

Autorise la connexion FTP sur TLS. C'est-à-dire que la connexion sera sécurisée.

Interdit la connexion sans TLS. C'est-à-dire qu'une connexion en clair ne sera pas autorisée

Il va crypter les données qui seront transférées en mode FTP sécurisé.

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

II.24 Paramètres pour la connexion FTP sur TLS

### III.10. Rubrique Autoban

Cette rubrique permet d'activer l'option pour bannir les 'utilisateurs illégitimes'. Pour ce faire, il suffit de cocher 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.

**Autoban** FileZilla Server

By default, FileZilla Server throttles incoming connects after some failed login attempts. This makes attempts to break into the server impractical. However, a legitimate user would still be able to logon at any time.

If you're feeling paranoid, you can enable an automatic ban upon multiple failed login attempts. Be advised that automatic banning can be abused for denial of service attacks, so use with care.

Enable automatic bans

Ban IP address after  failed attempts within one hour (min. 10).

Ban for  hours (1-999).

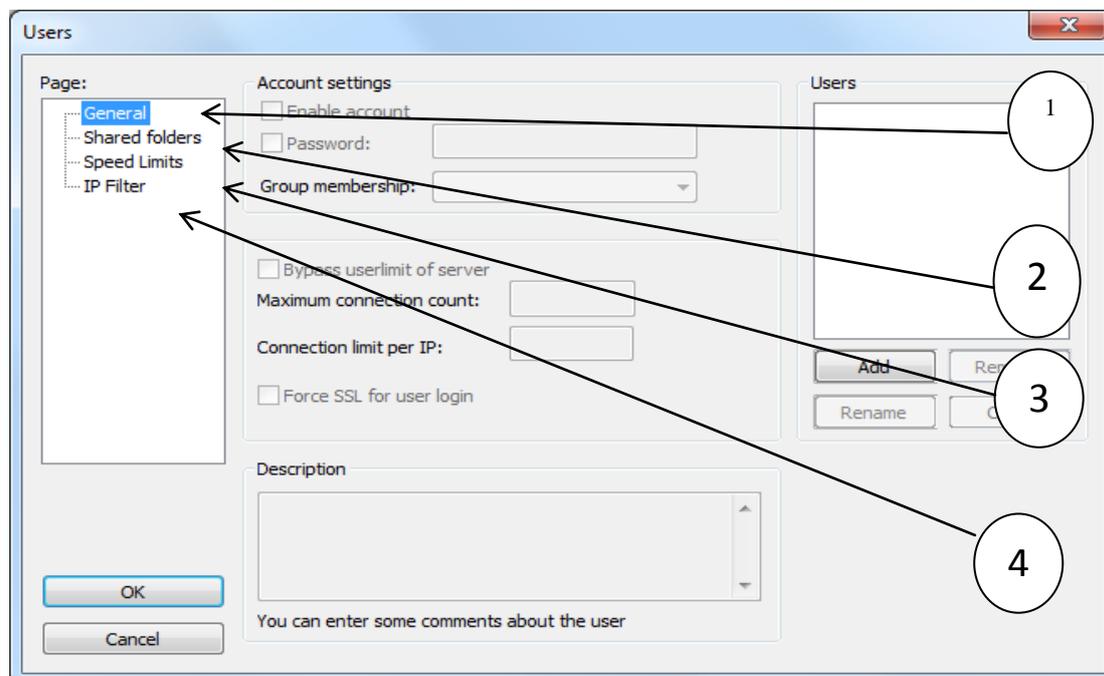
Par défaut, si l'utilisateur essaye de s'authentifier **10 fois en 1 heure** alors l'utilisateur sera **banni pour 1 heure**.

Figure II.25 paramètres pour bannir quelques utilisateurs

## IV. La Gestion des Utilisateurs

Pour gérer les utilisateurs il faut aller dans le menu **Edit** de l'interface de **configuration** puis **Users**.

Sur la gauche de la fenêtre qui apparaît (**Figure II.26**), les options du menu permettent de définir les paramètres liés au compte utilisateur que l'on souhaite créer.



**Figure II.26** Fenêtre principale de la gestion des Utilisateurs

### II.3.1. Rubrique Générale

Dans cet écran nous avons plusieurs paramètres pour définir le compte utilisateur lors de sa création, la première étant d'ajouter un compte utilisateur il suffit pour cela de cliquer sur le bouton **Add**.

Par exemple, si on souhaite ajouter le compte **TOTO**, il suffit de cliquer sur le bouton **ADD**, de rentrer le nom d'utilisateur puis de valider.

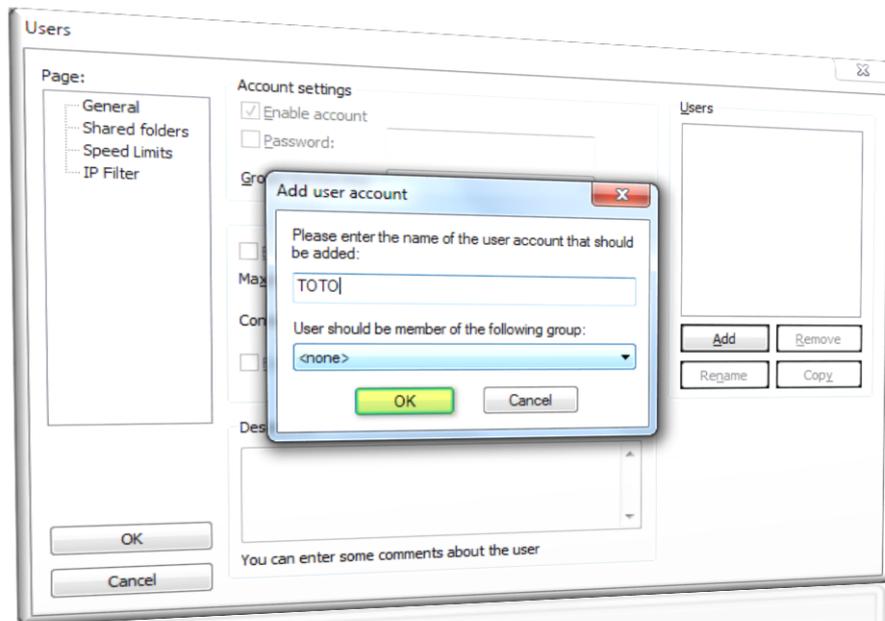


Figure II.27 L'ajout d'un compte utilisateur

Il est de plus nécessaire, de spécifier le mot de passe correspondant au nouveau utilisateur ainsi que les droits d'accès aux ressources du serveur.

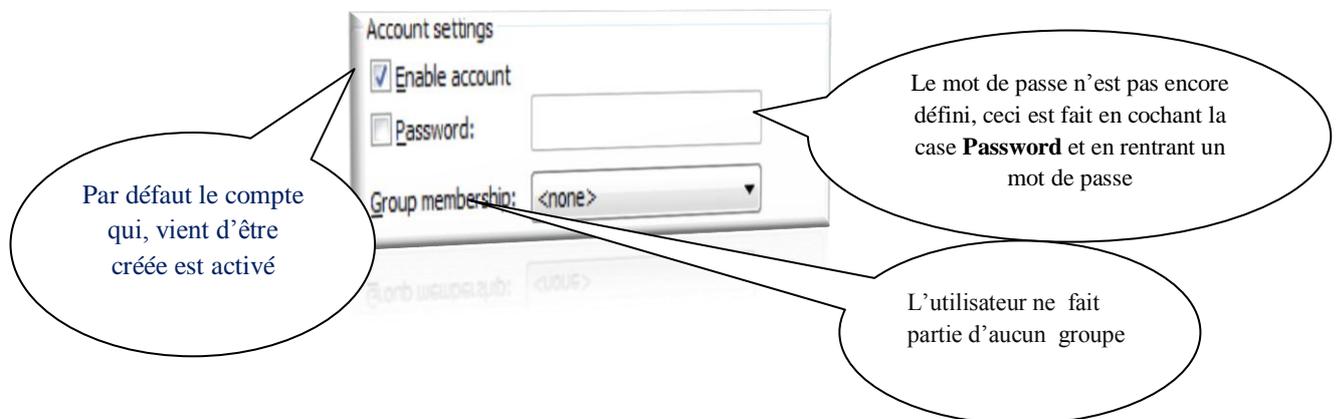
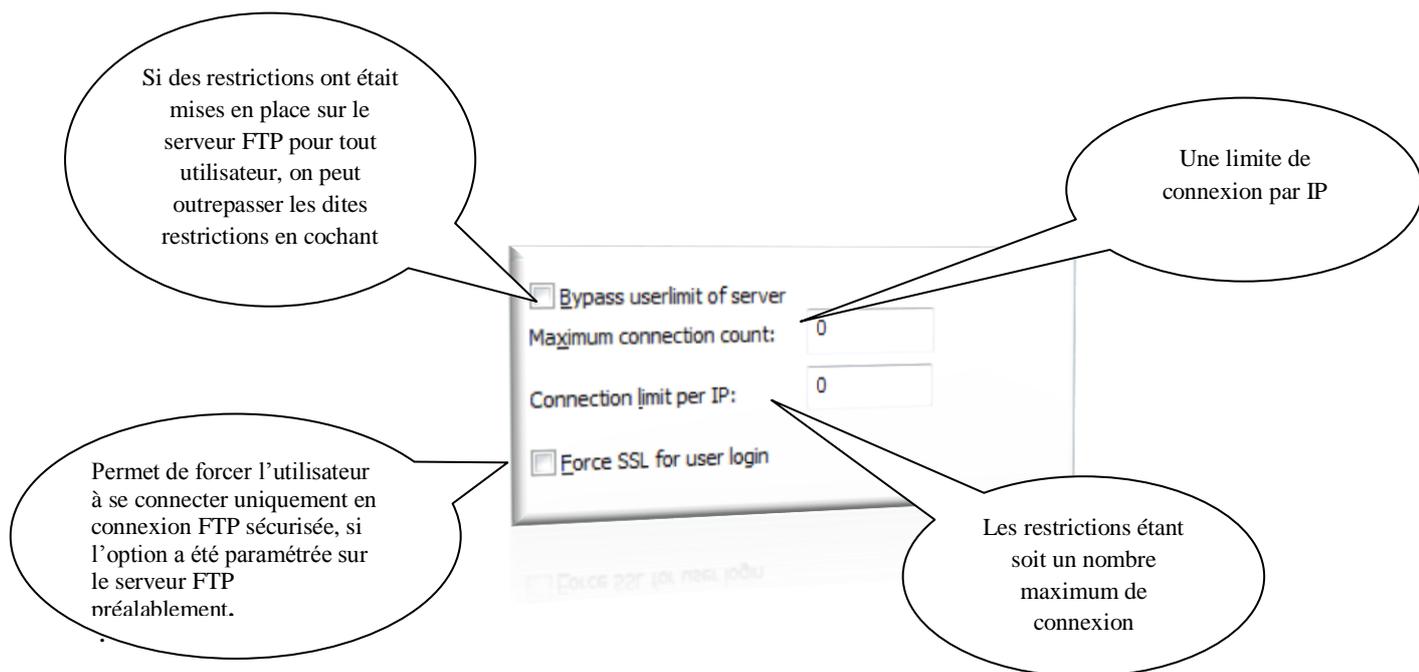
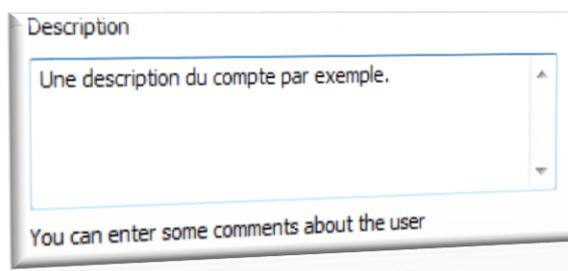


Figure II.28 Définition du mot de passe du nouvel utilisateur



**Figure II.29** Paramétrage des restrictions sur le serveur FTP

Pour chaque compte utilisateur, l'administrateur peut optionnellement ajouter une description du compte.



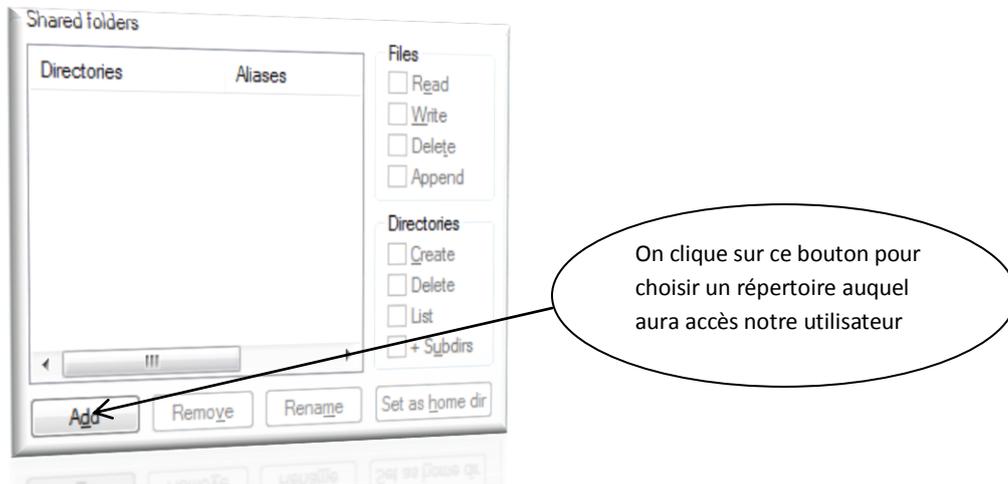
**Figure II.30** Description textuelle optionnelle du compte utilisateur

Notons que cette rubrique pour la configuration générale des comptes utilisateurs permet également de renommer, ajouter ou supprimer un compte existant.

A présent que notre compte utilisateur est créé, nous allons voir les autres options disponibles.

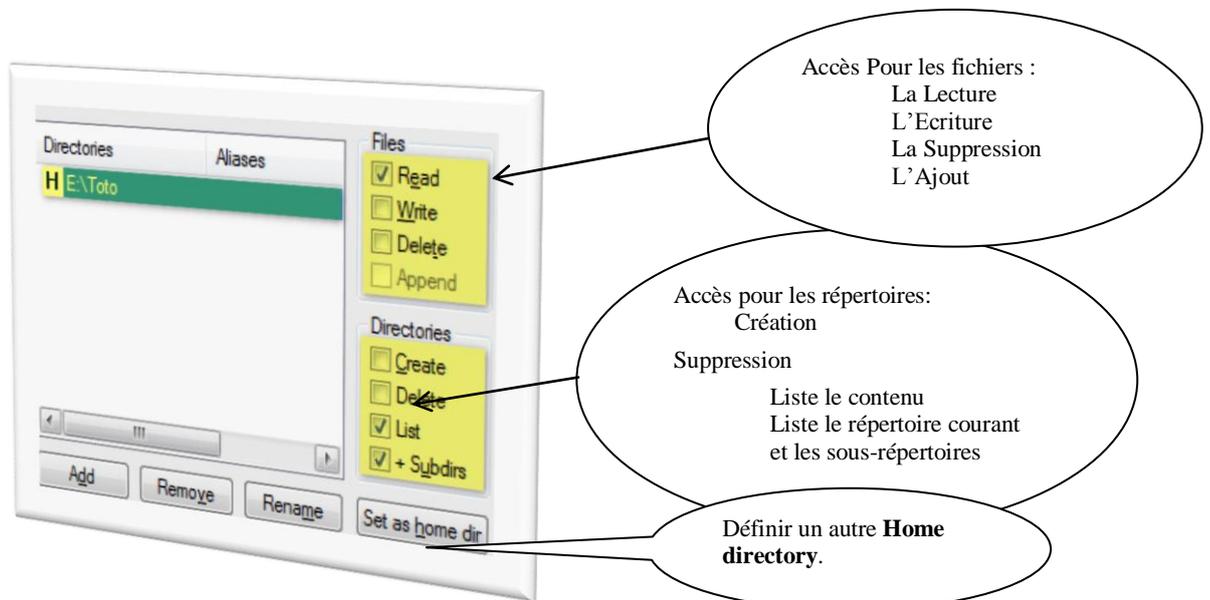
IV.2. Rubrique Shared folders

La fenêtre de cette rubrique (**figure II.31**) permet de définir les fichiers et répertoires auxquels aura accès chaque utilisateur ainsi que les droits d'accès.



**Figure II.31** paramétrage des répertoires et droits d'accès

Le premier répertoire ajouté est considéré, par défaut, comme le **Home directory** (petit H à la gauche du répertoire) de l'utilisateur. C'est le répertoire dans lequel l'utilisateur sera placé quand il se connectera au serveur FTP.



**Figure II.32** Les droits d'accès pour les fichiers et les répertoires

Dans le cas de l'utilisateur TOTO, créé précédemment, il aura accès à e:\toto en tant que répertoire racine, ainsi lorsqu'il fera une connexion sur le serveur FTP, il arrivera

directement dans ce répertoire. Au niveau des accès aux fichiers et aux répertoires, il pourra lister le contenu du répertoire ainsi que des éventuels sous répertoires. Pour les fichiers ils seront accessibles en lecture.

Ceci permet à l'utilisateur de créer (Create) de nouveaux répertoires au sein du répertoire racine et il pourra envoyer des fichiers Files – Append, modifier Files – Write et lire des fichiers Files – Read.

Si on souhaite autoriser l'utilisateur à envoyer des fichiers sur le serveur, l'administrateur lui donne accès à un autre répertoire, tel que E:\temp sur la figure suivante.

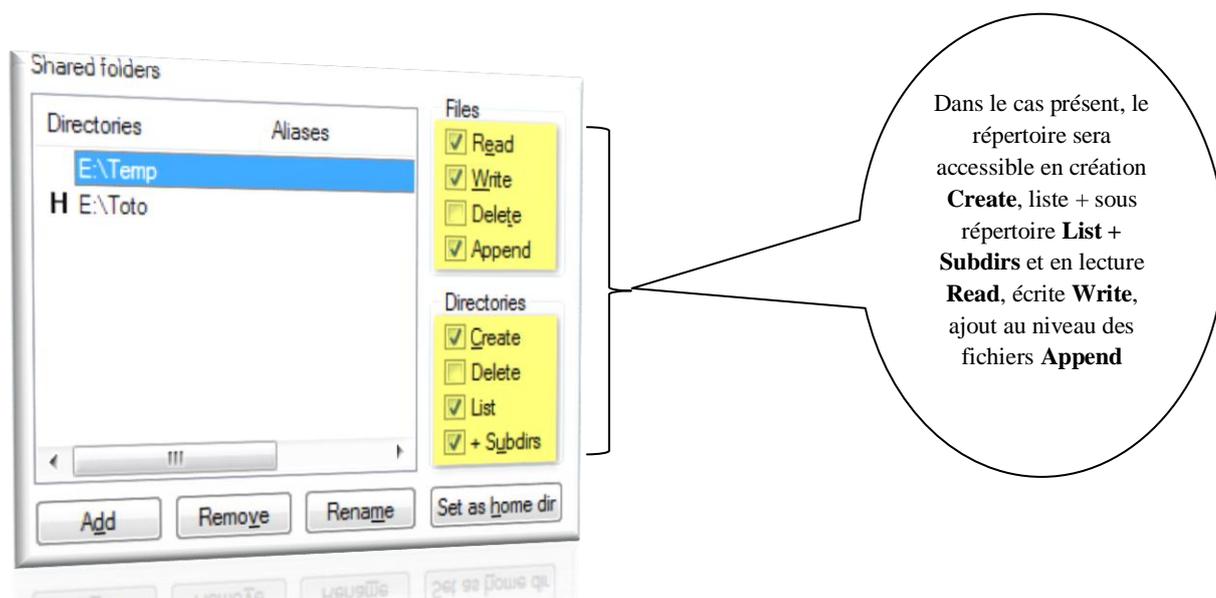


Figure II.33 Créer un nouveau répertoire

### IV.3. Rubrique des comptes utilisateur Speed Limits

Cet écran permet de définir les limites des débits en émission et en réception pour les comptes utilisateurs du serveur FTP à travers les commandes : Upload Speed Limit et Download Speed Limit.

Si l'administrateur souhaite outrepasser les règles de limitation mises en place au niveau des paramètres du serveur FTP, il coche l'option Bypass server download speed limit [4].

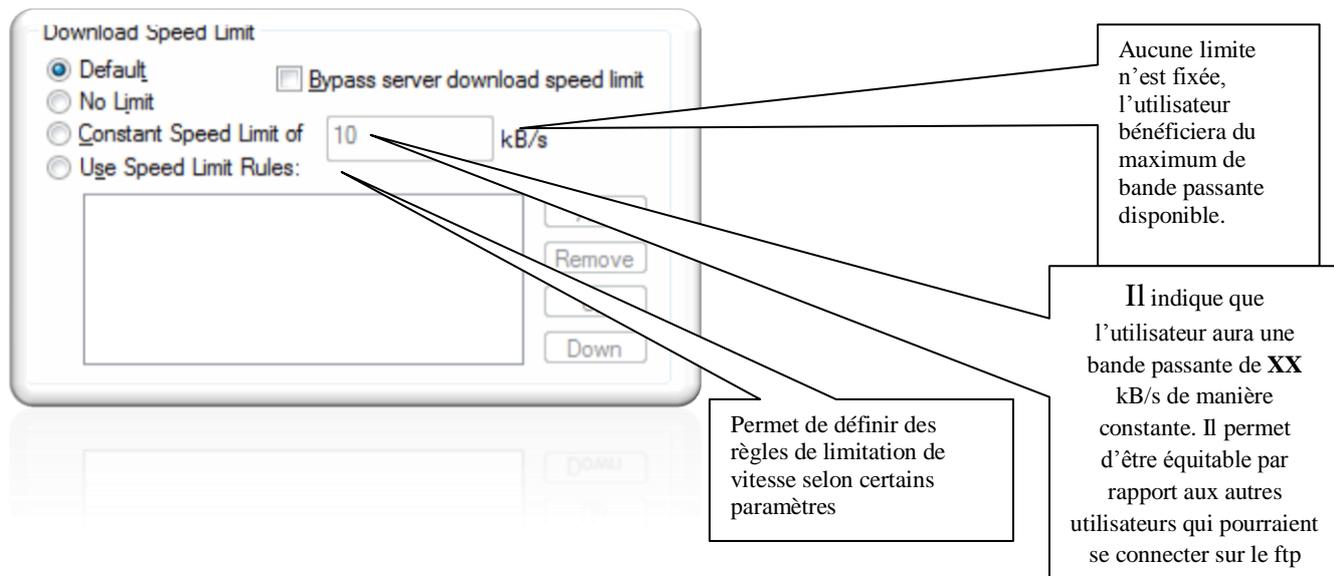


Figure II.34 Définir les limites de réception

➤ L'administrateur peut également définir une stratégie sous critères selon :

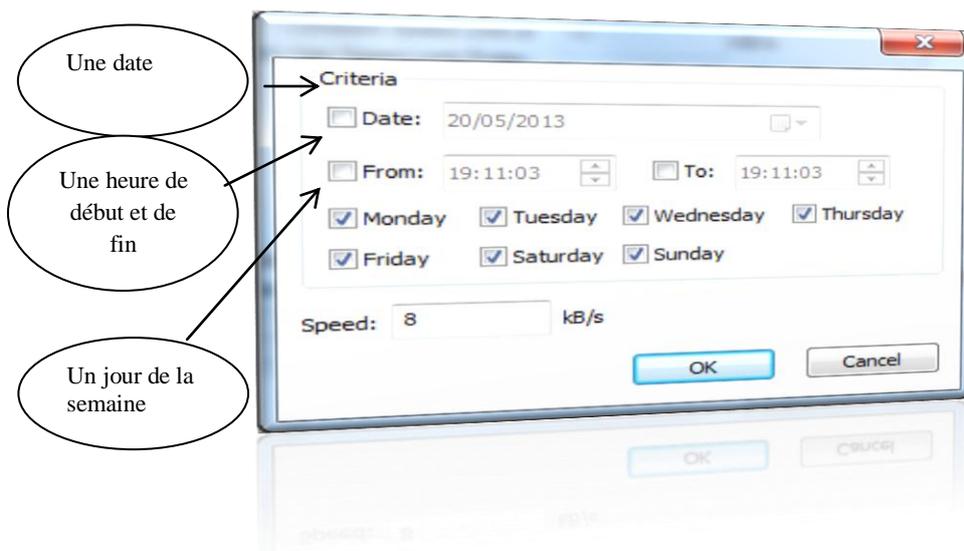


Figure II.35 Définir des critères

**Conclusion :**

Dans ce chapitre, nous avons présenté le serveur FTP **FileZilla Server** à travers l'exploration de quelques-unes de ces fenêtres et fonctionnalités.

*Chapitre III*  
*Implémentation*

**Introduction**

Après exploration du fonctionnement standard de FTP, nous pouvons alors passer à l'étape finale de ce mémoire à savoir l'implémentation. Elle a comme objectif d'aboutir à un produit final exploitable par les utilisateurs.

Nous essayerons dans cette phase de présenter une brève description de l'environnement de programmation Delphi7, mettre l'accent sur quelques caractéristiques de notre application client FTP.

Cette application a pour but de se connecter à n'importe quel serveur FTP pour assurer la gestion des fichiers telle que l'envoi, la suppression et le téléchargement.

**I. L'environnement de programmation DELPHI 7****I.1. Choix de l'outil**

Delphi est un environnement de développement de type RAD (*Rapid Application Development*) créée par Borland. Il est basé sur le langage Pascal. Il permet de réaliser rapidement et simplement des applications visuelles pour Windows [5].

La prise en main de Delphi est assez facile à adopter. Delphi permet de développer des programmes Win32 et .NET à partir du même langage [6].

D'une part le nombre de composants fournis avec Delphi, les possibilités en termes d'interface graphique sont très grandes. D'autres parts, les contrôles ActiveX, des composants actifs utilisables dans une application, permettent d'avoir accès à des fonctions avancées :

- ✓ Accès à des bases de données.
- ✓ Accès à des fonctionnalités réseau.
- ✓ Accès à des fonctions d'entrée-sortie [7].

De plus il présente les caractéristiques suivantes qui encouragent à l'utiliser :

- ✓ Delphi apporte une grande souplesse aux développeurs ;
- ✓ Delphi permet de réaliser des applications Microsoft Windows très efficace avec un minimum de codage manuel.
- ✓ Delphi offre une bibliothèque de composants visuels.
- ✓ Delphi possède un utilitaire (Report) qui permet de créer la forme désirée de la feuille d'impression.

- ✓ Delphi permet de développer des applications client/serveur.
- ✓ Delphi permet de simplifier le développement des applications Windows car il inclut plusieurs experts et d'autres outils spécialisés accélérant le développement.
- ✓ Delphi génère un vrai fichier exécutable (.EXE), aucun autre fichier n'est nécessaire pour son exécution donc on obtient une application plus facile à distribuer et à maintenir [8].

## I.2. Interface de développement de Delphi

L'interface de développement se divise en trois zones :

- ✓ Dans la partie supérieure de l'écran figure la fenêtre de programme, la barre d'outils ainsi que la palette des composants. Les fonctions concernant le projet sont situées à gauche dans la barre d'outils, à droite se trouve la palette des composants.
- ✓ Sous la fenêtre de programme à gauche, figure l'inspecteur d'objets. Il permet de définir l'aspect et le comportement de composants.
- ✓ A droite de l'inspecteur d'objets, Delphi affiche la première fenêtre de programme. A chaque fenêtre correspond un texte source qui détermine son fonctionnement.

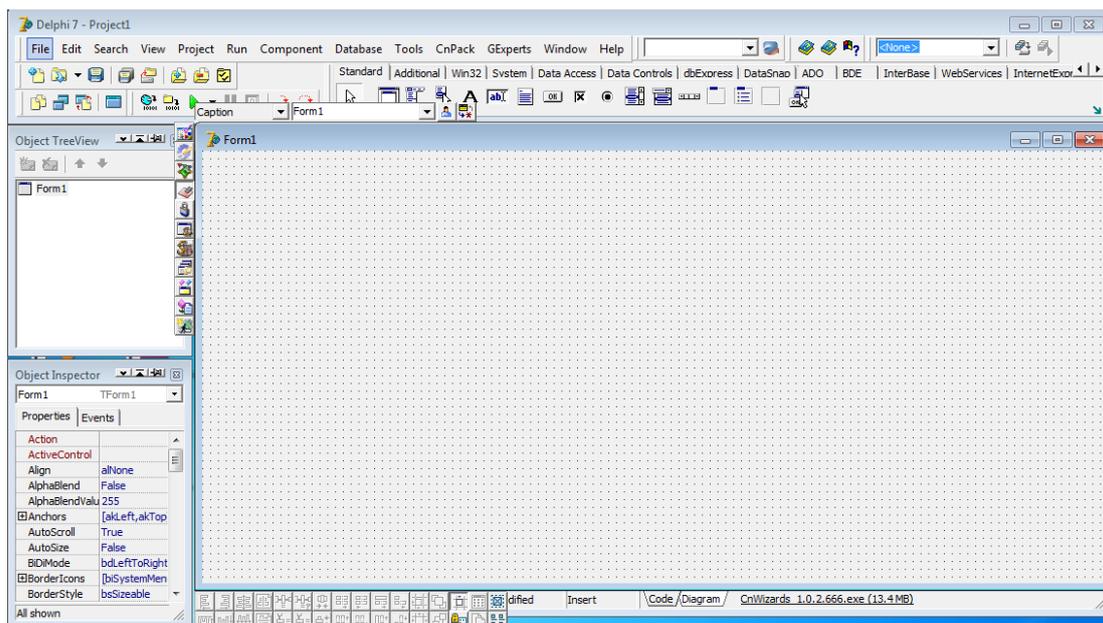


Figure III .1 Interface de Delphi

### **I.3. Pourquoi utiliser Delphi ?**

Delphi offre une façon de développer des applications sous un environnement très riche qui est Windows.

Delphi apporte une grande souplesse aux développeurs ;

Delphi permet de réaliser des applications Microsoft Windows très efficace avec un minimum de codage manuel.

Delphi nous offre un compilateur optimisé qui donnera une application rapide sans qu'il soit nécessaire de fournir plus d'efforts pour optimiser le programme.

Delphi offre une bibliothèque de composants visuels.

Delphi possède un utilitaire (Report) qui permet de créer la forme désirée de la feuille d'impression.

Delphi permet de développer des applications client/serveur.

Delphi permet de simplifier le développement des applications Windows car il inclut plusieurs experts et d'autres outils spécialisés accélérant le développement.

Delphi génère un vrai fichier exécutable (.EXE), aucun autre fichier n'est nécessaire pour son exécution donc on obtient une application plus facile à distribuer et à maintenir.

## **II .L'Interface graphique**

Dans cette partie nous allons présenter quelque fenêtre « notre application » :

### **II.1. Interface authentification**

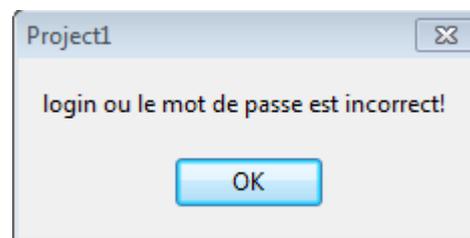
Au lancement de l'application, la fenêtre d'authentification s'affiche. L'utilisateur doit

Entrer son login et son mot de passe pour pouvoir accéder à son compte



**Figure III.2** Interface authentification.

Si login ou le mot de passe est erroné alors on affiche la boîte de dialogue suivant



**Figure III.3** Boîte de dialogue.

## II.2.La Fenêtre principal client FTP

Après la saisie correcte du login et le mot de passe, la fenêtre principale client FTP s'affiche.

Pour activer le transfert, il est nécessaire de saisir, dans la partie demande la connexion, les informations suivantes : adresse du serveur (numérique ou le nom du serveur), nom d'utilisateur, mot de passe, le port utilisé et le time out. L'utilisateur doit valider par le bouton connexion pour établir une connexion.

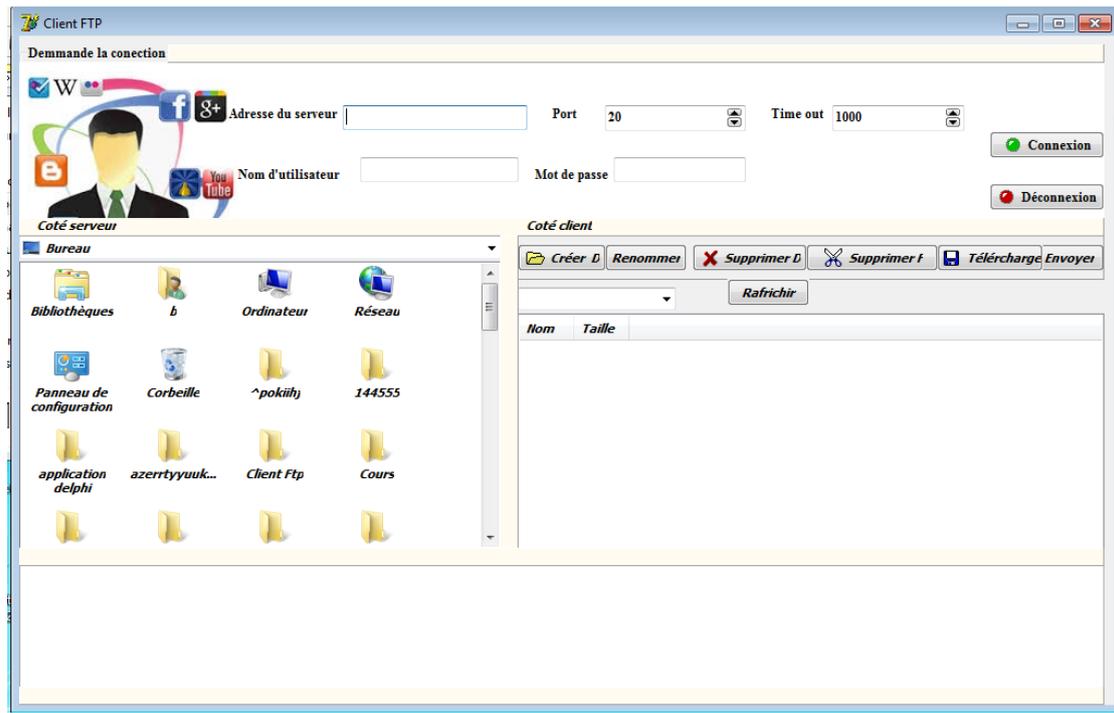


Figure III.4 La Fenêtre principal client FTP

### II.3. Fonctionnalités du client FTP

Quand on clique sur le bouton créer dossier une fenêtre s'affiche pour saisir le nom du dossier à créer.

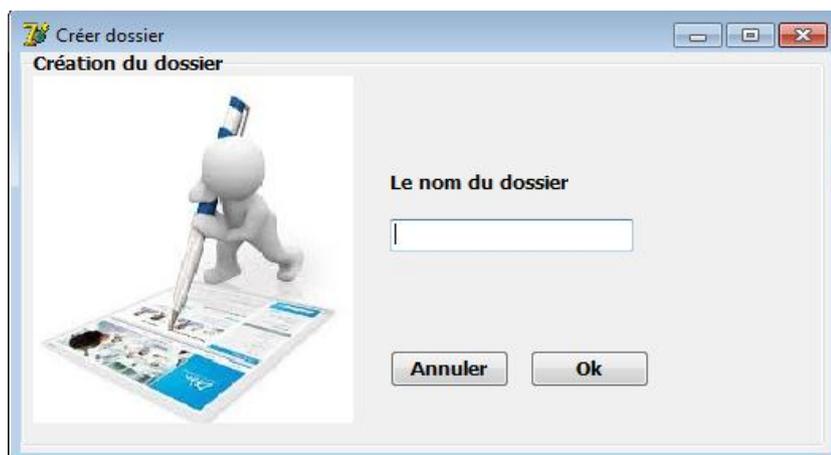


Figure III.5 L'option créée dossier

Dans la liste des fichiers et les dossiers du côté client on sélectionne le fichier ou le dossier qu'il doit renommer, puis une fenêtre s'affiche pour saisir le nouveau nom

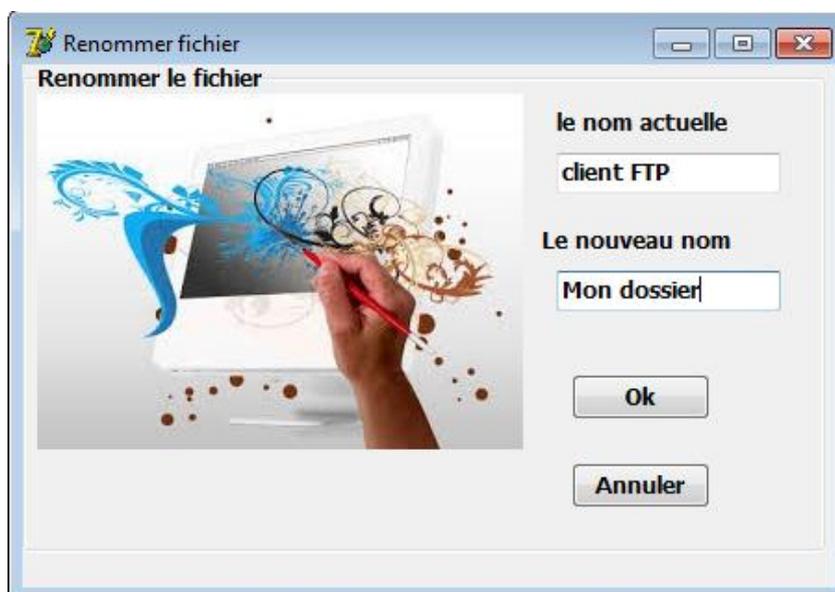


Figure III.6 L'option renommée dossier ou fichier

De même, on sélectionne le dossier que nous avons choisi pour la suppression et on clique sur le bouton supprimer D. un message de confirmation de l'action provient en réponse à partir du serveur.

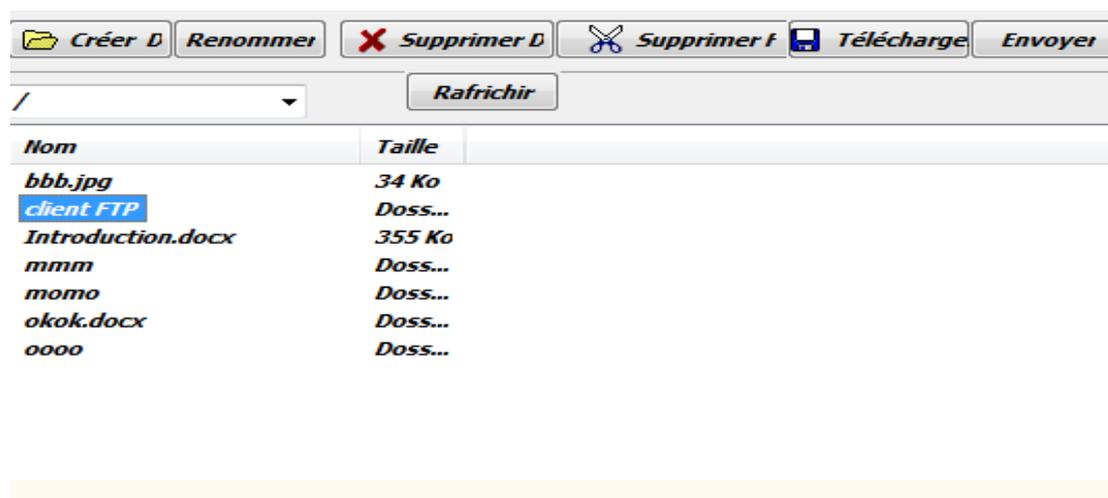


Figure III.7 L'option supprimée dossier

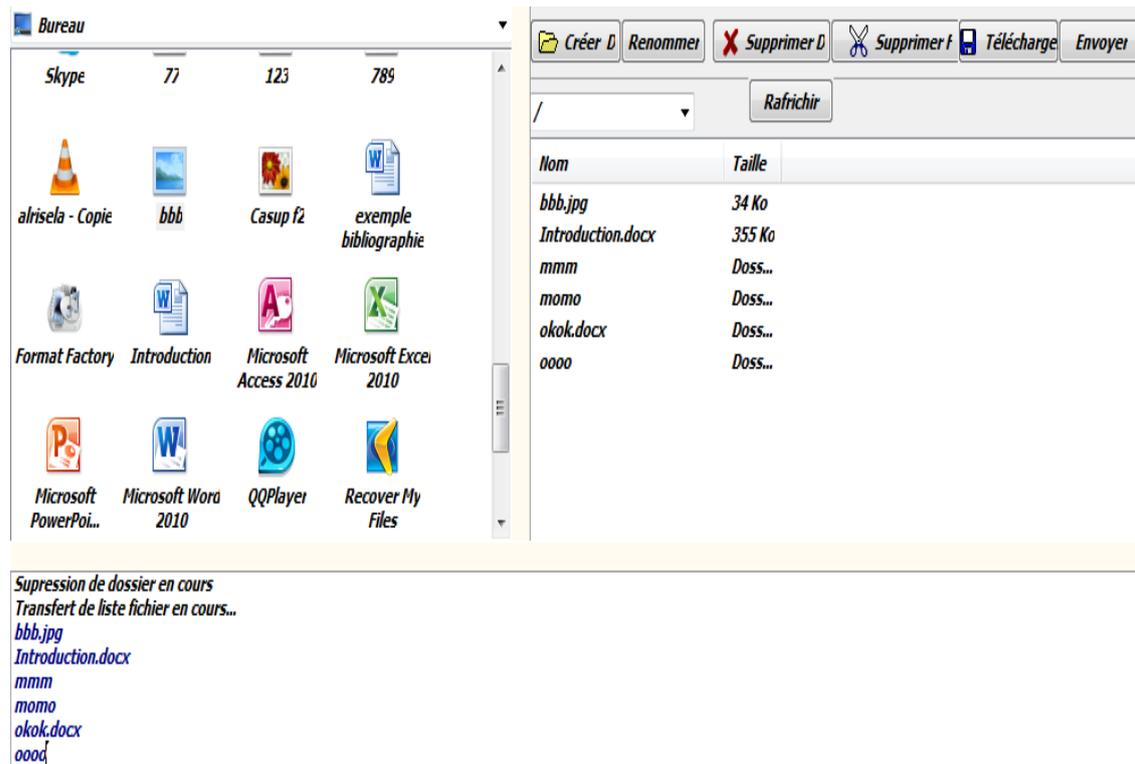


Figure III.8 Message confirmation de la suppression

Il est également possible de procéder à une suppression de fichier à partir du client FTP. Le serveur rend une réponse textuelle qui apparait sur la fenêtre du client.

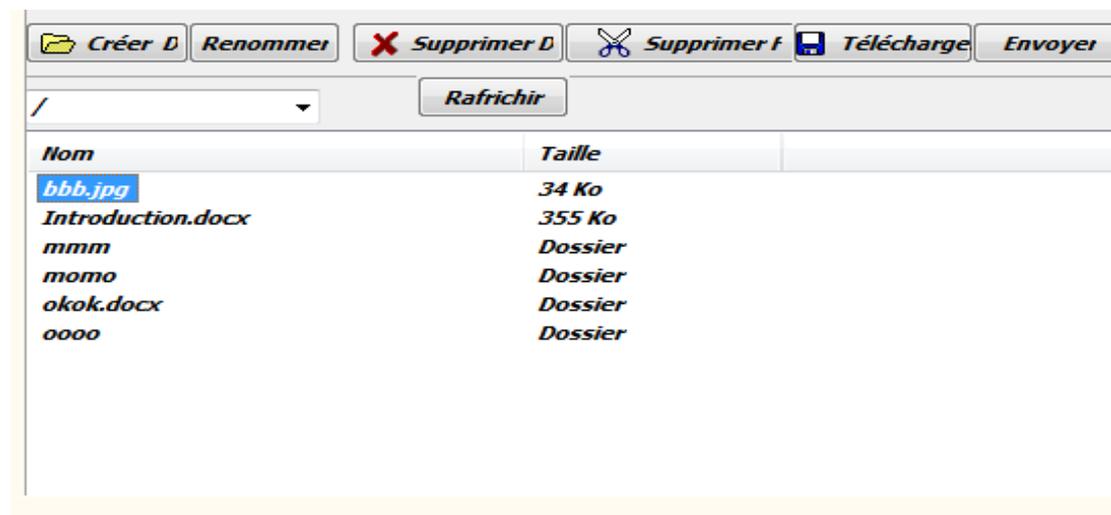


Figure III.9 L'option supprimée fichier

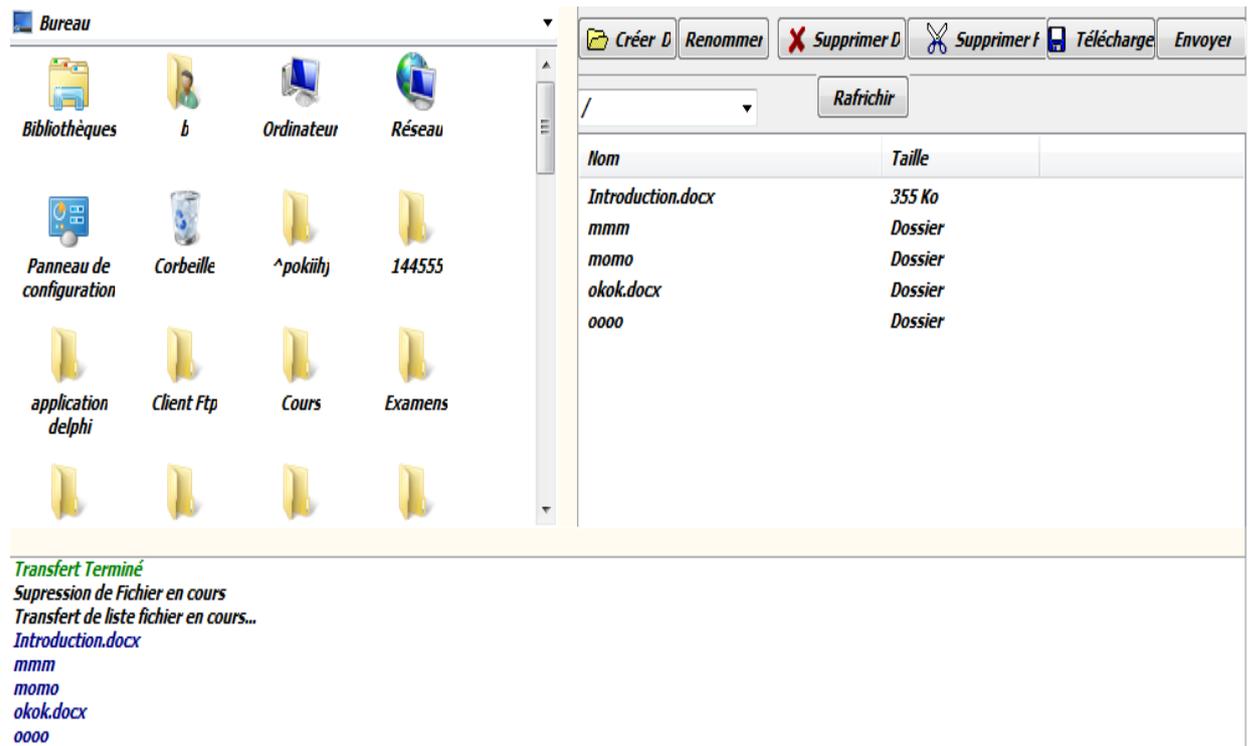


Figure III.10 Confirmation de la suppression du fichier

Dans la partie Coté serveur on choisit le fichier de n'importe quel type (document Word, document PDF, photo, vidéo...) qu'il on veut le télécharger et on clique sur le bouton Télécharger.

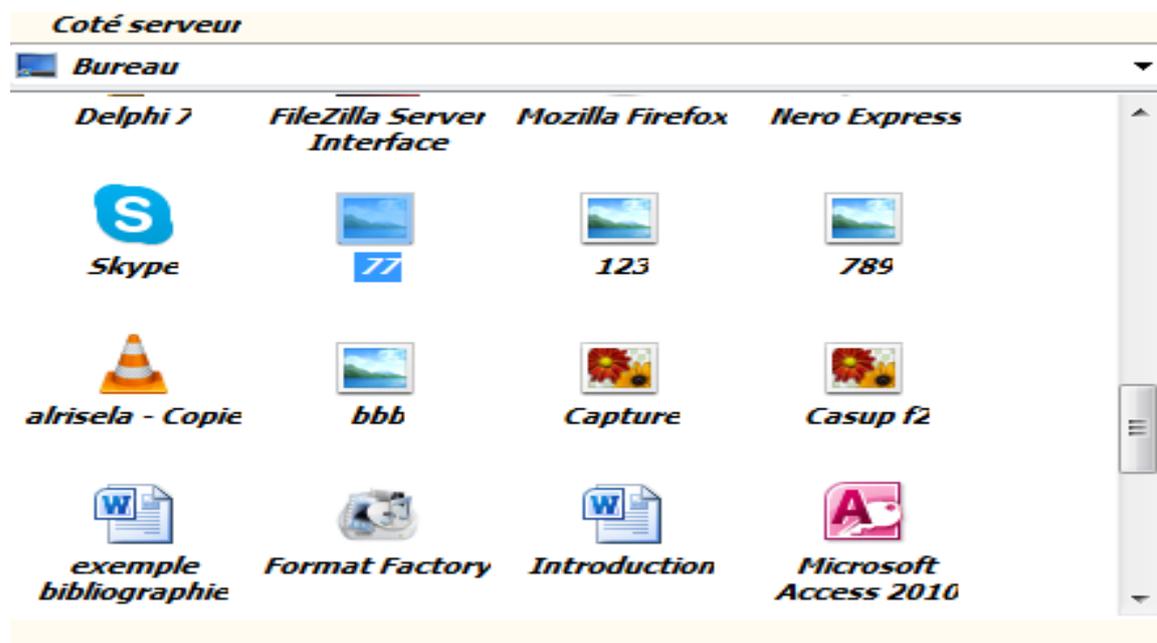


Figure III.11 L'option Télécharger

Dans la partie coté client, on observe la réception du fichier (le fichier est inclus dans le répertoire du client). De plus le serveur envoie un message visible au niveau de l'interface client qui confirme le téléchargement du fichier.

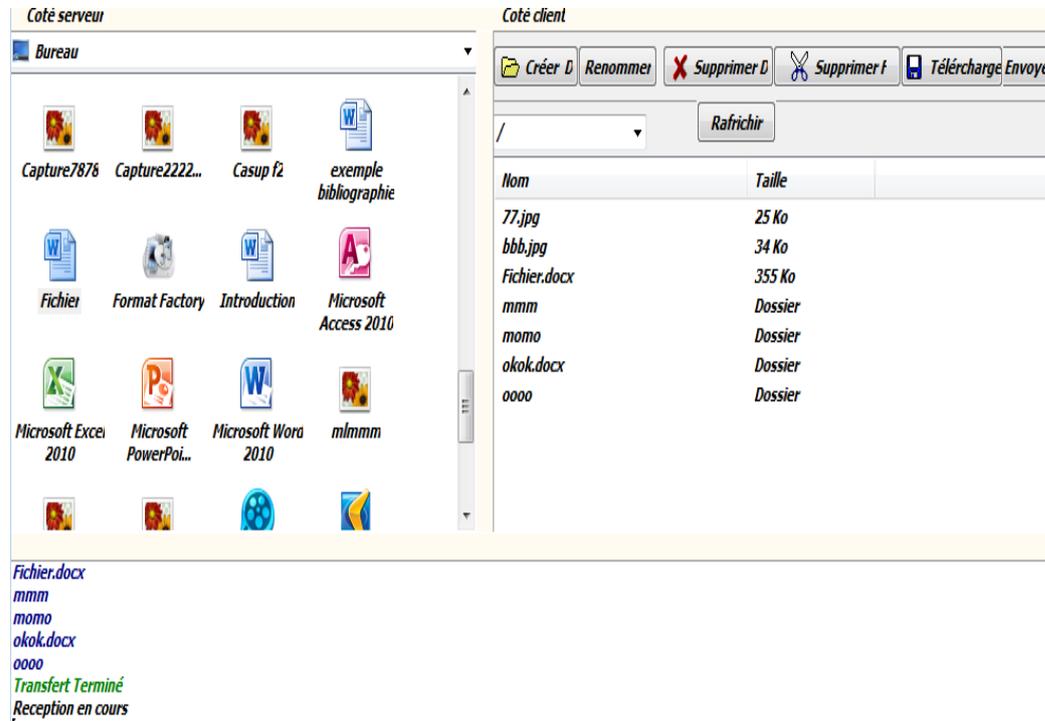


Figure III.12 Confirmation du téléchargement du fichier.

Dans la partie coté client on choisi le fichier ou le dossier que l'on veut envoyer. Il se doit également de préciser le répertoire de réception. Après confirmation, le serveur répond par un message



Figure III.13 L'option envoyée

Dans la partie coté serveur on observe la réception que le fichier(le fichier est inclus dans le répertoire du client).

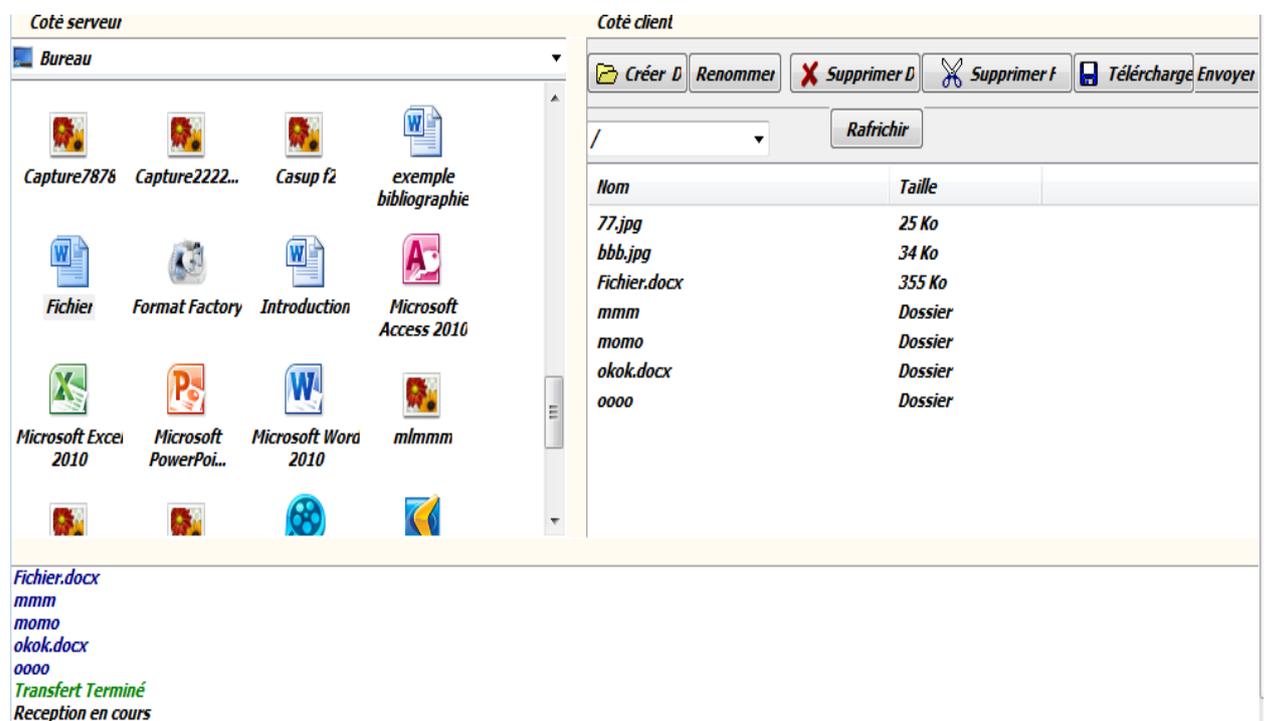


Figure III.14 Confirmation d'envoi du fichier.

*Conclusion*

Dans ce chapitre, nous avons décrit brièvement la phase d'implémentation tout en donnant une vue globale sur les outils utilisés ainsi que l'application que nous avons réalisée.

Nous avons présenté l'interface de l'application avec toutes les options qu'elle contient afin de faciliter son utilisation.

# *Conclusion général*

Notre projet s'articule autour de la conception et la réalisation d'un client ftp.

Pour mieux comprendre son fonctionnement, on a étudié le protocole FTP avant sa réalisation. Nous avons porté notre choix respectivement sur l'environnement Delphi pour implémentation et le serveur FileZilla Server pour l'administration du serveur.

La période passée au développement de notre application, nous a été d'un apport considérable. En effet, c'est une expérience qui nous a permis d'enrichir nos connaissances dans un nouveau domaine lié aux réseaux informatiques.

Nous avons produit un client FTP qui permet une communication avec n'importe quel serveur FTP pour assurer le transfert et la gestion des fichiers. Cette application peut servir de base pour de nouvelles fonctionnalités relatives aux nouveaux RFC de FTP, pour implémenter la sécurité ou autres extensions.

# *Bibliographie*

- [1] Le protocole FTP, <http://fr.wikipedia.org>.
- [2] Le Modèle Ftp, [http:// www.comment sa marche.com](http://www.comment sa marche.com).
- [3] Mémoire fin d'étude, « Réalisation et configuration d'un système FTP », Université Badji Mokhtar de Annaba, Tazerouti Oussama-Zerizer Anis '2007'.
- [4] Le serveur FileZilla Server, <http:// www.comment sa marche.com>.
- [5] Programmation sous Delphi, [http : // www.www.comment sa marche.com](http://www.www.comment sa marche.com)'.
- [6] Le langage programmation Delphi, <http:// www.www.comment sa marche.com>'.
- [7] Mémoire fin d'étude, « conception et implémentation d'un outil de délibération », Centre Universitaire de Mila, Benmakhlouf Amal - Laieb karima, 2011-2012.
- [8] Conception et réalisation d'un système pour la gestion du matériel électrique et gazier au sein de la CAMEG de Mila (filiale du groupe Sonelgaz), Mehamda Rima - Maamar Selma, 2010-2011.