
Rapport de stage

Développement d'une application pour
la gestion des Mises à jour

**Société I.N.M.C.
Idéation Informatique**

**Responsable en entreprise : M David Comptdaer
Responsable universitaire : M Olivier Flauzac**

Sommaire

Remerciements -----	4
Introduction -----	5
Présentation de l'entreprise -----	6
Composition de l'entreprise -----	7
Le Sujet -----	8
Les raisons du sujet -----	8
Présentation du sujet -----	8
Idée principale -----	8
Présentation de WINDEV -----	9
Le Produit WINDEV -----	9
<i>Les Caractéristiques importantes de WINDEV</i> -----	10
<i>Les Concepts</i> -----	11
<i>Les Caractéristiques du langage W-Langage</i> -----	11
<i>Optimisation pour les bases de données</i> -----	11
<i>Les langages extérieurs à Windev</i> -----	12
Présentation de l'environnement de programmation -----	13
Analyse et prise de choix -----	14
Les communications externes -----	14
Schéma final -----	15
Le Cryptage -----	15
La Base de Données -----	16
Type de base de données -----	16
La base de données existante -----	16
<i>Les tables réutilisées</i> -----	17
<i>Représentation des tables</i> -----	18
Ajout et Modification de l'analyse -----	18
<i>Les tables pour la Traçabilité</i> -----	18
<i>Les tables pour le Référencement des Mises à Jour</i> -----	19
<i>Interconnexion des tables</i> -----	20
Schéma de l'analyse -----	21
Utilisation de la base de données -----	21
<i>DDMDL</i> -----	22
<i>Serveur de sockets</i> -----	22
<i>Service LiveUpdate</i> -----	22
<i>Service Mises à jour</i> -----	22

Développement et communication en interne	23
Communication DDMDL-Serveur de Sockets	23
<i>Communication par sockets</i>	23
<i>Base de données partagée</i>	23
Communication Serveur de sockets – Serveur FTP Windev	23
Communication DDMDL – Serveur FTP Local Windev	24
Schéma de communication interne	24
Schéma interne	24
Communication globale entre interne et externe	25
Utilité de chaque élément	25
Schéma global des communications	26
Récapitulatif de l'utilité des entités	27
<i>Serveur de sockets</i>	27
<i>DDMDL</i>	27
<i>Module Externe</i>	27
<i>Module Client</i>	27
Les Protocoles	28
Protocole LUD	28
Début protocole	29
Protocole MAJ	30
Début protocole	31
Gestion des communications	33
Interface Homme-machine	34
Le Serveur de Sockets	34
Interface du module MAJ	35
Phase 1 : Mise en place de la MAJ	35
Phase 2 : Spécification de MAJ	36
Interface du module LUD	37
Phase 1 : Identification et sélection du produit à mettre à jour	37
Phase 2 : Sélection de la MAJ à télécharger et à installer	38
Conclusion	39
Glossaire	40
Annexe 1 : Base de données globale	41
Annexe 2 : Code de la fonction principale du serveur	43
Annexe 3 : Interface du DDMDL	48

Remerciements

Je tiens à remercier en premier lieu, mon tuteur de stage, Monsieur David Comptdaer, pour m'avoir si bien encadré durant toute la durée de mon stage, ainsi que son père, Monsieur Michel Comptdaer, fondateur de l'entreprise qui a bien voulu m'accueillir. Je remercie aussi Monsieur Julien MIART pour m'avoir beaucoup aidé à découvrir ce nouveau langage de programmation.

Introduction

L'entreprise INMC idéation informatique est une petite société qui recherchait une solution pour mettre à jour les logiciels de leurs clients à distance, à travers les réseaux locaux mais également internet. Pour cela, ils ont choisi de me prendre en stage pour développer une solution simple et efficace en quatre mois. Pour la réalisation de ce projet, il fallait utiliser une plate-forme de développement qui m'était inconnue : Windev. Le travail fut réalisé en plusieurs étapes afin d'obtenir un résultat accessible à tous les clients.

Présentation de l'entreprise

La société dans laquelle j'ai réalisé mon stage de DESS est la société :

INMC - Idéation Informatique.

Cette société existe depuis 1989. Elle base son activité principalement sur le développement d'applications spécifiques (Progiciels), en grandes parties spécifiques à des communes. Leur travail est donc de développer, installer et maintenir une large gamme de progiciels.

INMC – Idéation Informatique offre aussi les services suivants à ses clients :

- Analyse
- Audit
- Développement
- Migration
- Installation
- Formation
- Traitement de données
- Maintenance et assistance

La force de cette société réside, grâce à son équipe technique en adéquation avec la force de vente, dans le fait d'adapter ses logiciels, basés sur un noyau commun, vers des besoins parfois très spécifiques de ses clients.

Fort de son expérience et disposant de collaborateurs et partenaires compétents, la société **INMC - Idéation informatique** permet de répondre aux besoins dans une multitude de domaines tels que :

- Les bases de données
- Le développement Réseau
- Client / serveur
- Le développement Internet - Intranet
- Les systèmes d'informations géographiques

Cette société développe de multiples progiciels tels que « Population » : un logiciel de gestion de la population d'une commune ou encore des logiciels de gestion de centre de loisirs, des logiciels de gestion de courriers,

- Population
- Cimetières (*)
- Locations de salles
- Courriers
- Protocole
- Prêts de matériels
- Restaurant scolaire
- Suivi des Interventions
- Patrimoine (*)
- Vaccinations
- Suivi scolaire
- Inventaire
- Centre de Loisirs

(*) Option cartographie

Composition de l'entreprise

L'entreprise dans laquelle j'ai effectué mon stage est composée de 2 parties très complémentaires qui sont :

- Une partie commerciale qui s'occupe de chercher et de répondre aux appels d'offres des diverses communes, entreprises ou encore préfectures qui recherchent un logiciel spécifique que l'entreprise a déjà conçu ou peut développer.
- Une partie développement qui s'occupe de concevoir des nouvelles applications ou encore de modifier (ou migrer) des applications déjà existantes afin de répondre au plus vite et au mieux aux différentes demandes.

La complémentarité de ces deux services est basée sur l'apport d'informations en provenance des clients. Cela veut dire que le service commercial peut, à tout moment, conseiller, voir orienter les développements en cours de réalisation.

Le SUJET

Les raisons du sujet

Dans l'entreprise INMC Idéation informatique, il y a de multiples logiciels qui sont créés et constamment mis à jour (une trentaine). Actuellement, chaque client ayant pris dans son contrat la maintenance de logiciel, a le droit à toutes les mises à jour qui sont recrées tout au long de l'année. Pour le moment, les mises à jour sont installées par un employé qui se déplace, de client en client, pour mettre à jour leurs logiciels. L'entreprise s'est rendue compte que le coup de revient d'un tel acte est assez important. Ce qui deviendra par conséquent de plus en plus difficile à réaliser vu le nombre de clients et le nombre de produits qui croient chaque année. De plus, il faut également prendre en compte la distance avec les clients qui s'étend dans toute la France et de plus en plus dans le sud.

Présentation du sujet

En vue des problèmes posés, il m'a été demandé de développer un système de mise à jour à distance des différents clients. Différentes architectures possibles m'ont été présentées. Il a finalement été décidé de développer cette application dans un langage que je ne connaissais pas : Windev7.5.

Ce développement devait répondre à certaines contraintes externes et internes afin d'avoir, au final, un ensemble cohérent.

- Contraintes internes :
 - Développement en Windev 7.5
 - Gestion simple des mises en place des mises à jour
 - Interopérabilité entre leur logiciel de gestion de clientèle et cette application
 - Gestion de toutes les applications en interne
 - Possibilité d'extension
 - Connexion des clients directement à l'entreprise par sa connexion ADSL qui change d'IP une fois par jour
- Contraintes externes :
 - Connexion via le réseau Internet
 - Utilisation de communications pouvant passer les différents FireWall, Proxy ou voire routeur de chaque client
 - Développement d'une application utilisable par des personnes non compétentes en informatique

Idée principale

Le client qui possède un logiciel et qui a le droit de faire une mise à jour, doit pouvoir, grâce à une connexion Internet, mettre à jour celui-ci. Pour cela, il doit utiliser un petit module qui lui permet de se connecter à l'entreprise et de s'identifier grâce à son Login et Password. Ensuite, l'entreprise lui retourne les informations demandées qui doivent s'installer implicitement sur le poste ou serveur du client.

Présentation de WINDEV



Windev est un atelier de génie logiciel qui permet la création et le développement rapide d'applications de gestion.

Cet outil est développé par la compagnie PC-SOFT, basée en France, ce qui fait aussi de WINDEV l'outil de développement numéro 1 en France et dans presque tous les pays francophones.

Plusieurs versions de ce logiciel ont été créées, notamment la version 4.11 qui a été la clé du succès de WINDEV. Ensuite est apparue la version 5 incluant des outils pour le développement Internet.

Les dernières versions en date sont la version 7(7.5) et aussi Webdev qui est un outil complet de développement d'applications Internet.

Windev suit l'évolution de Microsoft pour adapter ses versions aux différents Windows existants sur le marché, donc le développement d'applications est possible de Windows 3.1 à Windows XP en passant par Windows 2000(NT).

Le Produit WINDEV

Windev est un produit fonctionnant sous Windows. Il permet de créer très simplement des applications windows. En effet, Windev est un outil à multiples facettes qui permet de gérer aussi bien des fenêtres, des menus, des fichiers que des états imprimés et des requêtes. C'est aussi un éditeur de codes sources, débogueur et compilateur.

Les objets d'un projet windev sont facilement créés avec des éditeurs adaptés. Ces éditeurs existent en français et en anglais (If you like).

Tout développement est réalisé dans le langage spécifique mais pas unique de Windev qui est le **W-langage** (existant en français et en anglais).

La programmation en **W-langage** peut être procédurale ou orientée objet selon les besoins des développeurs (W-Langage sera présenté dans les pages suivantes).

Les Caractéristiques importantes de WINDEV

En dehors de sa simplicité d'utilisation, il apporte de nombreux avantages :

- ***Description des objets à l'aide des éditeurs conviviaux***

Afin de créer les différents objets d'un projet Windev, plusieurs éditeurs sont directement accessibles depuis l'éditeur principal (éditeur de fenêtres, éditeur de menus, éditeur d'analyse, éditeur d'états imprimés,...). Aussi, un éditeur de code source permet de saisir les traitements associés à chacun des objets.

- ***Choix du langage de programmation par le développeur***

Le développeur, en créant son projet Windev, peut faire le choix de son langage de programmation. La programmation en W-langage peut-être « *traditionnelle* » ou « *orientée objet* ».

- ***Développement de programmes pouvant fonctionner en monoposte ou en réseau***

Windev permet de créer des applications fonctionnant en réseau ou en monoposte. Les fonctions du W-langage gèrent les accès concurrentiels aux fichiers. Au choix du développeur, elles permettent de bloquer les fichiers ou les enregistrements en lecture ou en lecture et en écriture.

- ***Facilité d'installation d'une application chez un utilisateur***

Windev facilite l'installation d'un programme chez un utilisateur. Avec son éditeur, le développeur a la possibilité de créer un jeu de disquettes d'installation (ou cd) contenant tous les modules nécessaires au fonctionnement du programme et le programme d'installation.

- ***Diffusion des programmes développés avec Windev et des librairies à liens dynamiques de Windev (DLL).***

Le programme exécutable créé s'exécute grâce aux librairies à liens dynamiques (*DLL*) de Windev. Ils peuvent s'exécuter sur plusieurs postes.

- ***Développement d'une application en plusieurs langues***

Une même application peut être développée dans plusieurs langues. Le développeur choisit les langues par rapport aux messages et aux objets, une fonction Windev se charge alors de passer d'une langue à une autre et un seul exécutable est créé.

- ***Développement en équipe***

Windev permet de développer une application en équipe grâce à son module de gestion du groupware. Le groupware Windev permet un partage en réseau, mais également en mono

poste. Chacun des développeurs, partageant le projet, se voit attribués différents droits sur les différents objets du projet partagé.

Les Concepts

□ Application Windev

Une application Windev est un ensemble logique d'objets qui utilisent la même base de données.

□ Analyse Windev

Une analyse Windev regroupe les descriptions des fichiers de la base de données. L'analyse, les projets et les fichiers (ou table) sont décrits directement avec les éditeurs de Windev.

□ Projet Windev

Un projet est un ensemble logique d'objets et de traitements dont le but est de réaliser un objectif donné. Ce projet gère les relations entre les objets (c'est-à-dire d'objets à objets) et les relations entre les objets et les fichiers de données aussi appelés table.

Les Caractéristiques du langage W-Langage

Windev est livré en standard avec un langage évolué de quatrième génération (LAG): le W-Langage. Le W-Langage est un langage de programmation très intuitif et puissant. Il reprend les avantages du C et du Pascal et comporte des avantages spécifiques. Le français permet une programmation intuitive et une maintenance facile. Le développeur n'a pas à se poser de questions quant à la signification réelle d'un ordre particulier : TableAjoute(), NumériqueVersChaine() ou encore FTPConnecte() sont des ordres faciles à comprendre sans aucun effort particulier, donc faciles à maintenir. Ce langage en français permet par conséquent une meilleure productivité et une meilleure qualité du code.

Optimisation pour les bases de données

Le W-Langage est optimisé pour manipuler les données de fichiers. Une ligne de W-Langage correspond souvent à des dizaines de lignes de langage classique. Ce qui explique sa vitesse et sa fiabilité pour gérer les données. Windev est principalement orienté vers les bases de données. Ces bases de données utilisent des fichiers qui représentent les différentes tables de notre analyse (Model Conceptuel de données, « MCD »). Dans Windev le système de base de données se nomme Hyper File.

Le plus de Windev est son ouverture vers l'interopérabilité des systèmes puisque Windev intègre l'AS400 qui permet un accès natif au base de données.

Windev AS/400 permet l'accès natif (sans ODBC, sans OLE DB, sans ActiveX) en temps réel aux bases de données AS/400 ainsi qu'aux programmes et aux commandes de l'AS/400.

La programmation est la programmation habituelle de WinDev. Aucune programmation AS/400 (RPG, Cobol,...) n'est requise pour l'utilisation de WinDev AS/400. Les programmes, déjà écrits avec WinDev, fonctionnent automatiquement sur les données AS/400, si vous le désirez. La portabilité est immédiate.

A partir d'une même application développée avec WinDev AS/400, il est possible d'accéder soit aux fichiers natifs AS/400, soit aux bases de données au format WinDev (Hyper File), soit aux deux, et même à d'autres bases de données (Oracle,...) de façon totalement transparente.

Vous pourrez mixer des données de bases différentes.

Les stations clientes sous Windows bénéficient avec Windev de sa puissance de développement et de la vitesse de l'accès natif aux données AS/400.

Les langages extérieurs à Windev

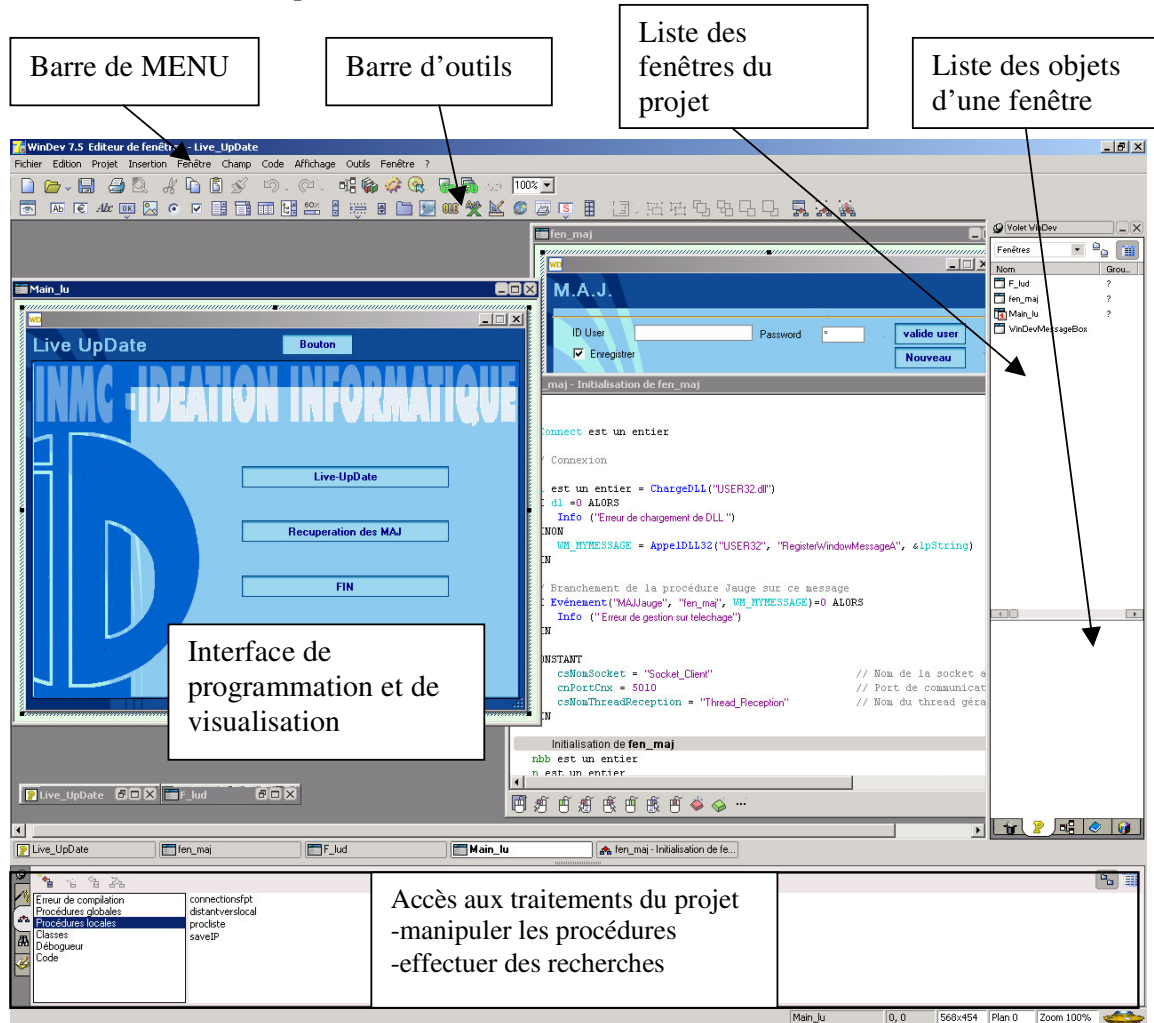
Le W-langage est un langage de programmation déjà très puissant par lui-même. Mais PC-SOFT, le créateur de Windev, a préféré, en gage d'ouverture, laisser la possibilité aux développeurs d'utiliser un des autres langages afin qu'ils ne se sentent jamais bloqués. Cette fonctionnalité permet de profiter des particularités de chaque langage comme par exemple le calcul pour Fortran, l'orientation «système» pour le C,...

Les autres langages de programmation sont :

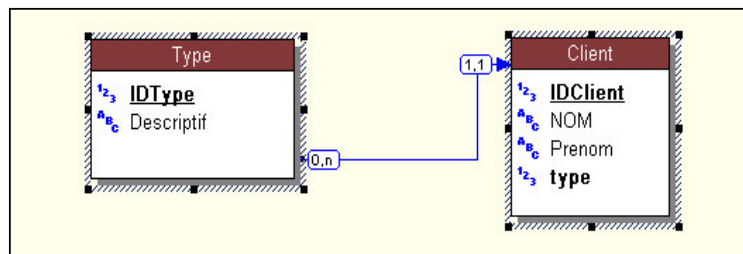
- C
- C++
- Visual Basic
- Pascal
- Cobol
- Fortran

Présentation de l'environnement de programmation

L'interface générale de Windev est comme toute plate-forme de développement Windows, elle est intuitive et complète à la fois.



Création des tables (fichiers) directement en visuel grâce au MLD de Windev.



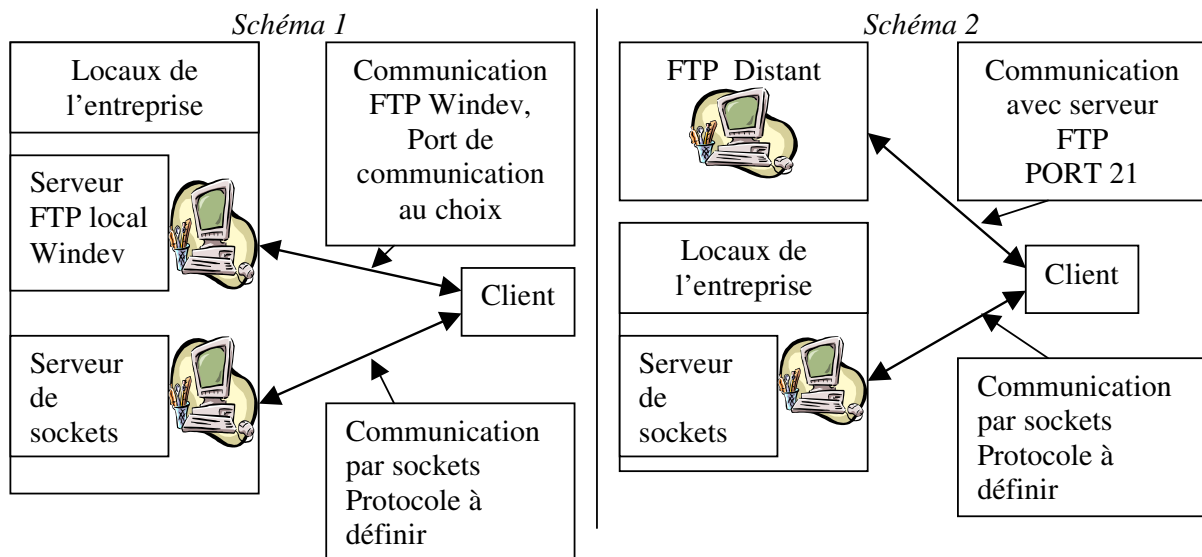
La plate-forme de travail est très conviviale et agréable à utiliser. Elle apporte beaucoup à la simplicité de ce langage (grâce à l'écriture intuitive ou la visualisation des différents objets présents dans le projet).

Analyse et prise de choix

Le sujet et la plate-forme de développement étant présentés, je vais maintenant passer à la phase d'analyse des solutions à apporter au projet.

Les communications externes

Dans un premier temps, il a fallu réfléchir aux solutions possibles de communications entre un client et l'entreprise, en prenant en compte que le développement serait réalisé avec Windev. Après quelques recherches, nous avons choisi de programmer toutes les communications grâce aux sockets et à des accès aux serveurs FTP (Soit FTP interne à l'entreprise avec un serveur géré par Windev, soit externe avec un serveur FTP distant que l'entreprise a sur paris). Les sockets apportent une assurance de communication entre le client et l'entreprise à condition que les ports de communications utilisés par notre application ne soit pas fermés chez le client. Donc deux schémas s'offrent à nous pour la suite de notre analyse.



Une fois ces deux solutions trouvées, il a fallu définir les avantages et les inconvénients de chacun des schémas.

Le schéma 1 apporte les avantages suivants :

- Implantation 100% contrôlée car développée avec Windev
- Communications possibles en interne entre le serveur de sockets et le serveur FTP local

Inconvénients :

- Rapidité de transfert du serveur FTP local au client (128Kbit/s)
- Serveur FTP développé en Windev n'est pas stable à 100% (problème de gestion des erreurs)

Le schéma 2 a les avantages suivants:

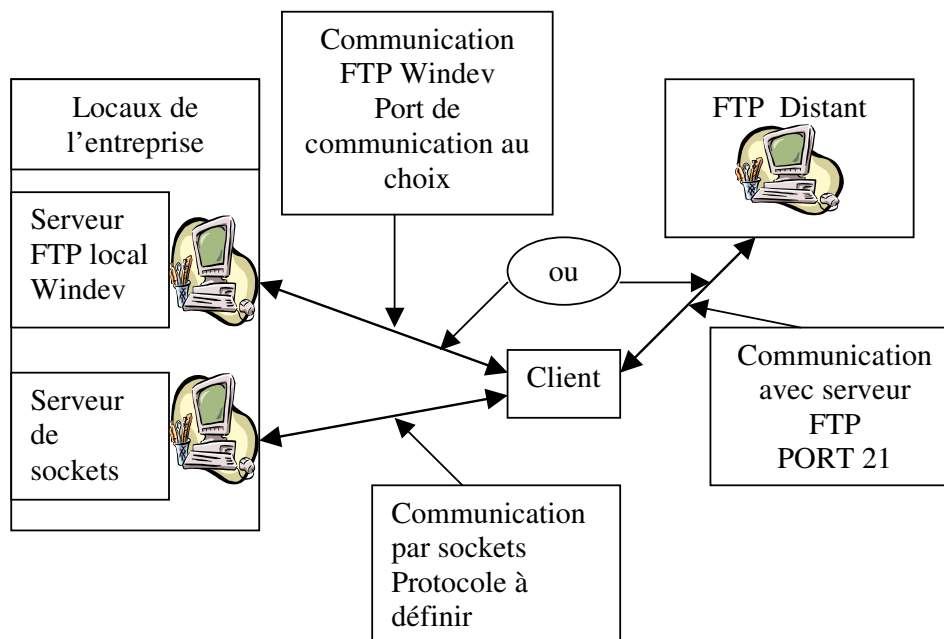
- Utilisation d'un vrai serveur FTP qui offre un débit de téléchargement élevé
- Plus d'encombrement de la bande passante de l'entreprise

Inconvénients :

- Gestion distante du serveur FTP (impossible de savoir quand il y a des perturbations)
- Temps de transfert des mises à jour vers le serveur FTP (problème redondant de la rapidité de la connexion ADSL en upload)

La solution aux problèmes rencontrés fut de garder tous les avantages en essayant d'avoir le moins d'inconvénients possibles. L'inconvénient principal est le temps de transfert d'une mise à jour vers un client, car plus la mise à jour est importante en taille, plus le temps est conséquent. Donc la solution finale de cette analyse est de faire un système de communication qui utilisera les deux serveurs FTP (distant et local). Cette solution répond aux attentes en terme de débit, puisque seront mises sur le serveur FTP distant toutes les mises à jour de tailles importantes et sur le serveur FTP local les autres MAJ de petites tailles. Ainsi chaque MAJ aura son serveur FTP qui lui sera affilié à sa mise en place.

Schéma final



La détermination pour savoir si le client doit se connecter au serveur FTP local ou FTP distant est réalisée grâce à la mise à jour que le client désire installer. Une fois le schéma de communication externe (car communication avec le client) mis en place, un autre problème doit être résolu. Car si l'on utilise les sockets pour communiquer entre deux modules, il faut faire attention à la sécurité des trames qui vont voyager par les sockets.

Le cryptage

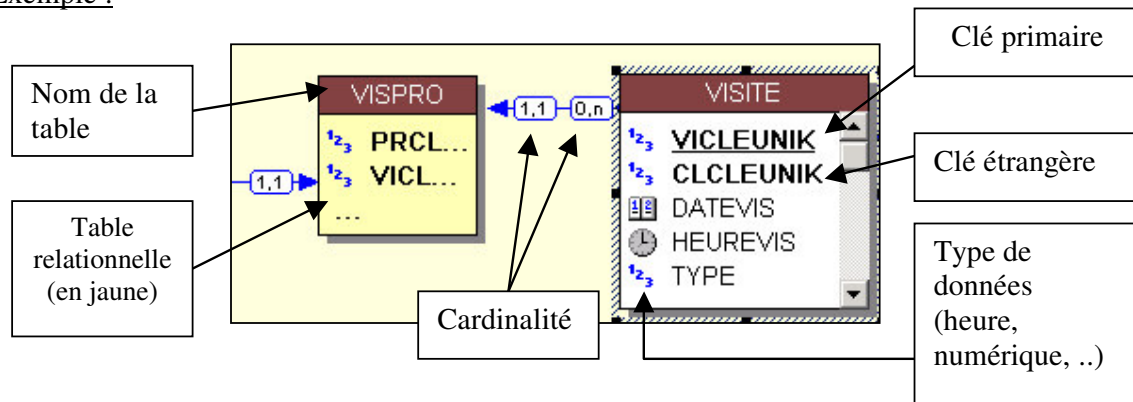
Pour s'assurer une sécurité de confidentialité entre le client et nous, on crypte les trames à l'aide des options de Windev. Windev apporte à la programmation la simplicité car il y a directement une fonction dans Windev qui permet de crypter selon plusieurs algorithmes. Celui que j'ai choisi d'utiliser est l'algorithme RC5 sur 128bits. Dans Windev, C5 est le cryptage de haute sécurisé par référence, avec l'utilisation d'une même clé secrète pour crypter et décrypter les trames.

La Base de données

Type de base de données

Windev incorpore un système de base de données, basé sur les hyper files. Les hyper-files sont des fichiers indexés à accès rapide. Dans la pratique, il est possible de voir ce fichier comme des tables puisque que rien ne les différencie. L'utilisation est similaire à toutes autres bases de données (comme Access ou aussi bien oracle). Windev offre la possibilité de faire toutes les actions désirées, comme il est possible de faire sur les SGBD existants à ce jour sur le marché. La représentation de ces tables est graphique (comme sous Access). Leur implémentation/modification est réalisée à partir de l'interface de Windev.

Exemple :



Avec ce type de base de données, tournée vers le visuel, la compréhension et l'implantation est simplifiée puisque l'implantation est l'image parfaite du Modèle Conceptuel de Données (MCD).

Sur ce SGBD de Windev, il y a la possibilité de lancer des requêtes en langage SQL de la forme suivante :

```
SELECT [DISTINCT] .... FROM ... [WHERE ....] [GROUP BY ....] [ORDER BY ...]
INSERT INTO ... VALUES ...
UPDATE ... SET ... WHERE ...
DELETE FROM .... WHERE ...
```

Ce sont les principaux ordres SQL pouvant être utilisés sur les fichiers de données.

La base de données existante

Comme dans toutes entreprises ayant un contact avec la clientèle, l'entreprise IDEATON-informatique a, elle aussi, un logiciel développé sous Windev qui leur permet de référencer leurs différents clients ainsi que leurs historiques. Ce logiciel, nommé DDMDL (Demande De Mise à jour De Logiciel), détient une base de données avec toutes les informations utiles au développement du projet. La base de données du DDMDL est composée de près de cinquante tables interconnectées qui leur permet d'avoir une gestion parfaite de leurs clientèles. Pour réaliser cette application, il faut savoir quel client possède quel produit, mais aussi des informations sur les supers utilisateurs (ce seront les seuls à pouvoir mettre en ligne une mise à jour). On a donc utilisé quatre tables qui existaient déjà dans le SGBD du DDMDL, et on a ajouté d'autres tables qui nous permettaient d'avoir une gestion parfaite des mises à jour mis en ligne et téléchargées par les différents clients.

Les tables réutilisées

Comme indiqué précédemment, il y a quatre tables du SGBD qui sont intéressantes pour la création de l'application. Ces tables sont :

- Table CLIENT
- Table PRODUIT
- Table relationnel CLIPRO
- Table UTILISATEUR

- **La table client**, comme son nom l'indique, comporte toute la clientèle de l'entreprise. Chaque client est référencé par un numéro de client unique et de 36 autres champs qui ne nous sont pas tous utiles. Les champs (de la table client) que l'on va utiliser pour cette application sont :

- Numéro de client (CLCLEUNIK est la clé primaire de la table)
- Le Nom
- La ville
- L'adresse

- **La table produit** est elle moins fournie que la précédente. Elle n'est composée que de 5 champs dont trois vont être réutilisés par la suite. Cette table détient les informations sur tous les logiciels qui ont été créés depuis le début.

Les champs qui nous sont utiles dans la table produit sont les suivants :

- Numéro de produit (PRCLEUNIK qui est la clé primaire de la table)
- Le Nom (le nom du produit)
- La description (du produit)

- **La table relationnel CLIPRO** est en fait une relation entre la table PRODUIT et la table CLIENT. Cette relation permet de savoir pour un client quel(s) logiciel(s) (ou produit(s)) il a acheté à l'entreprise. Cette table détient ainsi les informations suivantes :

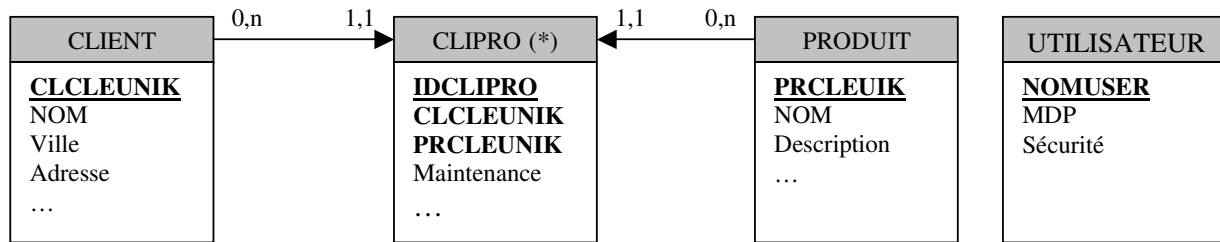
- Identifiant de la relation (IDCLIPRO)
- Numéro de client (CLCLEUNIK qui ici est une clé étrangère)
- Numéro de produit (PRCLEUNIK qui est ici une clé étrangère)
- Maintenance (qui permet de définir les clients qui ont le droit de mettre à jour leurs logiciels grâce au LiveUpdate)

- **La table Utilisateur** contient tout simplement les utilisateurs du logiciel DDMDL. Elle est composée des 3 champs suivants :

- NOMUSER (clé primaire sur le nom de l'utilisateur)
- MDP (mot de passe)
- Sécurité (qui est le niveau de sécurité de chaque utilisateur)

La table des utilisateurs est la seule des trois tables qui n'est pas en relation avec les 3 autres tables définies au-dessus.

Représentation des tables



(*) Dans Windev, les tables relationnelles ont l'obligation d'avoir une seule et unique clé primaire, contrairement à un développement sur base Oracle. Par exemple, nous aurions pu, pour la relation CLIPRO, définir comme clé unique CLCLEUNIK,PRCLEUNIK, mais ici avec la Base de données de type Hyper files, ce n'est pas possible.

Ajout et Modification de l'analyse

Afin d'implémenter l'application, on a eu besoin d'ajouter ou de modifier les tables de l'analyse de la base de données existantes, à cause de l'orientation, principalement basée vers le référencement et le traçage des mises à jour. Donc l'analyse existante, forte de sa cinquantaine de tables, va lui voir être ajoutée six nouvelles tables : trois spécifiques pour le traçage des clients qui se connecte, et trois autres qui vont être utilisées pour référencer les mises à jour qui seront disponibles sur les serveurs. Toutes ces tables seront interconnectées entre elle et entre les tables incontournables que sont CLIENT et PRODUIT.

Les tables pour la Traçabilité

Lors d'une connexion d'un client, ce dernier va s'identifier avec un login et un mot de passe. Par conséquent, il faut ajouter une table *Password*. Ensuite, pour garder un trace de toutes les connexions, il faut une table *Connexion* et une troisième table *Chemin* qui elle sauvegarde des informations spécifiques aux clients.

La table Password est en relation directe avec la table client. C'est ce qui va permettre à un client de s'identifier personnellement lors de l'utilisation du logiciel. Un Login et un Password seront spécifiques à un seul et unique client et un client pourra avoir de zéro à n (un entier indéfini) Login/Password. Les informations contenues dans cette table sont les suivantes :

- IDPassword (Clé primaire de cette table, ce numérique est le Login)
- Pass (comme prévu, ce champ est le Mot de passe)
- CLCLEUNIK (ce champ est la clé étrangère qui est reliée à la table CLIENT)

Ces trois champs sont suffisants pour l'identification d'un client déjà référencé dans la base de données.

La table Connexion est la table qui va permettre de garder toutes les connexions qu'un client X pourra avoir effectué au service de LiveUpdate. Les informations contenues dans cette table sont dans un premier temps le client, ensuite, la date de connexion puis son adresse IP, à titre indicatif, et aussi des informations sur son logiciel (si c'est la première connexion qu'il effectue ou encore s'il a modifié son chemin d'accès à son exécutable, ...). Cette table est en relation directe avec les tables Client et Produit puisque les informations contenues sont spécifiques à un client par rapport à un produit (logiciel). Pour avoir toutes ces informations, il a fallu les champs suivants :

- IDconnexion (Clé primaire de cette table)
- Date_connect (contient la date de cette connexion)
- IP_Cleint (adresse IP du client juste à titre d'informations préventives)
- Message (Information sur le logiciel du client)
- CLCLEUNIK (clé étrangère de la table client)
- PRCLEUNIK (clé étrangère de la table Produit)

La table Chemin est ici pour garder en mémoire les informations sur le répertoire où se trouve le logiciel. Cette table est en relation avec la table Produit et la table Password (non pas client puisqu'un client peut détenir plusieurs fois le même produit sur des réseaux ou des postes différents). De ce fait, un client peut, pour chaque installation que celui ci va détenir, avoir un login/password. Par conséquent, le chemin où se trouve le logiciel est unique à chaque login/password. Cette table est composée d'un nombre minimal de champs qui sont:

- IDChemin (clé primaire de la table)
- Chemin (chaîne de caractères contenant le chemin d'accès au logiciel du client)
- IDPassword (clé étrangère sur la table PASSWORD)
- PRCLEUNIK (clé étrangère sur la table PRODUIT)

Ainsi, ces trois tables permettent d'assurer une certaine sécurité lors de la connexion et aussi une traçabilité de chaque client qui se connecte au serveur de LiveUpdate.

Les tables pour le Référencement des Mises à Jours

Comme pour la partie précédente, 3 nouvelles tables ont été nécessaires pour avoir une gestion convenable des mises à jour qui vont être à la disposition des clients de la société. Pour la gestion des MAJ, il est obligatoire d'avoir une table nommée MAJ qui va contenir toutes les informations importantes d'une MAJ. Ensuite, il faut une table MAJ-CLI qui permet d'effectuer des spécifications de MAJ, c'est à dire qu'une MAJ est réservée à un client ou à un groupe de clients. Pour finir, il faut également une table Serveur qui référence les différents serveurs FTP disponibles pour stocker les MAJ.

La table MAJ (Mise à Jour) est une table centrale pour la gestion des MAJ. Elle détient les différentes informations indispensables au bon fonctionnement de l'application. Elle est en relation avec trois autres tables qui sont les tables PRODUIT, MAJ-CLI et Serveur. Les champs de cette table sont au nombre de 8 :

- IDMAJ (clé primaire de la table)
- Chemin_FTP (chemin où se trouve la MAJ sur le serveur FTP défini dans le champ IDServeur)
- Date_maj (date de mise en place de la MAJ)
- Descriptif_maj (description de la MAJ)
- Difficulté_install (niveau de l'installation, pour savoir le type d'installation (défini dans les pages suivantes))
- Réalisateur (Nom de la personne qui a mis en ligne la MAJ)
- PRCLEUNIK (clé étrangère de la table PRODUIT)
- IDServeur (clé étrangère de la table Serveur « FTP »)

La table Serveur est une liste de serveurs FTP définie afin de donner le choix aux super utilisateurs (les personnes qui mettent en ligne les MAJ) de choisir quel type de serveur sera utilisé (FTP local Windev ou Vrai serveur FTP). Pour cela, la table contient des informations telles que le nom, l'adresse et le password :

- IDServeur (clé primaire de la table)
- Nom (nom du serveur FTP)
- Adresse (adresse du serveur, ex : <ftp.ideation.fr>)
- Login (login de la connexion au serveur FTP)
- Password (mot de passe de la connexion)

La table MAJ-CLI est une table qui va permettre de spécifier une mise à jour à un ou des clients. Cette table est en relation avec les deux tables Client et MAJ, afin de répertorier pour une MAJ donnée, les utilisateurs qui ont le droit de l'exécuter (si aucune spécification n'est relevée dans cette table, pour une MAJ, c'est que cette MAJ est pour tous les clients). Les champs qui sont instaurés dans cette table sont les suivants :

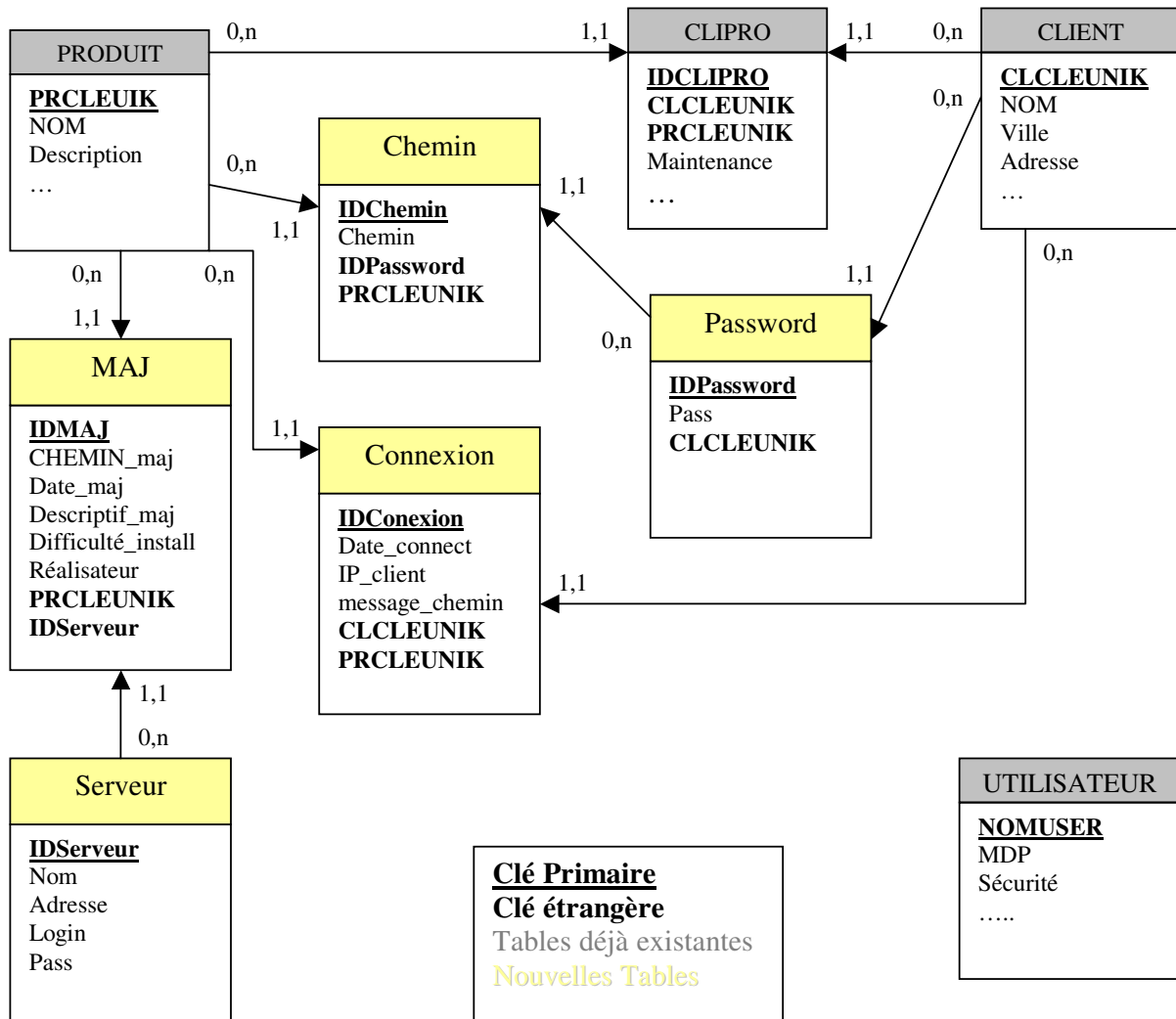
- IDMAJ-CLI (clé primaire de cette table)
- CLCLEUNIK (clé étrangère de la table client)
- IDMAJ (clé étrangère de la table MAJ)

Cette table est en définitif, une table de transition (relation) entre CLIENT et MAJ.

Interconnexion des tables

La présentation des différentes tables utilisées dans l'application étant finie, je vais maintenant vous présenter l'analyse qui les interconnecte entre elles. Cette analyse est restrictive, c'est à dire qu'elle ne va comporter que les tables utilisées (l'analyse complète sera disponible en annexe 1).

Schéma de l'analyse



Utilisation de la base de données

A l'origine, cette base de données n'est affiliée qu'à une seule et unique application. Mais pour les besoins de ce projet, on est dans l'obligation de l'utiliser en parallèle, entre deux applications. Ces deux applications sont concurrentes et réalisent des interrogations et des insertions (modifications), mais elles ne sont pas orientées vers le même but final. Le premier à l'utiliser est le DDMDL (Demande De Mise à jour De Logiciel). Il est plus orienté vers la gestion des clients et la visualisation des MAJ, alors que l'application utilisant cette base de données est en faite un serveur de sockets, qui, lui, va plutôt ajouter des MAJ ou interroger la base sur les produits et les mises à jour.

DDMDL

Ce logiciel possède tous les pouvoirs sur cette base de données puisque, à l'origine, elle a été développée spécifiquement pour lui. Il a donc accès à toutes les options possibles sur cette elle. Mais pour que le DDMDL garde tous les pouvoirs sur cette base de données, après les modifications apportées, il a fallu le modifier et lui ajouter de nouvelles fonctionnalités qui lui permettent aujourd'hui de pouvoir gérer, en partie, les MAJ. Le DDMDL a les capacités de modifier :

- les MAJ
- les spécifications de MAJ
- les Password
- les serveurs

Il peut aussi voir les informations concernant les dernières connexions des différents clients. Il peut également ajouter des MAJ, mais il ne peut pas le faire seul, il est obligé de passer par le serveur de sockets.

Serveur de sockets

Le but premier de ce serveur est de répondre aux questions que les clients vont lui envoyer (pour ce logiciel qu'elles sont les mises à jour possibles ?). Il est donc obligatoire qu'il soit développé à partir de la base de données qui contient les informations (donc la base de données du DDMDL).

Service LiveUpdate (LUD*)

Le serveur offre les services LUD comme prévu : il vérifie l'identification des clients puis il retourne des informations les concernant. Il garde aussi une trace de leurs passages sur le serveur afin de savoir quel client s'est connecté, quand et pour quel logiciel. Pour cela, le serveur de sockets est obligé de faire des accès en écriture sur la base de données, mais ce serveur et là aussi pour simplifier la démarche des clients qui se connectent car il sauvegarde des informations concernant l'emplacement des logiciels sur le disque dur des clients.

Service Mise à jours (MAJ*)

Le service MAJ est une idée qui est venue après le service LUD. Le concept est de pouvoir, à distance, grâce à ce serveur de sockets, mettre de nouvelles mises à jour en ligne. Pour cela ce serveur de sockets est obligé de faire des accès en lecture et écriture sur la base de données. Ainsi, les utilisateurs de cette base de données, à travers les deux applications existantes, ont un accès à deux services différents mais complémentaires qui sont l'information pour le DDMDL et le postage, ainsi que la réception de MAJ pour le serveur de sockets.

(* le principe des services LUD et MAJ sera approfondi par la suite)

Développement et communication en interne

Après avoir présenté la base de données utilisée pour les applications développées, je vais maintenant vous expliquer le développement des communications qui vont être réalisées en interne (dans l'entreprise). Comme défini juste avant, il existe une base de données partagée entre deux applications, mais également un serveur FTP interne, développé en Windev. Il y a donc des communications possibles entre modules deux à deux, comme les communications réalisées entre le DDMDL et le serveur de Sockets, de même qu'entre le serveur de sockets et le serveur FTP interne. De plus, une communication entre le DDMDL et le serveur FTP Local est nécessaire pour la transmission d'informations au serveur FTP.

Communication DDMDL-Serveur de Sockets

Ces deux applications bien distinctes ont la possibilité de communiquer de deux manières différentes : en premier lieu, elles peuvent utiliser les sockets qui sont un moyen de communication par messages. Puis, il existe la communication grâce à la base de données partagée.

Communication par sockets

L'application DDMDL a besoin de communiquer avec le serveur de sockets et cela ne se produit que dans un seul et unique cas : quand on demande au DDMDL de placer en ligne une nouvelle mise à jour. Les informations transmises, pour cette MAJ, sont alors mises en place. Il est possible que le DDMDL transmette directement le fichier (compressé au préalable) grâce à la communication par sockets au serveur de sockets (cette partie sera exposée dans la partie Protocole). La communication entre le DDMDL fonctionne sur un modèle d'acquiescement de trame. La communication est donc bidirectionnelle.

Base de données partagé

La base de données étant partagée, on peut la considérer comme étant une communication à mémoire partagée. C'est pourquoi, tout ce que l'une des applications va faire, la seconde aura la possibilité de le lire (de le retrouver). Un exemple simple qui tourne toujours autour de l'ajout de MAJ est que dans le DDMDL, lors de l'ajout d'une MAJ, on utilise le serveur de sockets car c'est le seul qui a la possibilité de les mettre en ligne. Ensuite le serveur de sockets sauvegarde cette dernière dans la table des MAJs et termine. Puis le DDMDL doit afficher sa liste des MAJs comprenant la nouvelle qui vient d'être mise dans la table par le serveur de sockets. Pour cela, il va faire une recherche dans la table pour trouver la nouvelle MAJ et l'afficher. Il existe donc une communication en mémoire partagée (Base de données partagée).

Communication Serveur de sockets – Serveur FTP Windev

Cette communication est simple et obligatoire : sachant que le serveur de sockets est l'application centrale du projet, il gère donc de A à Z la mise en ligne de MAJ. Par conséquent, la communication existante est juste un simple transfert de fichiers en interne. Cette copie de fichiers n'utilise pas la communication FTP, c'est le serveur de sockets qui va directement copier dans le bon répertoire le fichier MAJ compressé pour que le serveur FTP puisse le retrouver facilement quand on lui demandera.

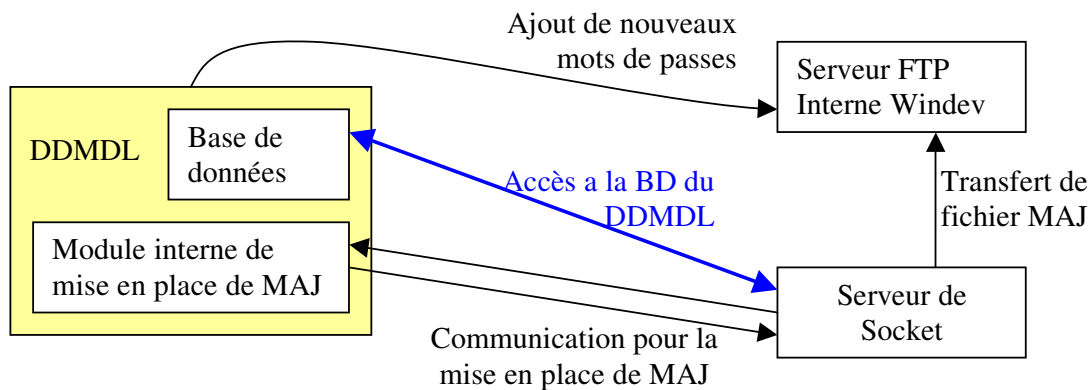
Communication DDMDL – Serveur FTP Local Windev

Cette communication est tout aussi basique que la précédente : on utilise le protocole FTP pour réaliser un transfert de fichiers. Le fichier concerné par ce transfert est le fichier des mots de passe de chaque client (car chaque client a la possibilité de se connecter au serveur FTP interne grâce à son propre mot de passe). Lors de la modification ou de l'ajout de mot de passe dans le logiciel DDMDL, le fichier modifié est automatiquement envoyé au serveur FTP local et celui-ci écrase l'ancien.

Schéma de communication interne

Pour les communications internes, le DDMDL comporte une base de données et un module de mise en place de mises à jour en interne qui communique avec le serveur de sockets. Le serveur de sockets, qui est le point central, communique avec tous.

Schéma interne



Ce schéma simplifié ne représente que les communications qui sont en interne.

Communication globale entre interne et externe

Dans un premier temps, je présenterai les communications existantes entre tous les éléments du projet, que ce soit des modules ou des applications. Chaque élément est obligatoirement en communication avec d'autres éléments. Les deux types de communication utilisés sont principalement les sockets, avec des protocoles à définir et le protocole FTP. Les 6 éléments utilisés du projet sont :

- Un serveur FTP Local Windev
- Un serveur FTP Distant
- Le logiciel interne de gestion (DDMDL)
- Un serveur de sockets (centre du projet)
- Un module client
- Un module externe (pour employer)

Utilité de chaque élément

- **Le serveur FTP Local** est développé avec Windev. Une des spécifications de ce serveur est que seul des clients (client FTP) programmés en Windev peuvent demander une connexion. Le principal inconvénient réside dans sa localisation géographique (les locaux de l'entreprise), ce qui implique une connexion peu rapide. C'est pourquoi ce serveur sera utilisé pour stocker les MAJs de petites tailles.
- **Le serveur FTP distant** est un serveur alloué sur Paris. C'est un serveur normal de stockage qui fonctionne comme tout autre serveur FTP.
- **Le DDMDL** est, comme je l'ai écrit précédemment, un logiciel de gestion de produits et de clients auquel j'ai apporté des modifications pour qu'il soit en mesure de gérer les MAJs ainsi que le suivi des clients qui se connectent au serveur de sockets. C'est aussi cette application qui contient la base de données centrale (présentation de l'interface en annexe 3).
- **Le serveur de sockets** est l'élément central de mon projet, c'est lui qui prend et qui communique avec toutes les autres entités de ce projet. C'est le seul élément qui communique avec les cinq autres. Il a également un accès direct à la base de données du DDMDL et répond à toutes les requêtes des différents modules (client ou externe).
- **Le module client** (LUP : Live Up Date) est réservé au client. Cette application lui permet de se connecter au serveur de sockets afin de récupérer des informations sur les MAJs. Ensuite, il se connecte au serveur FTP pour récupérer les MAJs proprement dites. Ce module installe uniquement la MAJ chez le client.
- **Le module externe** comporte deux parties : la première, comme pour le module client, a un système de LUD et la deuxième a un système de mises à jour en ligne à distance.

Récapitulatif de l'utilité des entités

Serveur de sockets

- Gestion des communications (par sockets)
- Gestion de la base de données du DDMDL

DDMDL

- Interrogation et modification de la base de données
- Visualisation des connexions pour chaque client
- Visualisation, ajout et modification des mots de passe (par client ou général)
- Visualisation, ajout et modification des spécifications
- Ajout de MAJ en ligne (avec une communication entre le serveur de sockets et le module MAJ du DDMDL)
- Modification de MAJ
- Gestion des serveurs FTP (ajout, modification et suppression)

Module Externe

- Ajout de MAJ en ligne (avec une communication entre le module MAJ et le serveur de sockets)
- Live UpDate qui permet l'installation d'une MAJ (avec une communication entre le module LUD et le serveur de sockets)

Module Client

- Uniquement réservé pour le Live UpDate qui permet l'installation d'une MAJ (communication entre le Module LUD et le serveur de sockets)

Les Protocoles

Cette partie ne va pas présenter les protocoles déjà définis, comme le FTP ou encore le FTP de Windev, car ces communications sont utilisées en état et donc ne sont pas modifiées. Mais elle exposera l'utilité des sockets pour communiquer entre les différents modules de ce projet, ceci, afin d'en sortir une communication cohérente qui permettra de faire fonctionner celui-ci. Pour cela, il a fallu mettre en place des protocoles de communications, fondés sur deux orientations : une, pour mettre une MAJ en ligne sur le serveur et l'autre, une communication protocolaire qui permet à un client de récupérer des informations sur les mises à jour à mettre en place sur la machine. La communication réalisée entre deux éléments du projet est toujours sur la base d'une trame de taille aléatoire qui sera cryptée à l'aide de l'algorithme RC5. Une autre règle est que chaque trame transmise sera obligatoirement acquittée par un message de taille non fixe selon la trame à acquitter.

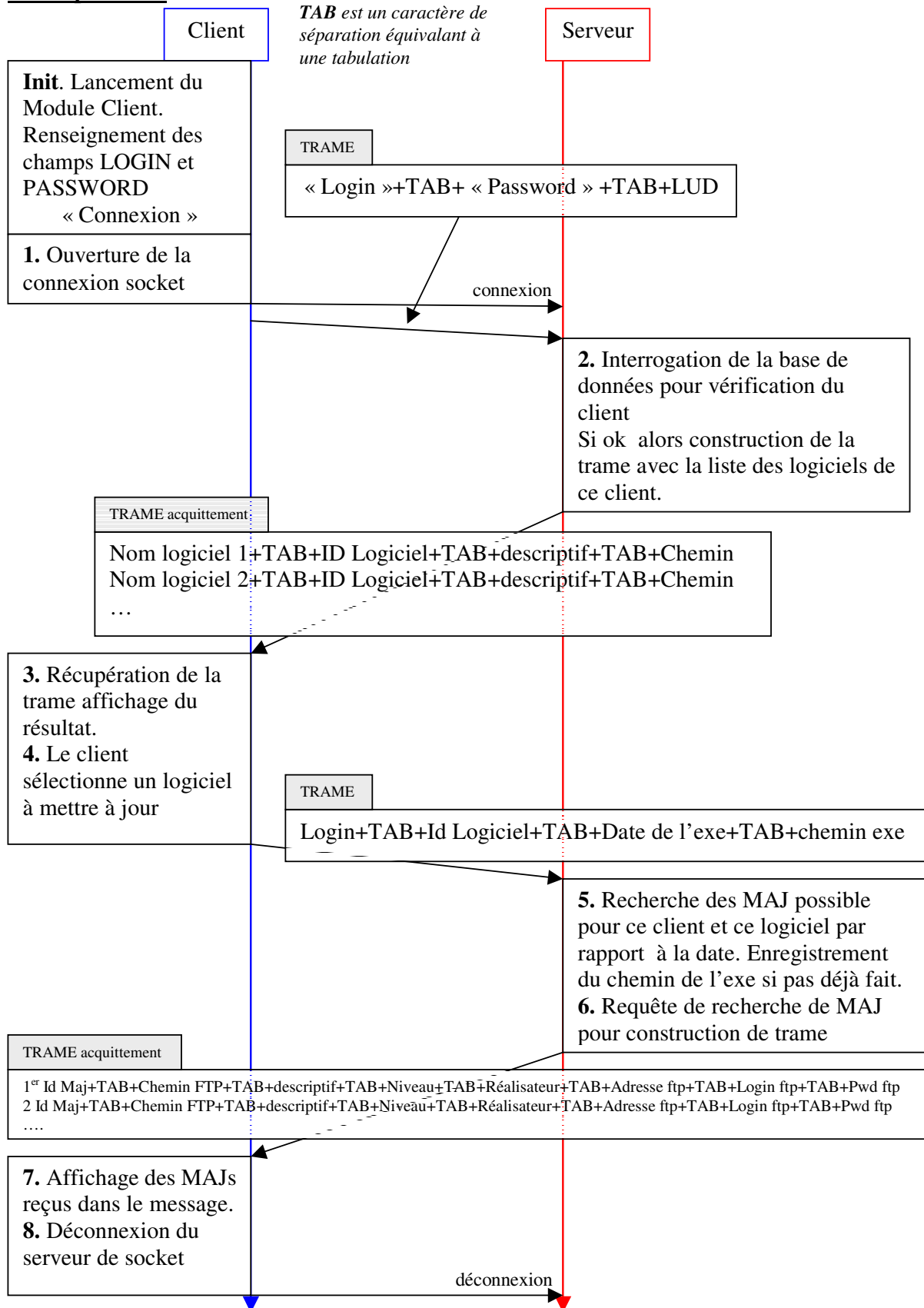
Exemple : Une trame client qui envoie le login et le mot de passe de celui-ci se verra acquitté par une trame qui contiendra la liste des logiciels de ce client (la taille de trame n'est donc pas fixe car un client peut avoir un ou plusieurs logiciels).

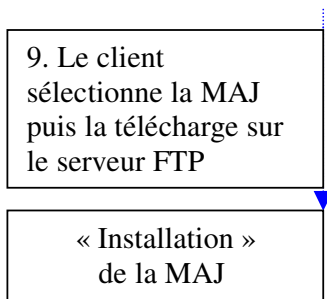
- Tout d'abord, la présentation se penchera vers le protocole le plus court qui permet à un client de s'identifier, puis de récupérer les informations sur les mises à jour qu'il a la possibilité de faire le jour où il se connecte au service.
- Ensuite une présentation du second protocole, plus complexe, sera exposée à son tour.

Protocole LUD

Le fonctionnement du protocole LUD est très simple. En effet, la procédure LUD peut être terminée en quatre trames (en ce qui concerne la récupération d'information ultérieure au téléchargement et à l'installation)

Début protocole





Le protocole présenté montre le fonctionnement dans le meilleur des cas. Mais il faut savoir que le client peut, à tout moment, demander l'interruption. Dans ce cas, il envoie au serveur de sockets la trame :

Login client+TAB+STOP

Le serveur répond alors « OK », ce qui implique une déconnexion immédiate.

D'autres problèmes effectifs, comme une mauvaise identification, sont gérés par le serveur qui retourne une trame contenant le mot « problème ». Ce mot est détecté par le module client qui met fin à la connexion.

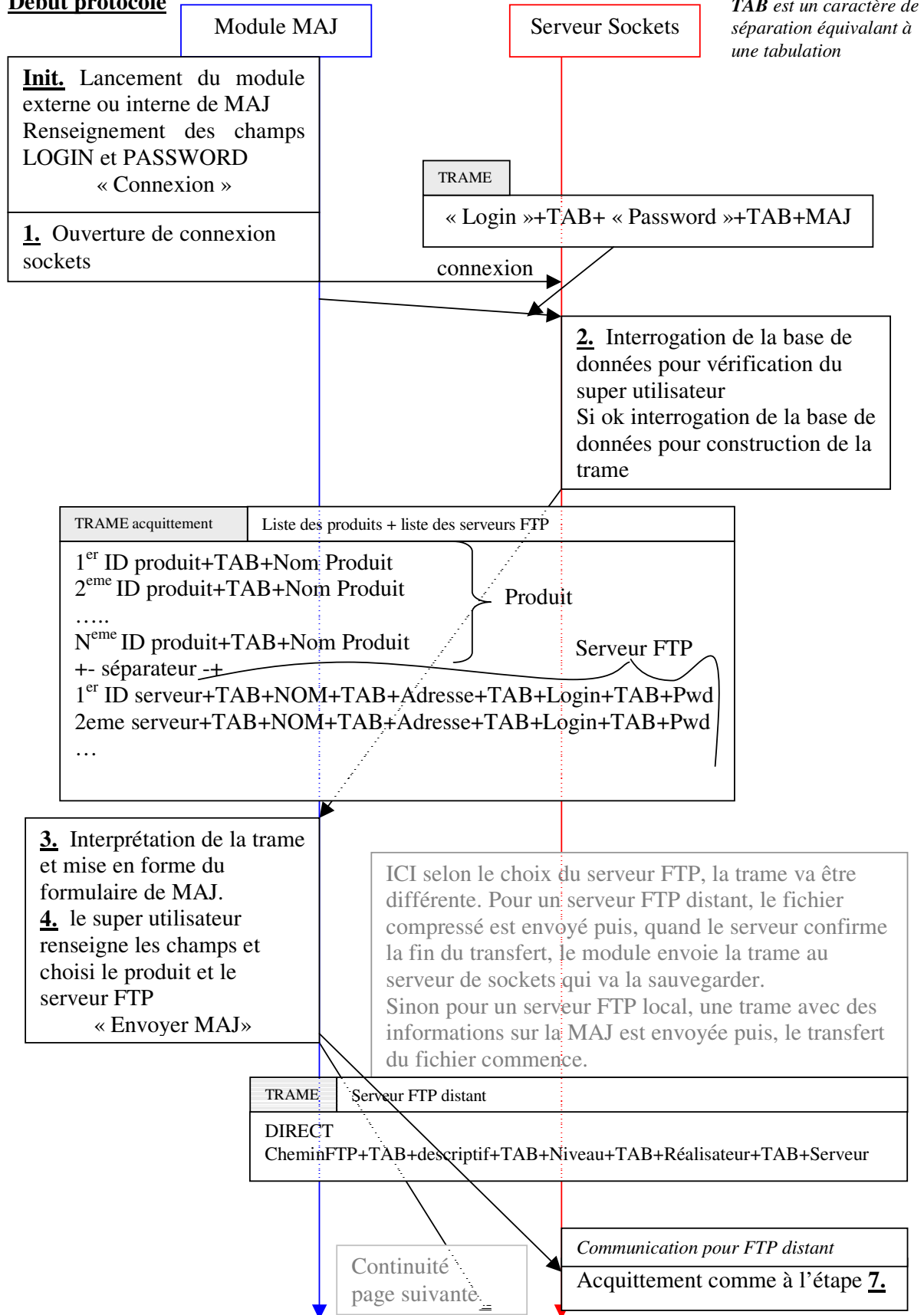
La sécurité des communications est assurée grâce à un cryptage des trames, à l'aide de l'algorithme RC5 qui fonctionne avec une clé privée (ce codage est réalisé sur 128bits). Les trames partantes sont en-cryptées grâce à RC5 128bits et lorsque le récepteur détecte un message, il le décrypte avec la même clé privée. L'utilisation du cryptage assure une confidentialité des communications de mots de passe et de Logins.

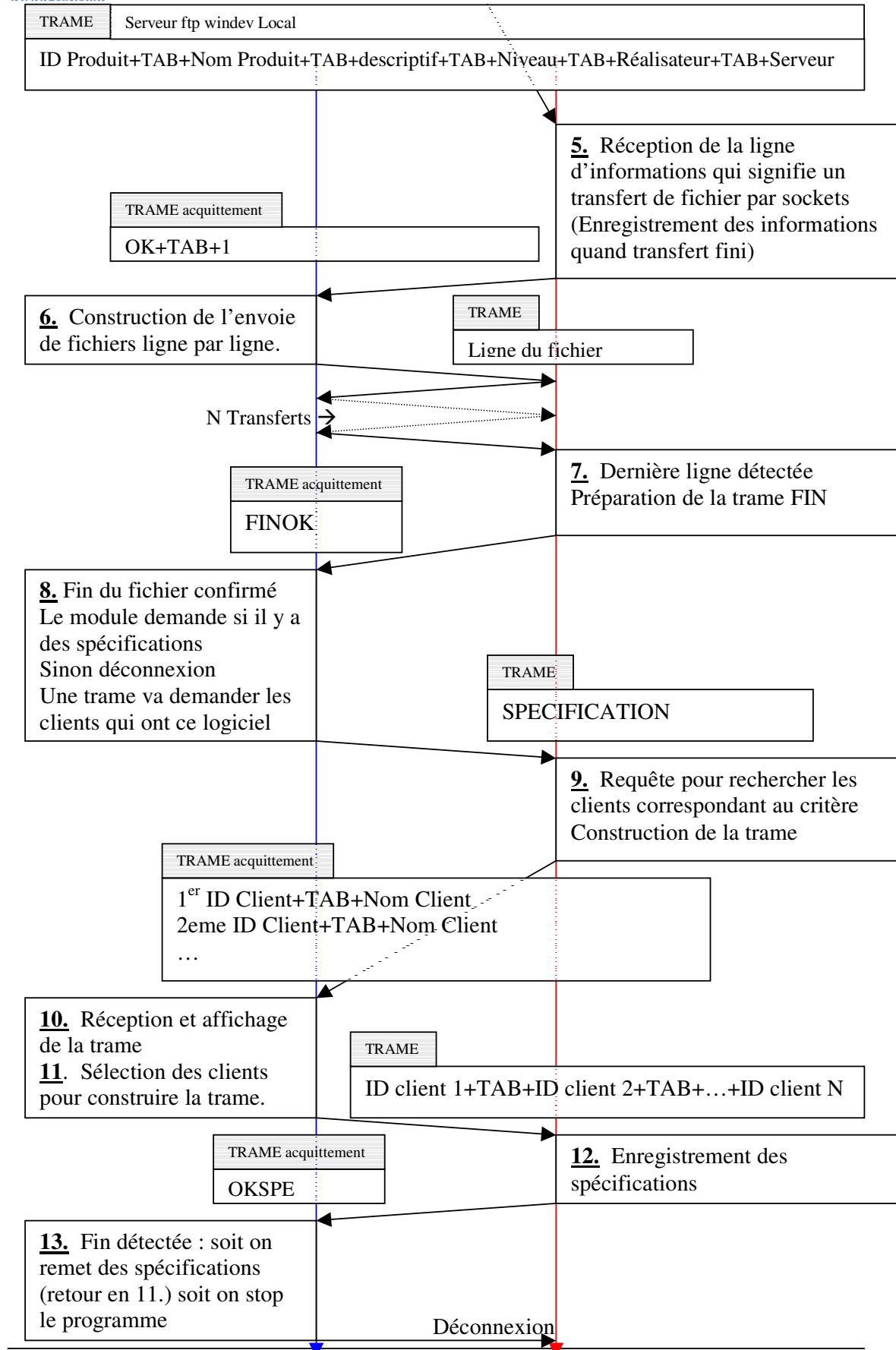
Protocole MAJ

Comme le protocole LUD, ce protocole est utilisé pour faire communiquer le module MAJ (interne et externe) avec le serveur de sockets. Le but de ce protocole est de permettre l'enregistrement de nouvelles MAJs pour que les clients puissent y accéder par la suite.

Le fonctionnement est le suivant : le super utilisateur (employé de l'entreprise) désire mettre une MAJ en ligne, soit du bureau, soit de chez un client (extérieur au bureau). Il se connecte donc à l'aide d'un des modules et entre son code qui part pour le serveur de sockets. Puis, le serveur lui retourne la liste des produits et la liste des serveurs FTP. Le client sélectionne et remplit alors les champs puis valide la MAJ. Ces informations sont envoyées au serveur qui, selon le serveur FTP choisi, va agir de deux manières différentes : si le serveur FTP choisi est en local, alors le fichier sera transmis par les sockets, sinon le serveur FTP est un serveur distant. Ce qui a pour effet que l'envoi du fichier est directement réalisé du module au serveur FTP (en FTP). Une fois qu'est validée la réception de ce fichier, le module envoie au serveur de sockets un message de fin de transfert. Le module client demande alors si la MAJ est spécifique à un ou des clients. Si oui, le module client demande au serveur de sockets les clients qui ont ce logiciel. Le module réceptionne ensuite la liste et l'affiche. Le super utilisateur sélectionne ainsi les clients spécifiques. S'il y en a, un autre message retourne au serveur la liste des clients correspondants, et celui-ci acquitte cette trame. Enfin, suite à l'acquiescement, le module se déconnecte et la MAJ est en ligne.

Début protocole





Les communications, comme définies pour le module MAJ, peuvent être à tout moment interrompues comme défini dans le protocole précédent. Il est aussi possible d'annuler le transfert d'une MAJ sur le serveur FTP local ou sur le serveur FTP distant.

Gestion des communications

Les communications sont divisées en deux parties : le client et le serveur. Le client est l'initiateur de la communication, en demandant une communication avec le serveur. Le serveur, lui, est en attente de communications. Quand il accepte une communication, il lance un thread qui sera réservé à ce client. Ce thread lance alors une procédure de gestion de communications qui répond aux différentes demandes des clients. Le client étant l'initiateur, c'est à lui de commencer à transmettre une trame d'identification qui sera acquittée par une trame réponse, en provenance du serveur. Toutes trames du client sera ainsi acquittées obligatoirement. La fin de la communication est aussi à la charge du client qui lance une trame « STOP ». A cet instant, les communications sont rompues par un `socketferme()`. Le client utilise aussi les threads afin de gérer la réception des messages. Tous les messages émis pas le client sont effectués suite à une démarche de l'utilisateur (sauf l'envoi de fichiers). Mais la réception est réalisée à l'aide d'un thread qui tourne et écoute la socket. Le serveur est en mesure de gérer un nombre important de clients en même temps grâce à l'utilisation parallèle des threads. Ce serveur est obligé d'utiliser des sections critiques afin de limiter les accès simultanés aux mêmes tables (Des parties de code seront mis en annexe).

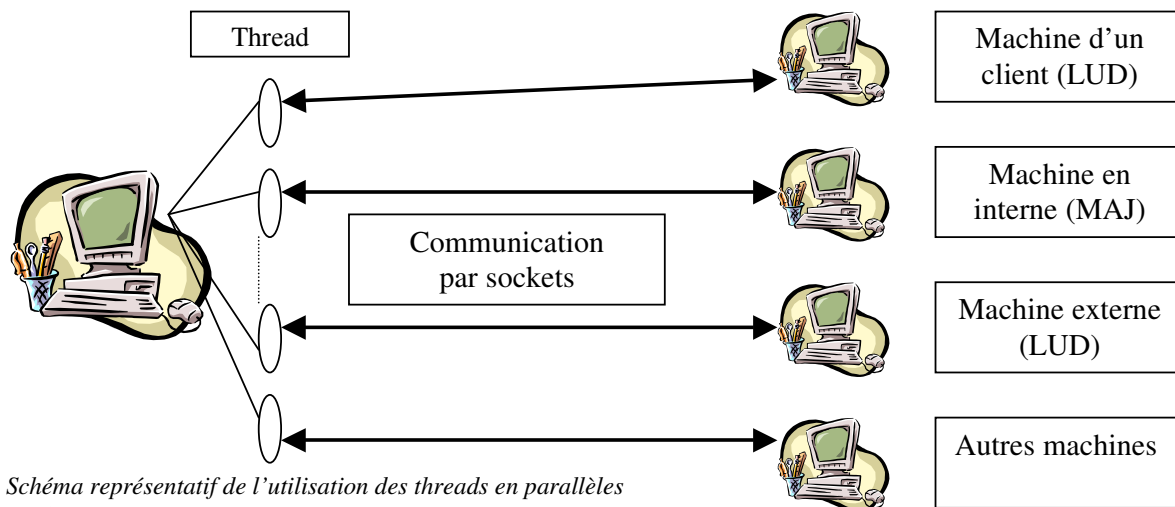


Schéma représentatif de l'utilisation des threads en parallèles sur le serveur de socket

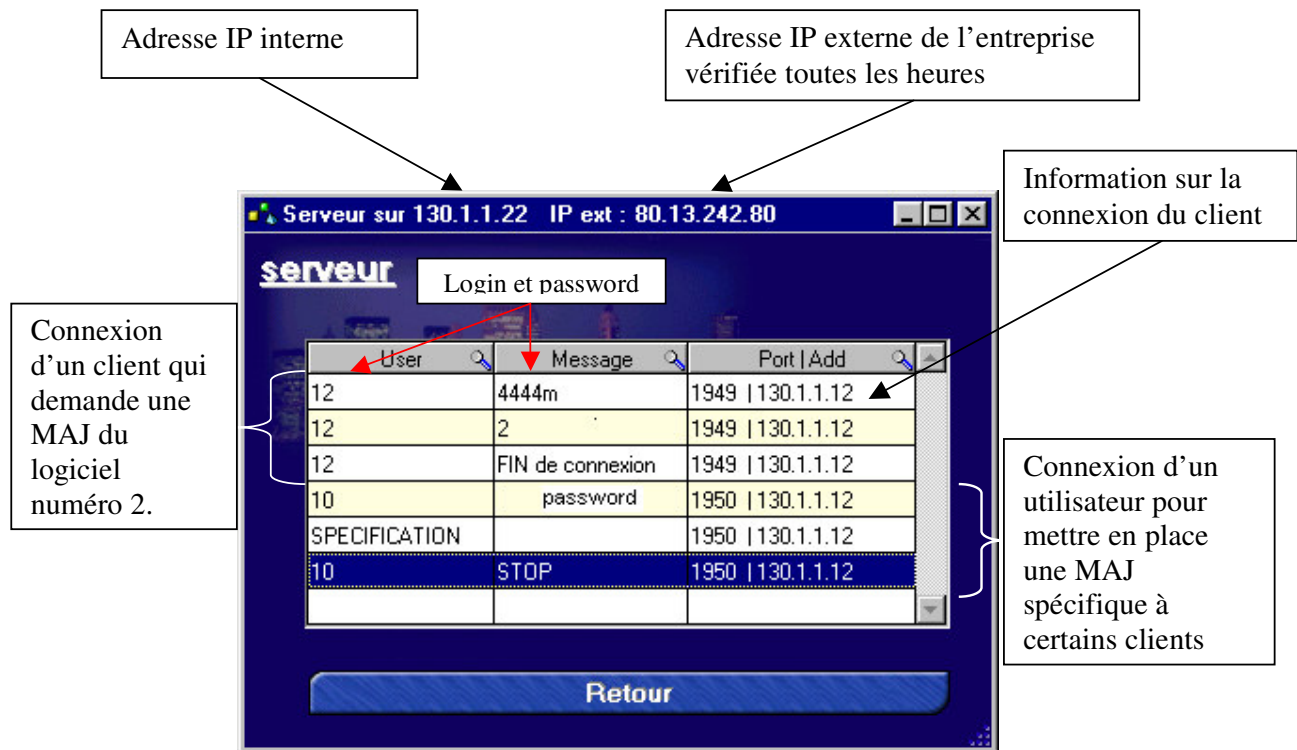
Toutes les communications sont gérées afin de ne laisser aucune question sans réponse. Le client a la possibilité de faire des choses simples (demander ou télécharger les MAJs), mais il est plus limité sur les installations. A tout moment, il peut couper la communication sans que cela est une influence sur le système.

Interface Homme-machine

Sur toutes les applications réalisées, je vais vous présenter en détail les trois plus importantes, c'est à dire le serveur de sockets ensuite, le module de mise en place de MAJ (Module MAJ) et pour finir le module client (LUD live UpDate). Les autres interfaces, comme tous les ajouts qu'il y a eu à faire dans l'application DDMDL, seront présentées en annexe 3.

Le Serveur de Sockets

Pièce centrale de ce projet, cette application n'a pas été la plus facile à programmer. Mais son interface homme machine est très simple puisque aucune opération ne doit être réalisée dessus. Cette interface doit juste répertorier les connexions du jour pour voir ce qui a été réalisé et voir l'affluence qu'il y a eu sur ce module de MAJ. L'interface présentée est précédée d'un menu simple qui permet de lancer ce module.



The screenshot shows a window titled "Serveur sur 130.1.1.22 IP ext : 80.13.242.80". The interface includes a "Login et password" field, a table of connections, and a "Retour" button. Callouts provide the following information:

- Adresse IP interne**: Points to the window title.
- Adresse IP externe de l'entreprise vérifiée toutes les heures**: Points to the window title.
- Information sur la connexion du client**: Points to the table headers.
- Connexion d'un client qui demande une MAJ du logiciel numéro 2.**: Points to the first row of the table.
- Connexion d'un utilisateur pour mettre en place une MAJ spécifique à certains clients**: Points to the last row of the table.

User	Message	Port	Add
12	4444m	1949	130.1.1.12
12	2	1949	130.1.1.12
12	FIN de connexion	1949	130.1.1.12
10	password	1950	130.1.1.12
SPECIFICATION			
10	STOP	1950	130.1.1.12

Interface principale du serveur de sockets

L'affichage est séquentiel, mais il peut se chevaucher. Cela signifie que les clients peuvent se connecter tous au même moment, ce qui affichera x début de connexion. Puis ils peuvent se déconnecter quand ils veulent, l'affichage est là, juste à but indicatif. L'adresse IP externe est vérifiée régulièrement car une connexion ADSL change d'IP toutes les 24H. Il y a par conséquent, une vérification et une mise à jour de l'adresse IP en moyenne toutes les heures. La complexité de ce module est la programmation qu'il a fallu instaurer derrière cette interface ; en Annexe 2 la fonction principale de gestion des MAJ sera commentée.

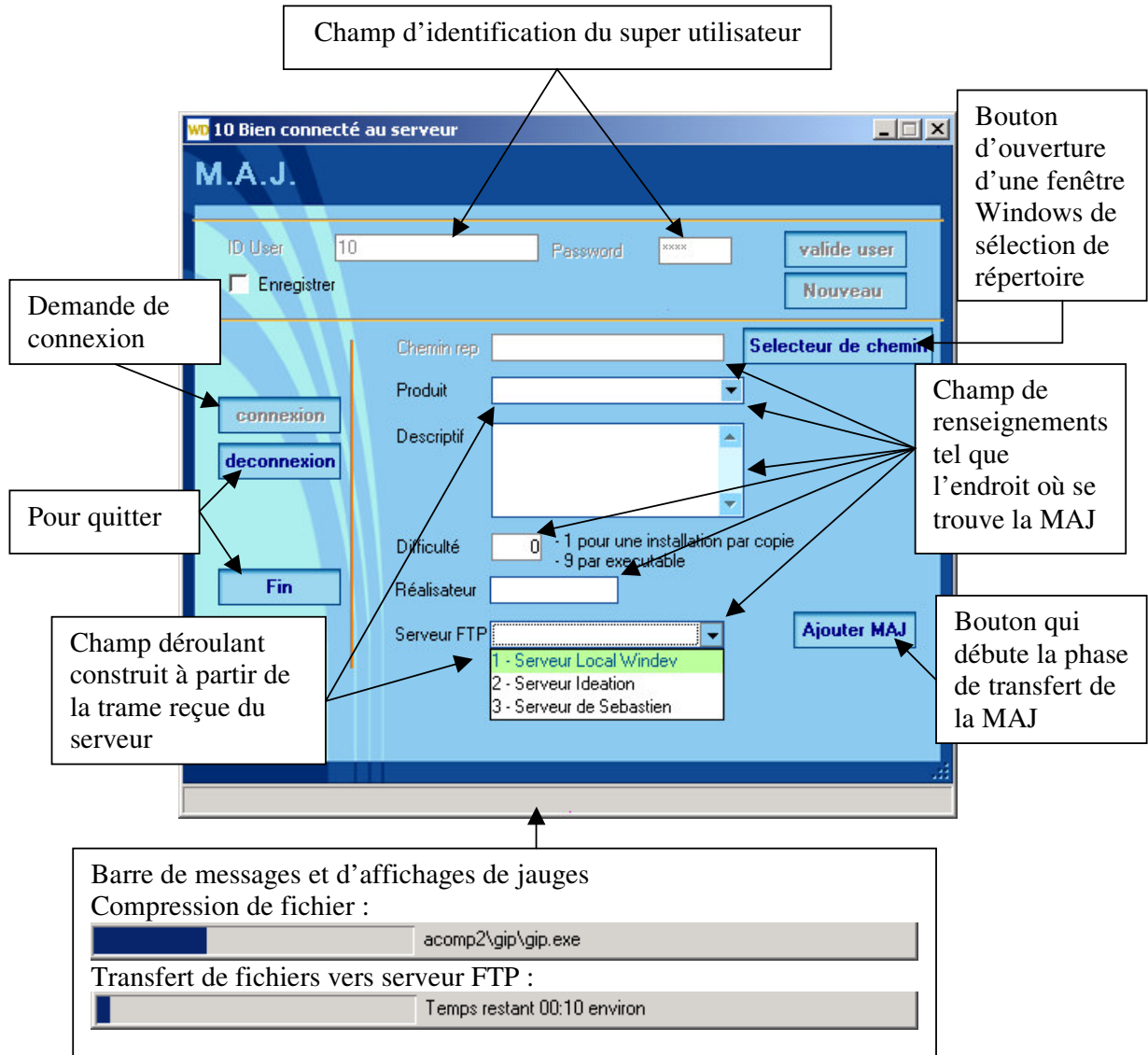
Interface du module MAJ

Ce module, de mise en place de MAJs, est décomposé en 2 phases :

1. phase de renseignements sur la nouvelle MAJ à mettre en ligne avec la phase d'envoi d'informations et du fichier aux serveurs.
2. phase de spécification : si la MAJ doit être réservée pour un client ou des clients.

La présentation va elle aussi être en 2 phases :

Phase 1 : mise en place de la MAJ



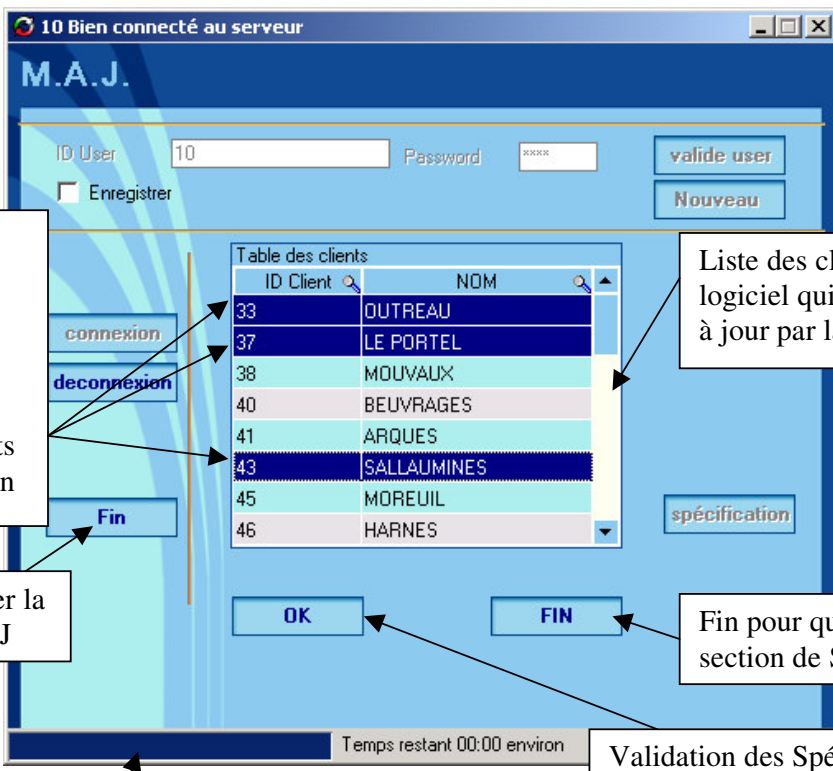
The screenshot shows a Windows application window titled 'M.A.J.' with a status bar 'W0 10 Bien connecté au serveur'. The interface includes a login section with 'ID User' (10) and 'Password' (masked), buttons for 'valide user' and 'Nouveau', and an 'Enregistrer' checkbox. A 'connexion' button is on the left, with 'deconnexion' and 'Fin' below it. The main area contains fields for 'Chemin rep', 'Produit', 'Descriptif', 'Difficulté' (0-9), 'Réalisateur', and a 'Serveur FTP' dropdown menu. A 'Selecteur de chemin' button is next to the 'Chemin rep' field. An 'Ajouter MAJ' button is at the bottom right. At the bottom, a message bar shows progress for 'Compression de fichier' and 'Transfert de fichiers vers serveur FTP'.

Annotations and their corresponding elements:

- Champ d'identification du super utilisateur: Points to the 'ID User' and 'Password' fields.
- Bouton d'ouverture d'une fenêtre Windows de sélection de répertoire: Points to the 'Selecteur de chemin' button.
- Demande de connexion: Points to the 'connexion' button.
- Pour quitter: Points to the 'deconnexion' and 'Fin' buttons.
- Champ déroulant construit à partir de la trame reçue du serveur: Points to the 'Serveur FTP' dropdown menu.
- Champ de renseignements tel que l'endroit où se trouve la MAJ: Points to the 'Chemin rep', 'Produit', and 'Descriptif' fields.
- Bouton qui débute la phase de transfert de la MAJ: Points to the 'Ajouter MAJ' button.
- Barre de messages et d'affichages de jauges: Points to the bottom status bar.

Une fois validées en appuyant sur « Ajouter MAJ », les informations sont transmises au serveur qui va les sauvegarder dans la base de données. Une fois cette sauvegarde effectuée, le serveur retourne un message qui permet au module de savoir que le travail est bien fini et qu'il faut demander si la MAJ est spécifique à un ou des clients. Si la réponse est non alors le programme MAJ se termine ici, sinon il passe à la page que je vous présente ensuite.

Phase 2 : Spécification de MAJ



The screenshot shows a window titled "10 Bien connecté au serveur" with the "M.A.J." interface. It includes a login section with "ID User" (10) and "Password" (XXXX), and buttons for "valide user" and "Nouveau". Below is a "Table des clients" with columns "ID Client" and "NOM". The table lists clients: 33 (OUTREAU), 37 (LE PORTEL), 38 (MOUVAUX), 40 (BEUVRAGES), 41 (ARQUES), 43 (SALLAUMINES), 45 (MOREUIL), and 46 (HARNES). Buttons for "connexion", "deconnexion", "Fin", "OK", and "FIN" are visible. A status bar at the bottom shows "Temps restant 00:00 environ".

Annotations:

- La liste comporte une sélection multiple qui permet de spécifier plusieurs clients en une sélection
- Liste des clients ayant le logiciel qui peut être mis à jour par la MAJ.
- Fin pour quitter la fenêtre MAJ
- Fin pour quitter la section de Spécification
- Validation des Spécifications et envoi au serveur de sockets
- Jauge disant que le fichier est bien arrivé

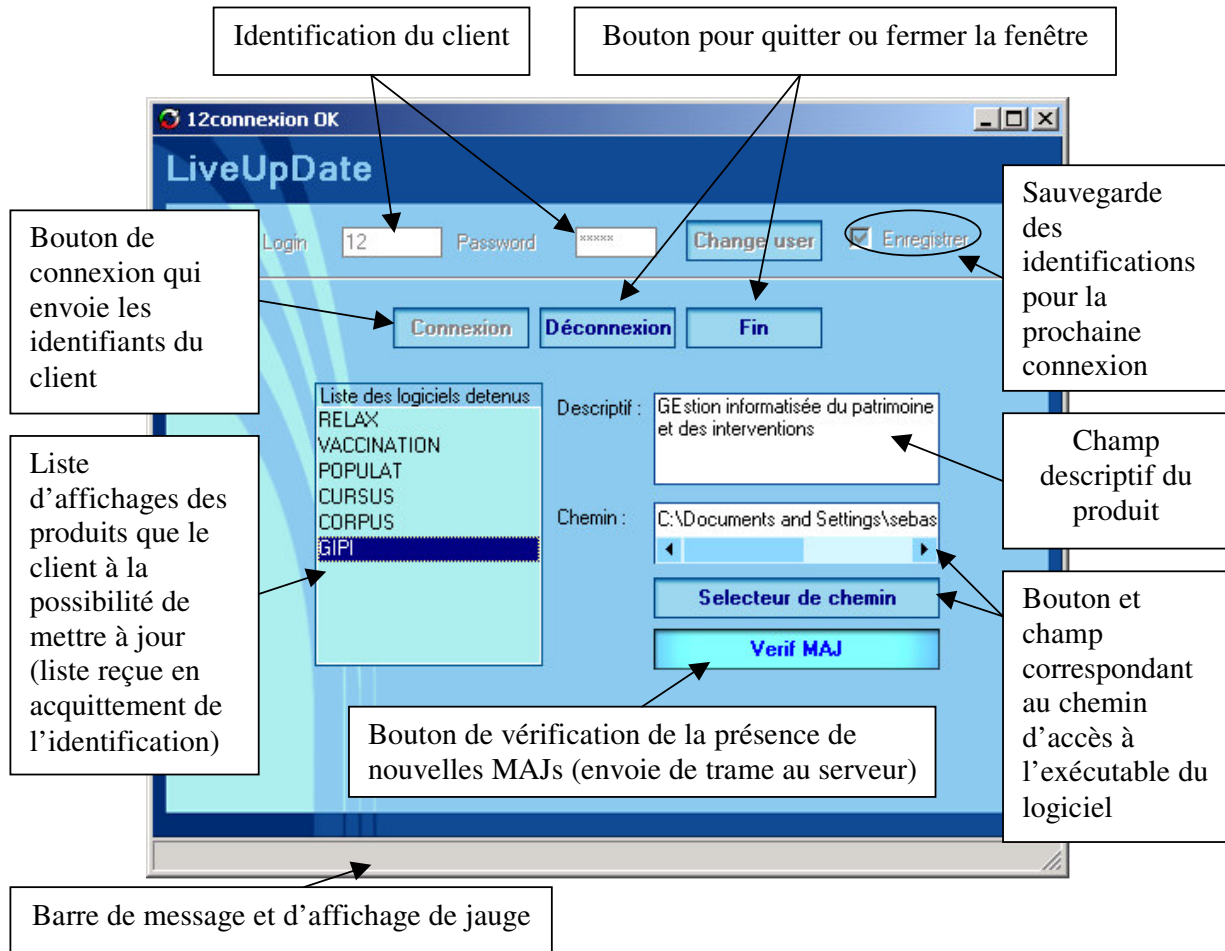
Lorsque l'on valide par « OK », la liste des clients sélectionnés est transmise au serveur de sockets. Puis, une fois acquittée cette trame, le module client propose à nouveau de spécifier des clients. Si l'on désire en finir là, on appuie sur le bouton « FIN » (l'un des 2 disponibles). Cette interface est l'une des plus complexes réalisée puisqu'il y a de nombreux objets qui sont présents sur cette fenêtre. Il faut les gérer en affichage et en réactivité au mouvement de la souris. Dans la suite le module réservé au client (LUD) vous sera présenté. Celui-ci est plus intuitif.

Interface du module LUD

Comme dans la présentation précédente, ce module fonctionne en 2 phases consécutives et complémentaires. Ces deux phases sont :

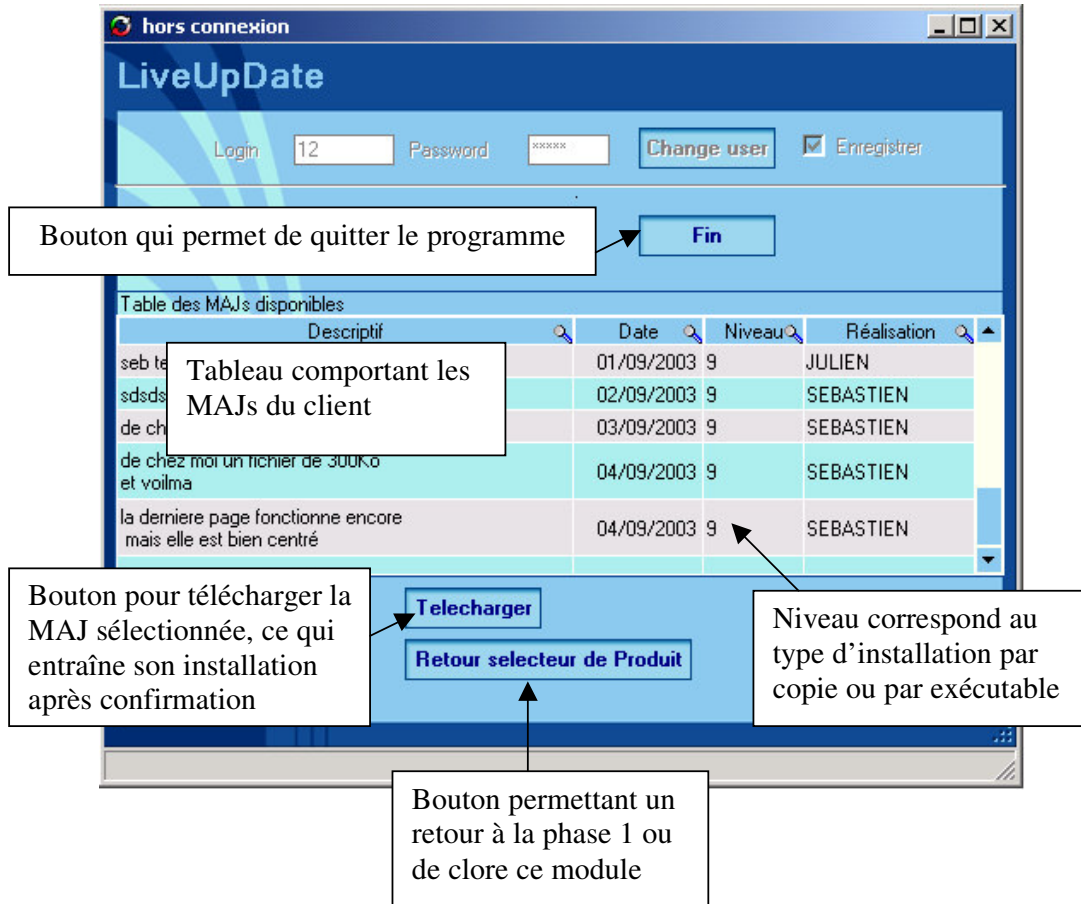
1. Phase d'identification et de sélection du produit à mettre à jour.
2. phase de sélection de la MAJ à télécharger et à installer.

Phase 1 : Identification et sélection du produit à mettre à jour



La première phase terminée par l'envoi d'une trame, suite à l'appui du client sur le bouton « Verif MAJ », le serveur réceptionne la trame qui contient l'ID_logiciel, l'id du client et aussi la date de la version du logiciel. De là, le serveur retourne toutes les MAJs possibles : celles qui sont communes et celles qui sont spécifiques à ce client. La phase 2 commence à la réception de la trame des MAJs.

Phase 2 : sélection de la MAJ à télécharger et à installer



Bouton qui permet de quitter le programme

Fin

Tableau comportant les MAJs du client

Descriptif	Date	Niveau	Réalisation
seb te	01/09/2003	9	JULIEN
sdsds	02/09/2003	9	SEBASTIEN
de ch	03/09/2003	9	SEBASTIEN
de chez moi un fichier de 300Ko et voilma	04/09/2003	9	SEBASTIEN
la derniere page fonctionne encore mais elle est bien centré	04/09/2003	9	SEBASTIEN

Bouton pour télécharger la MAJ sélectionnée, ce qui entraîne son installation après confirmation

Telecharger

Retour selecteur de Produit

Niveau correspond au type d'installation par copie ou par exécutable

Bouton permettant un retour à la phase 1 ou de clore ce module

A la suite de cette phase, le fichier compressé, qui contient la MAJ, est téléchargé. Puis il est décompressé et l'on demande au client si l'on doit installer cette MAJ. Si oui, la procédure d'installation selon le niveau va se réaliser. Si l'installation s'est bien passée, des messages d'informations en informe le client. Une fois fini, le client n'a plus qu'à fermer la fenêtre.

Je viens de vous présenter les interfaces les plus importantes du projet, les autres seront présentées en Annexe 3.

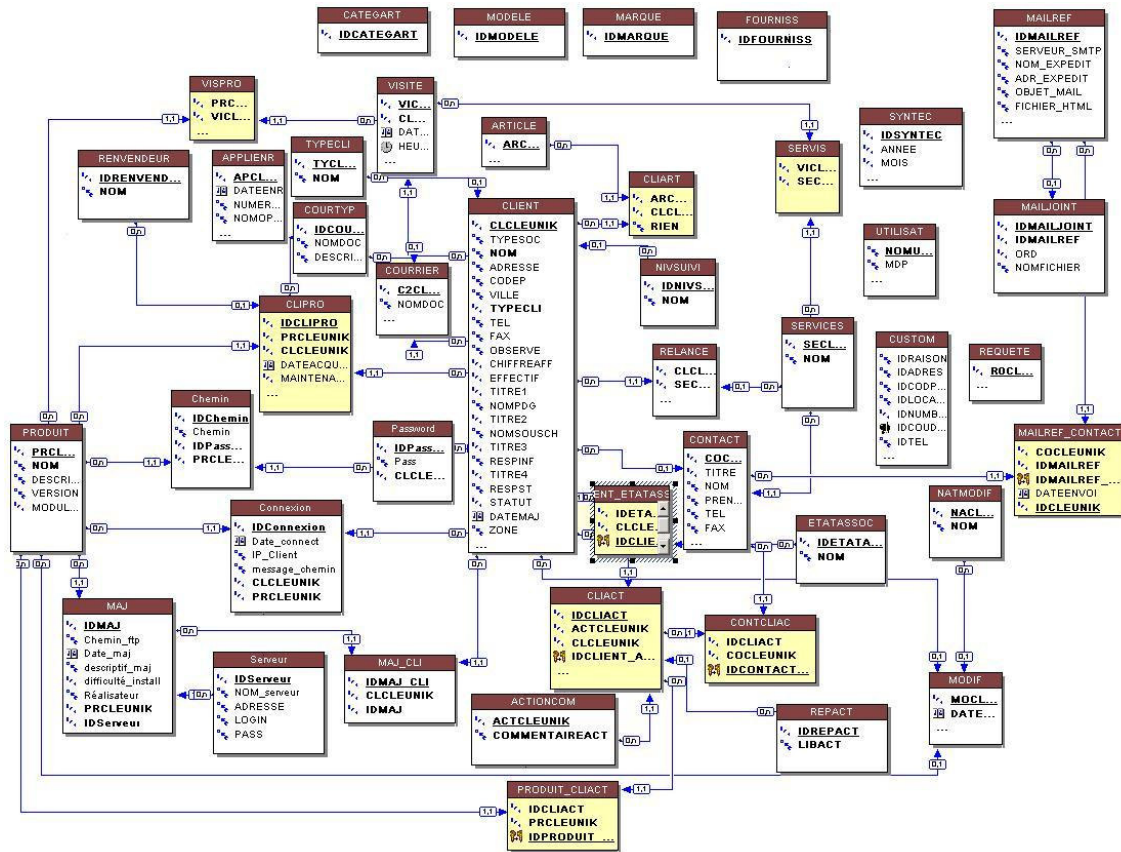
Conclusion

Durant ce stage de fin d'études, j'ai redécouvert le monde de l'entreprise à travers les obligations de résultats et les obligations en terme de temps de création. Cette expérience m'a fait prendre conscience des capacités techniques que j'ai acquises tout au long de mon cursus universitaire et plus certainement cette dernière année. J'espère que ce dernier stage va me propulser dans le monde du travail très prochainement.

Glossaire

BD :	Base de Données
DDMDL :	Demande De Mise à jour De Logiciel
FTP :	File Transfert Protocol
LUD :	Live UpDate
MAJ :	Mise à Jour
MCD :	Model Conceptuel de Données
MLD :	Model Logique de Données
PC-SOFT :	Concepteur de la plat forme Windev
RC5 :	Outil cryptographique à clé privé
Trame :	Message transmit sur le réseau

- Annexe 1 -

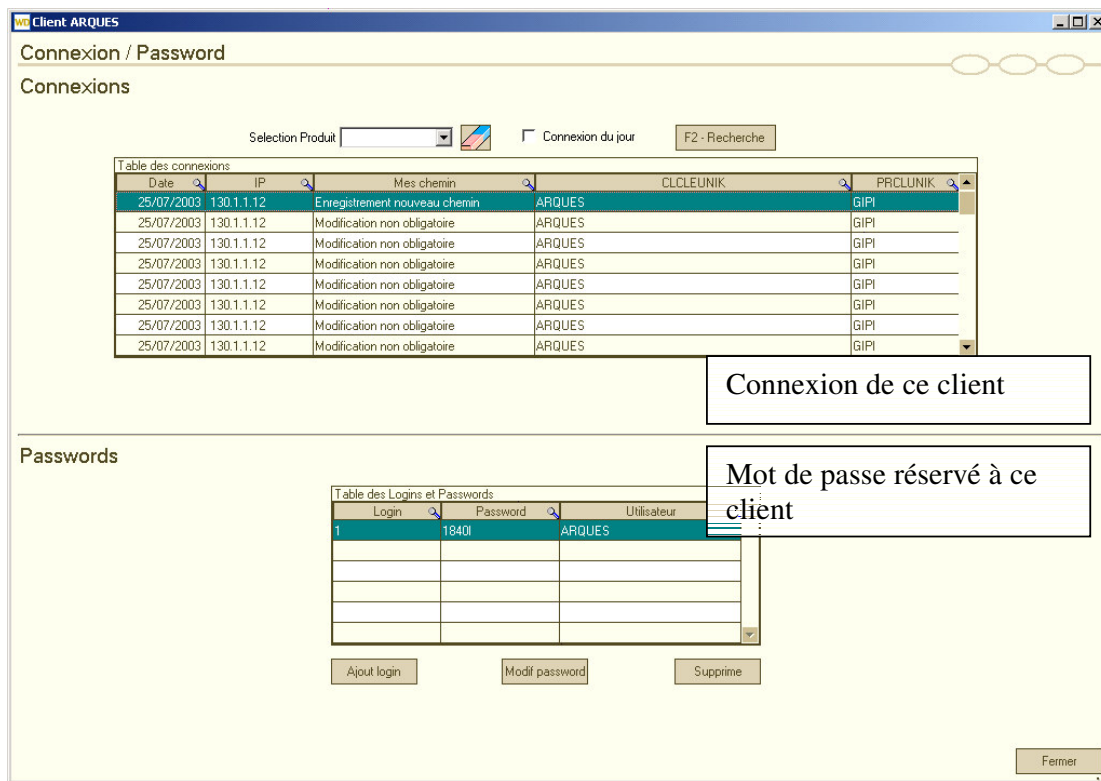


- Annexe 2 -

- Annexe 3 -

Fenêtre du DDMDL

Cette page est spécifique à un client.



Client ARQUES

Connexion / Password

Connexions

Selection Produit [] Connexion du jour [] F2- Recherche

Date	IP	Mes chemin	CLCLEUNIK	PRCLUNIK
25/07/2003	130.1.1.12	Enregistrement nouveau chemin	ARQUES	GIPI
25/07/2003	130.1.1.12	Modification non obligatoire	ARQUES	GIPI
25/07/2003	130.1.1.12	Modification non obligatoire	ARQUES	GIPI
25/07/2003	130.1.1.12	Modification non obligatoire	ARQUES	GIPI
25/07/2003	130.1.1.12	Modification non obligatoire	ARQUES	GIPI
25/07/2003	130.1.1.12	Modification non obligatoire	ARQUES	GIPI
25/07/2003	130.1.1.12	Modification non obligatoire	ARQUES	GIPI
25/07/2003	130.1.1.12	Modification non obligatoire	ARQUES	GIPI

Connexion de ce client

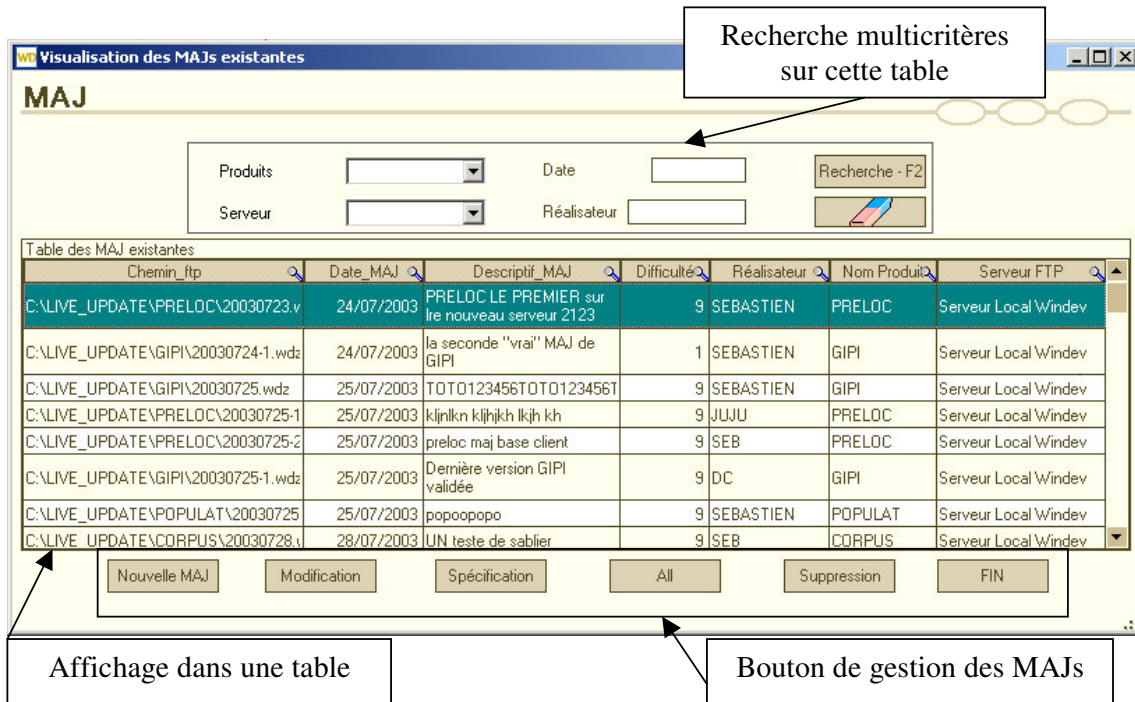
Passwords

Login	Password	Utilisateur
1	1840I	ARQUES

Mot de passe réservé à ce client

Ajout login Modif password Supprime Fermer

Gestion des MAJs



Visualisation des MAJs existantes

MAJ

Produits [] Date [] Recherche - F2

Serveur [] Réalisateur []

Chemin_ftp	Date_MAJ	Descriptif_MAJ	Difficulté	Réalisateur	Nom Produit	Serveur FTP
C:\LIVE_UPDATE\PRELOC\20030723.v	24/07/2003	PRELOC LE PREMIER sur lre nouveau serveur 2123	9	SEBASTIEN	PRELOC	Serveur Local/Windev
C:\LIVE_UPDATE\GIPI\20030724-1.wdz	24/07/2003	la seconde "vrai" MAJ de GIPI	1	SEBASTIEN	GIPI	Serveur Local/Windev
C:\LIVE_UPDATE\GIPI\20030725.wdz	25/07/2003	TOTO123456TOTO123456T	9	SEBASTIEN	GIPI	Serveur Local/Windev
C:\LIVE_UPDATE\PRELOC\20030725-1	25/07/2003	kljnlkn kljnlkh lkjh kh	9	JUUU	PRELOC	Serveur Local/Windev
C:\LIVE_UPDATE\PRELOC\20030725-2	25/07/2003	preloc maj base client	9	SEB	PRELOC	Serveur Local/Windev
C:\LIVE_UPDATE\GIPI\20030725-1.wdz	25/07/2003	Dernière version GIPI validée	9	DC	GIPI	Serveur Local/Windev
C:\LIVE_UPDATE\POPULAT\20030725	25/07/2003	popoopopo	9	SEBASTIEN	POPULAT	Serveur Local/Windev
C:\LIVE_UPDATE\CORPUS\20030728.1	28/07/2003	UN teste de sablier	9	SEB	CORPUS	Serveur Local/Windev

Nouvelle MAJ Modification Spécification All Suppression FIN

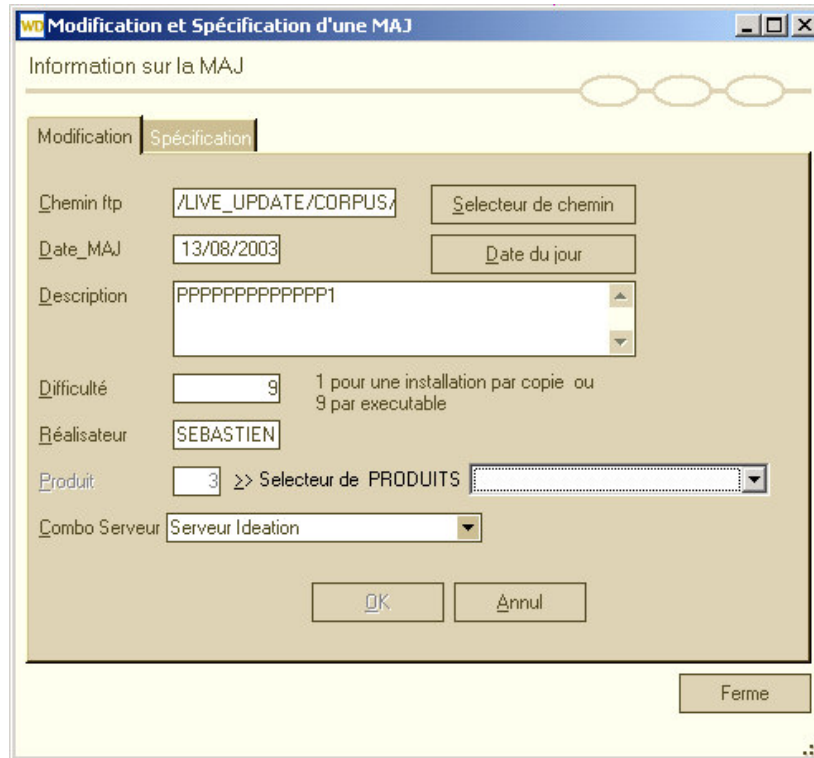
Affichage dans une table

Bouton de gestion des MAJs

Recherche multicritères sur cette table

Quand double clic sur une MAJ

Page de modification de la MAJ (des informations, des fichiers ou encore des spécifications)



Modification et Spécification d'une MAJ

Information sur la MAJ

Modification | Spécification

Chemin ftp: /LIVE_UPDATE/CORPUS/ [Selecteur de chemin]

Date_MAJ: 13/08/2003 [Date du jour]

Description: P P P P P P P P P P P P P P 1

Difficulté: 9 (1 pour une installation par copie ou 9 par executable)

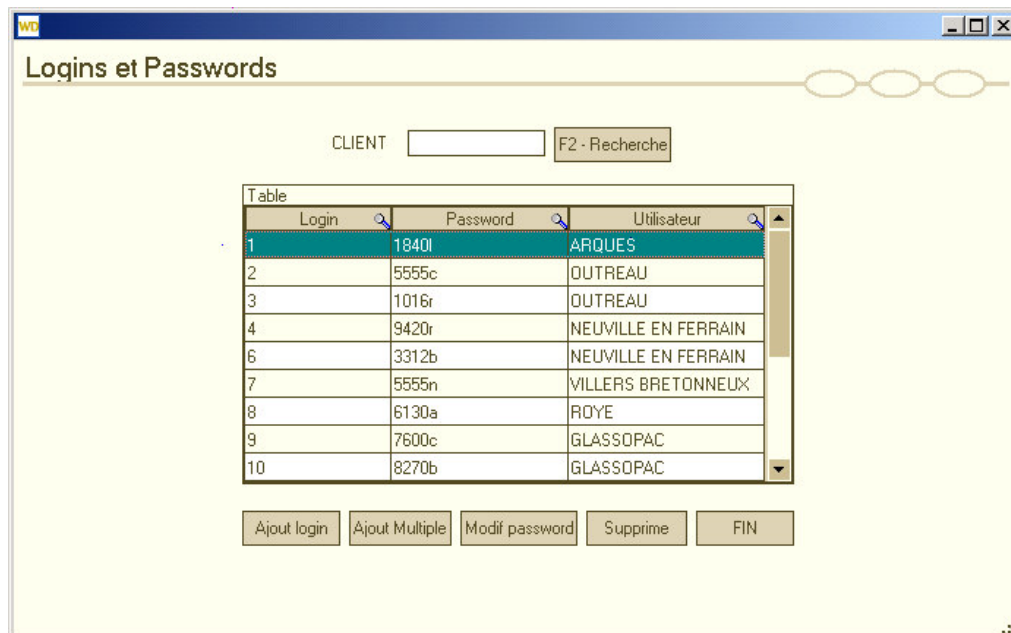
Réalisateur: SEBASTIEN

Produit: 3 >> Selecteur de PRODUITS

Combo Serveur: Serveur Ideation

[OK] [Annul] [Ferme]

Gestion des passwords.



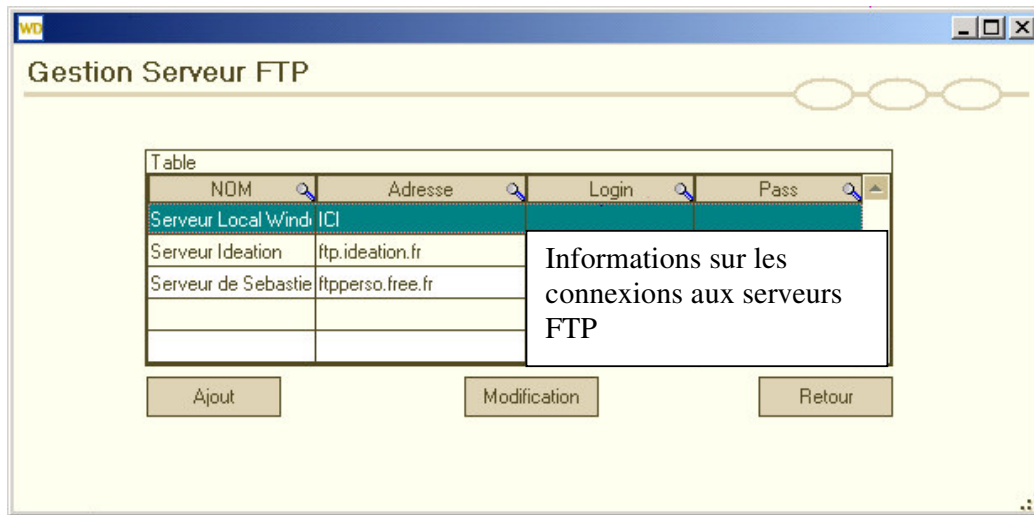
Logins et Passwords

CLIENT [] [F2 - Recherche]

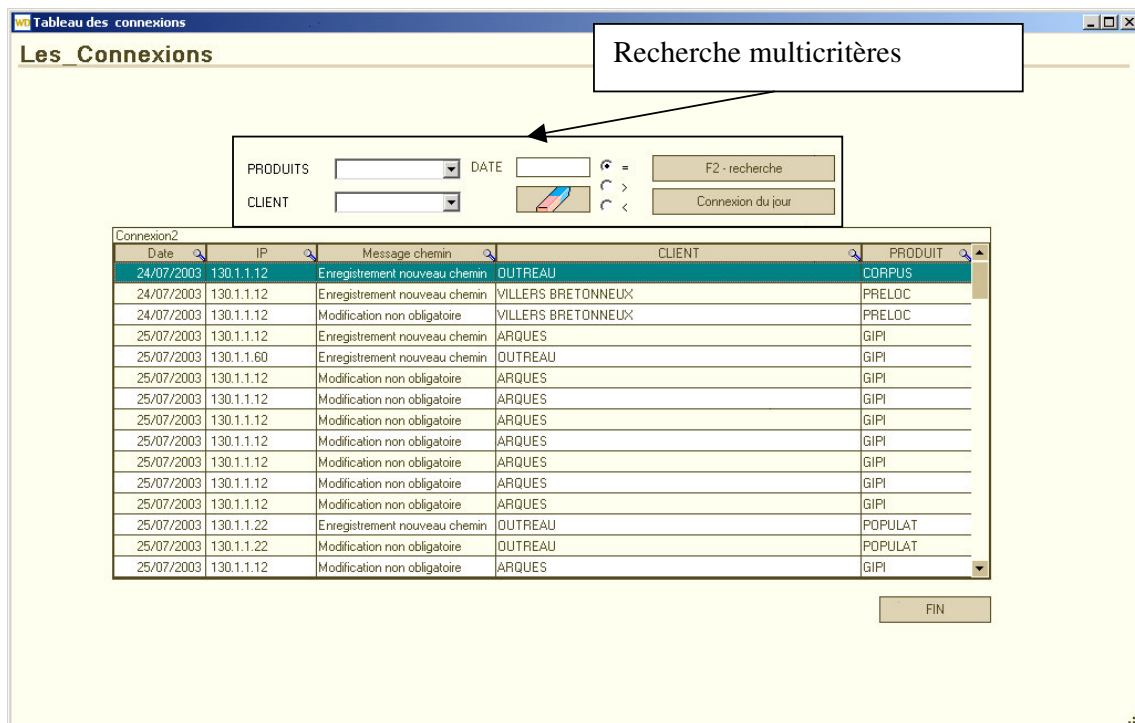
	Login	Password	Utilisateur
1		1840l	ARQUES
2		5555c	OUTREAU
3		1016r	OUTREAU
4		9420r	NEUVILLE EN FERRAIN
6		3312b	NEUVILLE EN FERRAIN
7		5555n	VILLERS BRETONNEUX
8		6130a	ROYE
9		7600c	GLASSOPAC
10		8270b	GLASSOPAC

[Ajout login] [Ajout Multiple] [Modif password] [Supprime] [FIN]

Gestion des serveurs FTP.



Visualisation des connexions



Configuration des connexions.

