

# Solution MSU

## **Langages et technologies utilisées :**

-La solution est développée en WLanguage, langage du logiciel Windev.

## Mission 1

### **Existant :**

Une solution MSU ainsi qu'une base de données HFSQL avec ces jeux de données.

## **Tâche 1 : Automatisation de la suppression d'un enregistrement**

### **Besoins :**

-Dans les fichiers PASFIL et PASPEL des dossiers MSUDEV, MSUTST et MSUTESTWEB, supprimer la ligne de manière répétée lorsque l'enregistrement OK\_server est à TRUE.

-Automatisation de la tâche : supprimer les enregistrements tous les jours à 19h

### **Mode opératoire :**

*J'ai créé un service Windev, dans ce service j'ai créé une procédure avec une connexion à la base de donnée en allant chercher le nom du poste le port et le nom de la base de donnée. Dans cette procédure j'ai d'abord créé une requête SQL de suppression mais lors du débogage, un problème de restriction de droits m'a empêché d'appliquer ces changements à la base de données. J'ai ensuite choisi de faire ces requêtes grâce aux fonctions Windev. J'ai dû pour cela déclarer statiquement les chemins des fichiers de données pour ensuite ouvrir une connexion.*

*Pour l'automatisation de la tâche j'ai créé la procédure d'installation du service Windev, j'ai ensuite créé une tâche grâce au planificateur de tâches Windows, en renseignant le fichier .exe de ma procédure. Pour vérifier j'ai simplement exécuté la tâche manuellement via*

*planificateur de tâches et vérifié dans ma base de données que les modifications ont bien été apportées.*

### **Documentation :**

## **Tâche 2 : Gérer dynamiquement la suppression pour tous les fichiers de données**

### **Contexte :**

*Dans le dossier BDD, il y a N répertoires, chaque répertoire contient des fichiers PASFIL et PASPEL.*

### **Besoins 1 :**

*Il faut pouvoir gérer dynamiquement la suppression des enregistrements de ces fichiers. Pour cela il faut boucler sur les fichiers de chaque répertoire et se positionner sur l'un d'entre eux pour le lire. Dans le cas où il contient des enregistrements, supprimer les enregistrements marqués comme étant traités par une application tierce puis passer au fichier suivant. A la fin d'un sous-répertoire, passer au répertoire suivant.*

*Sachant que d'autres répertoires peuvent être ajoutés par la suite.*

### **Mode opératoire :**

*J'ai utilisé une variable de WinDev pour récupérer tous les fichiers PASPEL et PASFIL en parcourant les répertoires et sous-répertoires d'un fichier racine. J'ai ensuite bouclé sur chaque fichier pour déclarer une connexion au fichier de données, effectuer la suppression et fermer la connexion à la base de donnée afin de passer au fichier suivant.*

*J'ai redéployé le service avec le planificateur de tâche afin d'exécuter le nouveau programme. J'ai ensuite testé que l'appel de la tâche fonctionne correctement avec un jeu de données enregistré au préalable dans un fichier temporaire pour pouvoir réutiliser les données.*

### **Besoins 2 :**

*Externaliser le répertoire pour pouvoir le déployer sur n'importe quel poste, et pas seulement en local avec le chemin en dur.*

### **Mode opératoire :**

*Choix d'utilisation d'un registre Windows REGEDIT plutôt que des variables d'environnements ou un fichier INI pour des raisons de sécurité. J'ai donc créé un registre dans LOCAL\_MACHINE\SOFTWARE\MSU avec comme clé « PATH » et comme valeur mon chemin C:\Mes Projets\Serveur HFSQL\BDD\.*  
*J'ai dû demander les droits au support informatique car il fallait une élévation de privilèges.*  
*Dans mon service j'ai déclaré un booléen qui vérifie que le registre existe bien, ensuite j'ai déclaré une variable récupérant le registre. J'ai aussi créé une condition pour que le service puisse écrire dans le journal Windows en cas d'erreur.*

### **Besoins 3 :**

*Créer un fichier de log pour vérifier que les fichiers soient bien copiés tous les jours + Créer un README.*

### **Mode opératoire :**

*J'ai créé un fichier texte que j'ai placé dans le dossier « .exe » où se trouve le déploiement de mon service. J'ai ajouté dans le code de mon exécutable des traitements permettant de supprimer tous les jours le contenu du fichier pour pouvoir avoir seulement le contenu ne datant pas de plus de 24h. J'ai aussi ajouté un fichier texte README dans le dossier où se trouve le service pour expliquer le programme ainsi que comment le déployer.*

## **ReadMe**

### **Programme de suppression automatique des enregistrements de MSU**

- Le programme permet de supprimer quotidiennement (à 19h), les enregistrements des fichiers PASFIL et PASPEL de tous les répertoires.
- Un journal de log est mis à disposition avec un remplissage quotidien, les données sont disponibles pour une période de 24 h.

Procédure à suivre:

Pour répliquer le programme, configurer la base de registre Windows en ajoutant une clé nommée "PATH" avec comme valeur le chemin de la base de données.

Pour planifier la tâche, ouvrez le planificateur de tâches Windows, cliquez sur "Créer une tâche de base", donnez-lui un nom, choisissez "tous les jours", puis dans "Démarrer le" mettez 19:00:00 pour l'heure.

Cliquez sur Suivant -> Démarrer un programme -> parcourir et sélectionnez le fichier "Suppression\_Projet.exe". Cliquez sur Suivant-> choisissez un nom -> Terminer.

Pour accéder au journal, il faut aller dans le dossier EXE et ouvrir « Journal\_MSU.txt »

## Mission 2

### **Besoin 1 :**

*Créer un projet permettant de gérer une base de paramètre, avec les fonctions d'ajout, de modification et de suppression*

### **Mode opératoire :**

*J'ai créé une fenêtre dans laquelle j'ai mis un tableau, un bouton supprimer, modifier et ajouter. J'ai créé une analyse avec une base de donnée nommée PARAMETRES, dans cette base j'ai ajouté tous les enregistrements d'une colonne d'un tableau Excel grâce à une procédure que j'ai créée avec WinDev. Ensuite j'ai bouclé pour remplir mon tableau de la fenêtre à partir des données de ma table. J'ai ensuite configuré le bouton supprimer qui supprime l'enregistrement sélectionné. Pour ce qui est de la modification et de l'ajout j'ai créé une nouvelle fenêtre avec un champ dans lequel rentrer le paramètre et un bouton valider pour soumettre la modification ou l'ajout.*

*Existant : un dossier MSU contenant deux dossiers MSUTEST et MSUDEV contenant des fichiers de paramètres.*

## **Besoin 2 :**

*Pouvoir visualiser les différences entre la table de paramètres de chaque instance (MSUTST et MSUDEV) et la table de paramètre créé dans le projet précédant.*

## **Mode opératoire :**

*Création d'une fenêtre avec deux tables. J'ai ajouté une combo box qui permettra de passer d'une instance à l'autre, ce qui changera la table qui lui est reliée. J'ai créé une fonction permettant d'afficher dans la table de gauche les enregistrements d'un fichier de donnée. Ensuite il fallait que le combo box permette de passer d'une instance à l'autre et donc d'afficher les données correspondantes dans la table : j'ai donc fait une clé étrangère entre la table reliée au combo box et la table. J'ai chargé la table de droite grâce à la procédure du Besoin 1 et j'ai mis en place un système de couleur permettant de mettre en évidence les paramètres qui ne sont pas commun aux deux tables. J'ai ensuite compris qu'on pouvait éviter de passer par le remplissage d'une table d'une analyse en récupérant directement les enregistrements des fichiers. Chaque fichier est affiché dans la table en fonction de la liste du combo sélectionnée. J'ai ensuite effectué une série de test et j'ai remarqué que les paramètres de la table de droite manquant dans la table de gauche n'étaient pas correct lors du changement de remplissage de la table de gauche. Pour régler le problème j'ai mis en place le rafraichissement de la table de droite à chaque Combo sélectionné.*

*J'ai créé la fonction permettant de supprimer un enregistrement de table ainsi que la fonction permettant d'ajouter un enregistrement d'une table à l'autre.*

## **Mission 3**

### **Besoin :**

*Fenêtre permettant de lier 3 fichiers. Une table doit afficher les groupes de paramètres, l'application doit permettre d'afficher tous les paramètres liés à un groupe de paramètre, ensuite cette liste de paramètre doit être comparé avec tous les autres paramètres.*

### **Mode opératoire :**

*J'ai commencé par ajouter à ma fenêtre un combo box permettant de naviguer entre les 2 instances. J'ai ajouté en dessous 3 tables nommées respectivement « TablePRF », « TablePPP » et « TablePRM ». J'ai créé une procédure permettant d'ajouter toutes les valeurs du fichier PRF dans « TablePRF » lorsque qu'une instance de la combo box est sélectionnée. J'ai ensuite placé un bouton « ajouter » permettant de récupérer l'indice sélectionné de TablePRF et remplir TablePPP avec les valeurs du fichier PPP dont l'id est égal à l'id de l'enregistrement sélectionné de TablePRF. J'ai créé une procédure permettant de remplir TablePRM avec les valeurs du fichier PRM dont l'id correspond à celui de chaque ligne de TablePPP. J'ai dupliqué ces tables pour pouvoir afficher grace a un deuxième Combo une autre instance, permettant de comparer deux instances.*

### **Besoin 2 :**

*Créer une fenêtre MDI permettant d'ouvrir la fenêtre des paramètres mais aussi la fenêtre des groupes de paramètres.*

### **Mode opératoire :**

*Création d'une nouvelle fenêtre de type MDI, sélection de cette fenêtre pour l'ouverture par défaut. J'ai créé une liste déroulante avec le nom de mes 3 fenêtres, lorsque l'on sélectionne une ligne, la fenêtre s'ouvre en plein écran juste en dessous. Pour que les fenêtres filles soient liées à la fenêtre mère, j'ai modifié le type des 3 fenêtres en « Fenêtre Fille MDI ». J'ai ensuite apporté quelques modifications graphiques pour réorganiser les objets grâce aux ancrages.*

### **Amélioration du code :**

*Modification de certaines boucles pour que le temps traitement soit moins long.*

### **Création de l'exécutable**

