

TP WinDev Numéro 3



Objectifs : Travailler avec un fichier de données :

Etude du gestionnaire d'analyse,
 Manipulation des tables mémoires,
 Manipulation de données,
 Création d'états,
 ...

Pré requis : TP WinDev Numéro 1 et 2

L'objectif de cette leçon est de vous familiariser avec l'utilisation des fichiers, du gestionnaire d'analyse et de la conception d'états.

Vous allez commencer par créer un nouveau projet nommé TP3. Dans l'assistant vous confirmerez le choix de création d'une nouvelle analyse, sélectionnez votre thème préféré et finirez le processus de l'assistant. Un nouvel assistant de création d'analyse va apparaître, cliquez sur **Suivant**.

Vous pouvez choisir entre un Modèle Logique de Données et un Modèle Conceptuel de Données. Cliquez sur **Suivant** pour choisir le MLD.

Dans la zone Nom de l'analyse appelez là : **TP3** et vérifiez qu'elle soit bien associée au projet en cours cliquez ensuite sur **Suivant**.

Nous ne donnerons pas de mots de passe à l'analyse donc cliquez encore sur **Suivant**.

Cette nouvelle fenêtre vous demande de choisir le type de base de données que vous voulez "attaquer". C'est ici par exemple que vous pourriez décider d'utiliser une base de données tierce du type Mysql, Access ou autres. Comme le but de cette leçon est de travailler avec WinDev nous allons faire en sorte de **Créer une nouvelle description de fichier**. Ce choix étant validé nous allons maintenant créer notre structure de fichier.

Ah ! oui, j'ai oublié de vous dire ce que nous allons faire... En fait nous allons programmer une mini gestion de budget familial. Pour cela on va utiliser un seul fichier des mouvements dans lequel on inscrira le descriptif des opérations, la date de l'opération, le montant au débit ou le montant au crédit. Cela devrait vous faire penser à votre relevé de compte bancaire.

Dans la zone Nom, vous allez indiquer le nom suivant : mouvement. Remarquez les champs suivant qui se remplissent automatiquement. Vérifiez que la case « Le fichier possède un identifiant automatique » soit cochée. Nous allons ainsi créer un identifiant automatique par WinDev. L'identifiant automatique est comparable à un compteur, c'est lui qui vous garantit l'unicité de vos tuples. Je vous renvoie au cours d'analyse Merise pour les compléments d'informations que vous pourriez avoir sur le rôle des identifiants !

Cliquez sur **Suivant**. La nouvelle fenêtre vous demande de confirmer le fait que vous voulez travailler avec des fichiers de type Hyper file (Format propriétaire WinDev). Cliquez sur **Suivant**. La nouvelle fenêtre vous propose des options RAD (Rapid Application Development). Le RAD est à oublier, c'est WinDev qui fait tout, vous n'apprendrez rien à le voir faire, il vaut mieux que ça soit vous qui créez que lui. Donc décochez ces cases, cliquez sur **Suivant** et enfin sur **Terminer**.

Vous voilà maintenant dans le gestionnaire d'analyse, vous vous trouvez plus exactement dans la fenêtre de description des fichiers. C'est ici que nous allons déterminer la composition de chaque attribut de notre fichier mouvement.

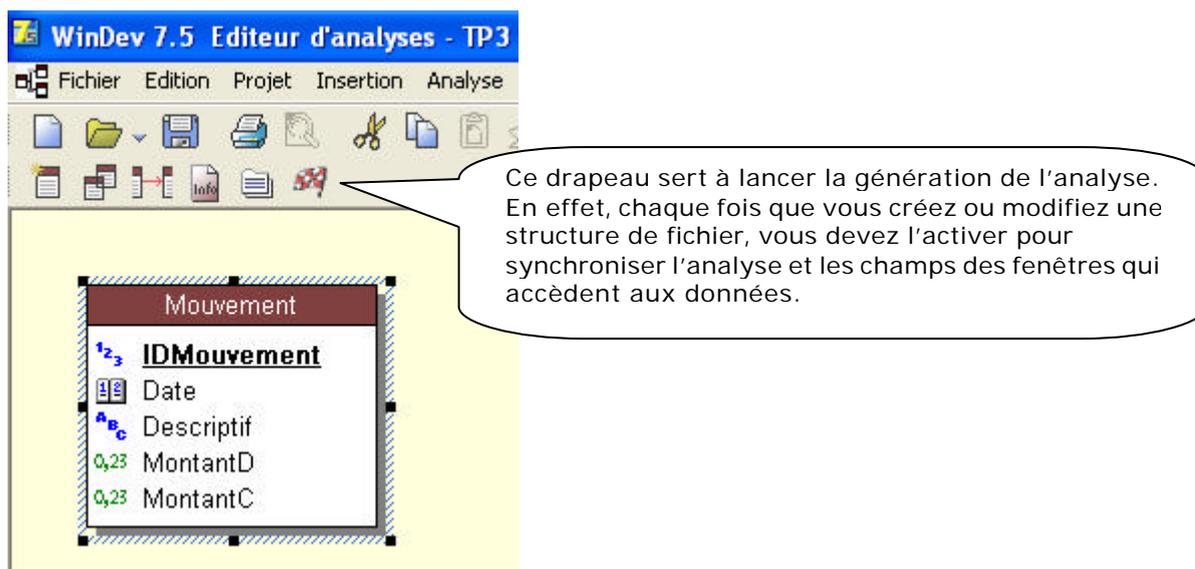
Vous pouvez remarquer que notre identifiant est déjà créé « Idmouvement » et vous voyez la clé jaune à gauche qui symbolise l'identifiant. Nous allons insérer les rubriques suivantes :

Nom	Libellé	Type	Taille
Date	Date de l'opération	Date	Sera rempli automatiquement
Descriptif	Description de l'opération	Texte	50
MontantD	Montant Débit	Numérique (Format ?999 999,99)	Sera rempli automatiquement
MontantC	Montant Crédit	Idem	Idem

Pour insérer, cliquez sur une ligne vide dans la rubrique choisie. A vous de bien remplir vos champs, cela ne devrait vous poser aucun problème. Une fois la saisie terminée, cliquez sur **Valider** (si le bouton est actif) et ensuite sur **Fermer**.

Vous pouvez maintenant **Retourner sous l'éditeur de WinDev**. Choisissez ensuite **Aller directement dans l'éditeur WinDev** et **OK**.

Voici une représentation de ce que vous devriez obtenir



Maintenant que votre fichier est décrit, il ne vous reste plus qu'à générer l'analyse.

Nous allons maintenant fabriquer les fenêtres de notre application. Allez dans le menu **Fichier / Nouveau** et cliquez sur **Fenêtre**. Choisissez une fenêtre vierge.

Cette fenêtre sera la première fenêtre de notre application. Dans sa description (clic droit sur la fenêtre), vous lui donnerez les caractéristiques suivantes :

Nom logique : **départ**

Description : **Fenêtre principale de l'application**

Titre : **Bienvenue dans votre mini compte bancaire**

Appliquez les modifications et cliquez sur **Ok**.

Nous allons insérer dans cette fenêtre une table mémoire qui sera le conteneur du fichier. Choisissez un champ table dans la barre d'outils et positionnez le sur votre fenêtre. Dites à l'assistant que vous remplirez la table vous-même par programmation, cliquez sur **Suivant**.

Choisissez « type de table en affichage » puis **Terminer**. La table est définie par défaut, allez dans la **Description** (clic droit sur la table).

Le nom de la table : Tmouv

Nom de la colonne 1 : tdate

Type de la colonne 1 : Date

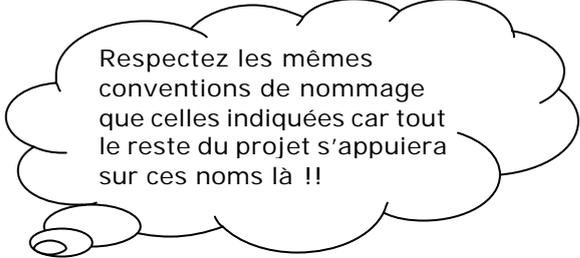
Titre de la colonne 1 : Date

Nom de la colonne 2 : tdescription

Type de la colonne 2 : Texte

Titre de la colonne 2 : Description de l'opération

Taille de saisie : 50



Nom de la colonne 3 : tdébit
Type de la colonne 3 : Numérique
Titre de la colonne 3 : Débit

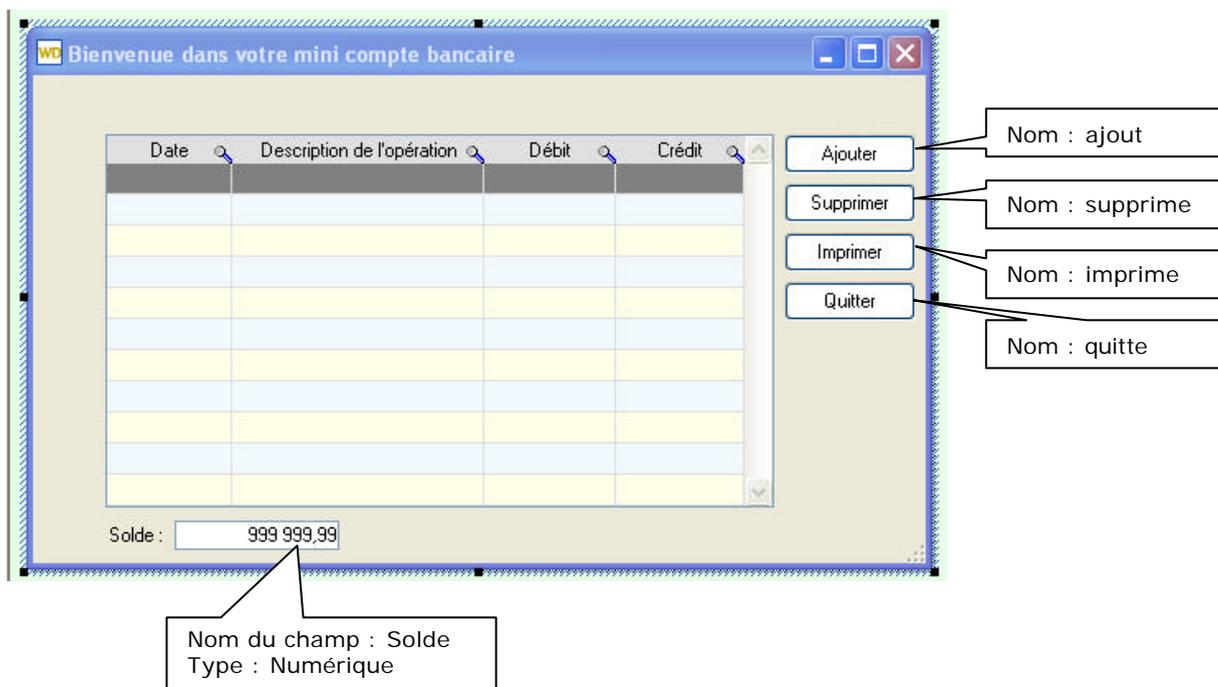
Nom de la colonne 4 : tcrédit
Type de la colonne 4 : Numérique
Titre de la colonne 4 : Crédit

Nom de la colonne 5 : clé
Type de la colonne 5 : Numérique
Masque de saisie : 999 999 999.

Dans l'onglet **Détail** de cette 5^{ème} colonne, décochez **Visible**, nous mettrons dans ce champ l'identifiant de la ligne. Il n'est pas nécessaire de le montrer à l'utilisateur, c'est pour ça que je vous conseille de le mettre invisible.

Une fois tous les champs renseignés cliquez sur **Appliquer** puis **OK**

Voici ce qu'il vous reste à concevoir pour terminer notre projet :



 Il me semble que ça fait longtemps que vous n'avez pas sauvegardé votre projet ! ce n'est pas très prudent... N'oubliez pas de déterminer cette fenêtre comme la première fenêtre du projet

Maintenant que le décor est planté nous pouvons commencer la programmation.

La première chose à faire est de dire à WinDev de nous créer le fichier mouvement s'il n'existe pas, pour cela allez dans le menu Projet / Code du projet et dans la zone initialisation inscrivez le code suivant :

```
HCréationSiInexistant(Mouvement)
```



Remarque : Cette ligne indique à WinDev de commencer à chercher si le fichier mouvement existe, s'il ne le trouve pas il le crée. Le code placé dans cette zone est exécuté avant le chargement de la première fenêtre.

Le code du bouton Quitter est très facile : Dans clic sur **Quitter** inscrivez :

```
Ferme
```

Voyons maintenant la décomposition possible des événements. Il faut qu'au chargement de la fenêtre, la table se remplisse avec les enregistrements contenus dans le fichier situé sur le disque dur. Pour cela nous allons donc parcourir l'ensemble des lignes du fichier « mouvement.fic » et les placer les unes après les autres dans la table mémoire. C'est ce que nous allons faire maintenant.

Allez dans le code de la fenêtre dans la zone « initialisation de départ » et saisissez le code suivant :

```
HLitPremier(Mouvement, IDMouvement)
TANTQUE PAS HEnDehors
    TableAjoute(Tmouv, Mouvement.Date+TAB+Mouvement.Descriptif+TAB+Mouvement.Montantd+TAB+Mouvement.Montantc+TAB+Mouvement.IDMouvement)
    HLitSuivant(Mouvement)
FIN
```

Explication du code :

HLitPremier(Mouvement, IDMouvement) // Cette ligne ordonne à WinDev d'ouvrir le fichier Mouvement et de lire la première ligne en plaçant les champs correspondants en mémoire.

TANTQUE PAS HEnDehors // Ici on commence une boucle qui sera vraie tant que l'on reste dans le fichier. Le mot clé HEndehors renvoie Vrai si on est hors du fichier. Donc PAS HEnDehors est vrai tant que la fin du fichier n'est pas atteinte. C'est tout simple en fait, non ?

La séquence tableajoute est connue elle permet de positionner du texte, ici les rubriques du fichier, à l'intérieur d'un fichier.

Hlitsuivant(Mouvement) // Fait descendre le pointeur du fichier d'une ligne vers le bas.

Le bouton **Ajouter** va ouvrir une fenêtre de saisie qui nous permettra de rentrer les informations. Donc le code sera dans Clic sur **ajout** :

Ouvre(saisie)

Il est normal que WinDev vous renvoie un message d'erreur si nous essayons d'exécuter le projet car la fenêtre saisie n'existe pas, créons-la de suite.

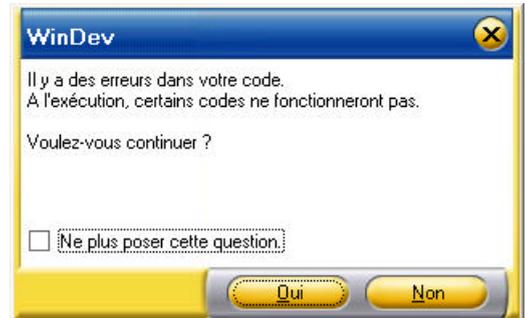
Fichier / Nouveau / Fenêtre

Dans la description de la fenêtre :

Nom logique : saisie

Description : Fenêtre de saisie

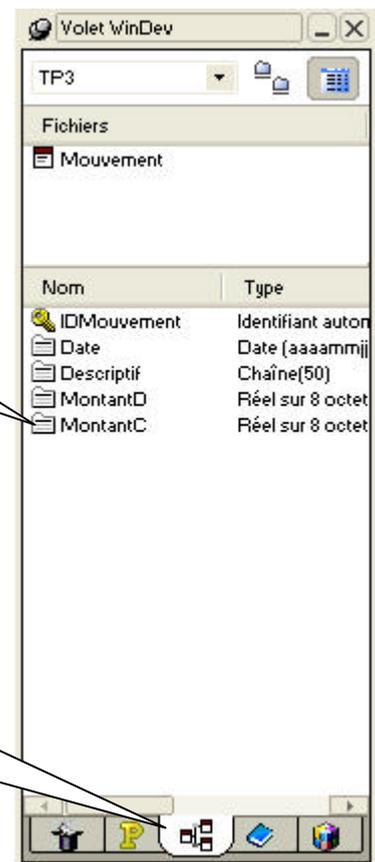
Titre : Saisissez votre opération



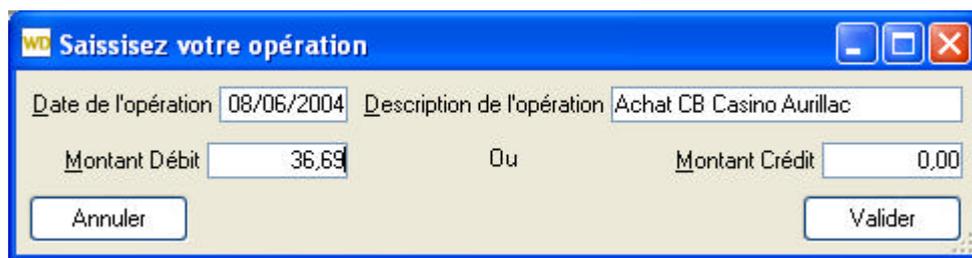
Faites glisser les champs **Date**, **Descriptif**, **MontantD**, **MontantC** sur la fenêtre **saisie**.

Cliquez sur un champ en tenant le bouton gauche de la souris enfoncé, faite glisser et déposez sur la fenêtre. N'hésitez pas à changer la mise en forme du champ. Le fait de faire glisser le champ vous garanti que WinDev peut faire l'association entre le champ sur la fenêtre et la rubrique du fichier.

Cet onglet vous permet de sélectionner les fichiers constituant votre analyse. Je vous laisse chercher à quoi correspondent les autres.



Votre fenêtre devrait correspondre à celle-ci :



Pour être sûr que chaque champ est bien lié à une rubrique du fichier, cliquez sur l'un d'entre eux et vérifiez dans la barre de message en bas à gauche que le message « Lié à : Mouvement.XXXXXX » soit présent.



Cette fenêtre ne comporte que 2 boutons, le bouton **Annuler** nous servira juste à fermer la fenêtre. Vous connaissez la séquence de code le permettant, ce n'est donc plus la peine que je vous indique la marche à suivre. En cas de problème je vous rappelle que l'aide de WinDev est accessible par la touche **F1**.

Consacrons-nous au bouton **Valider**, la validation consiste à placer les rubriques de la fenêtre dans le fichier et à valider l'ajout. Pour placer les rubriques de la fenêtre dans le fichier, l'ordre est le suivant :

```
EcranVersFichier(saisie) // saisie étant le nom de la fenêtre
```

La validation d'ajout est commandée par l'ordre suivant :

```
HAjoute(Mouvement) // Mouvement étant le fichier dans lequel on ajoute
```

Il nous reste plus qu'à rajouter un ordre de fermeture de la fenêtre.

Voici le code intégral du bouton **Valider** :

```
EcranVersFichier(saisie)
HAjoute(Mouvement)
Ferme
```

Vous pouvez faire en sorte de programmer le contrôle pour qu'il vérifie que l'on n'a qu'un débit ou qu'un crédit, ou alors que l'un des champs n'est pas vide. Le code à placer juste avant celui que l'on voit ci-dessus pourrait ressembler à ceci :

```
SI Date="" OU Descriptif="" OU (MontantD=0 ET MontantC=0) ALORS
    Info("L'un des champs n'a pas été rempli !")
    Ferme
FIN
SI MontantD<>0 ET MontantC<>0 ALORS
    Info("Vous ne pouvez saisir qu'un montant au Débit ou alors au Crédit !")
    Ferme
FIN
```

Testez votre fenêtre avec le bouton  ou en cliquant à droite de l'écran sur la fenêtre avec le bouton droit pour choisir **Tester**. Insérez des valeurs dans les champs et validez.

Pour voir si votre nouvelle ligne est présente dans le fichier, allez dans le menu **Outils / WDMAP** et choisissez **Mouvement** comme **Nom du fichier**.

Maintenant que vos lignes s'insèrent, lancez le projet en cliquant sur GO .

Cliquez sur le bouton **Ajouter**, saisissez et validez un nouvel enregistrement. Vous pouvez constater que la table mémoire ne réagit pas correctement : en effet l'insertion n'a pas été détectée et donc la table mémoire n'est pas synchronisée avec le fichier. Nous allons essayer de remédier à ce problème. En fait, il faudrait que lorsque la fenêtre **saisie** se ferme la fenêtre **départ** recharge la table mémoire.

Placez-vous dans le code de la fenêtre **départ**, vous devez trouver une zone nommée « **prise de focus de départ** ». La prise de focus est le fait de remettre active une fenêtre inactive, en cliquant sur la barre de titre par exemple.

Voici le code à insérer dans cette zone :

```
TableSupprimeTout(Tmouv) // Efface la table mémoire pour éviter d'insérer les
enregistrements à la suite des précédents
HLitPremier(Mouvement, IDMouvement)
TANTQUE PAS HEnDehors
    TableAjoute(Tmouv, Mouvement.Date+TAB+Mouvement.Descriptif+TAB+Mouve
ment.Montantd+TAB+Mouvement.Montantc+TAB+Mouvement.IDMouvement)
    HLitSuivant(Mouvement)
FIN
```

Testez cette modification, comme vous le voyez, les comportements sont maintenant cohérents.

Intéressons-nous au bouton **Supprimer**. Dans la table, nous avons une rubrique qui est l'identifiant de la ligne. Pour supprimer cette ligne dans le fichier nous allons donc rechercher cet identifiant dans le fichier et supprimer la ligne correspondante.

Voici la séquence de code nécessaire :

```
HLitRecherche(Mouvement, IDMouvement, clé)
SI H.trouve ALORS
    HSupprime(Mouvement)
    Info("La suppression est effective")
SINON
    Info("Grave problème de l'application")
FIN
```

Explications :

La première ligne fait rechercher dans le fichier Mouvement et sur l'identifiant la valeur du champ **clé** pointée dans notre table mémoire. Ce n'est pas parce que la rubrique **clé** est invisible que nous ne pouvons pas en connaître la valeur.

Si on trouve la ligne ayant le même identifiant que **clé** alors on la supprime et on affiche un message indiquant la bonne marche des opérations. Sinon dans un cas fort improbable où il ne trouve pas l'enregistrement on inscrit un message d'erreur.

Je vous laisse le soin de tester cette nouvelle fonctionnalité de votre programme. Comme vous venez de le remarquer, la mise à jour de la table mémoire ne s'est pas faite. La valeur a été supprimée mais la table ne le sait pas. Nous allons remédier à ce problème.

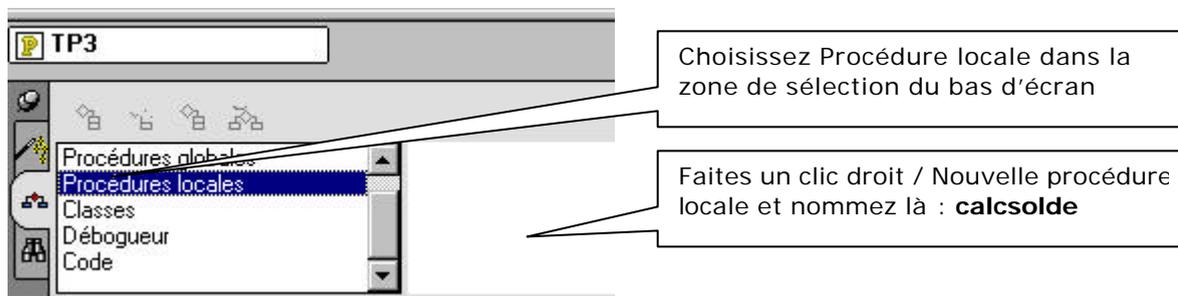
Le code de réaffichage de la table mémoire existe déjà (ex : dans la zone de **prise de focus** de la fenêtre **départ**) nous allons donc ré-exécuter un traitement existant. Sous la ligne « Info("La suppression est effective") » inscrivez la commande suivante :

```
ExécuteTraitement(depart, trtPriseFocus)
```

Cette commande fait rejouer un traitement existant.

Testez et vérifiez la cohérence de votre projet.

Maintenant **nous avons à calculer le solde** (Débit – Crédit), pour ce faire nous allons créer une procédure qui scannerera le fichier et fera les calculs pour nous.



Remarques : Les procédures locales ne sont vues que par les objets de la fenêtre, les procédures globales sont actives pour tous les éléments du projet

Maintenant, vérifiez bien que vous vous trouvez dans la zone **code** de la procédure **calcsolde**.

Le code doit parcourir le fichier Mouvement, affecter le contenu de **débit** dans une variable, le contenu de **crédit** dans une autre et cela jusqu'à la fin du fichier et ensuite affecter la différence entre le débit et le crédit au champ **solde**.

Voici le code de la procédure :

```

PROCEDURE calcsolde()
sdebit,scredit sont des réels=0 // on affecte la valeur 0 aux deux variables

HLitPremier(Mouvement, IDMouvement)
TANTQUE PAS H.endehors
    Sdebit +=Mouvement.Montantd // += signifie
Sdebit=Sdebit+Mouvement.Montantd
    Scredit +=Mouvement.Montantc
    HLitSuivant(Mouvement, IDMouvement)
FIN

solde=sdebit-scredit
    
```

Le code est suffisamment simple pour ne pas avoir à l'expliquer. La question qui se pose est où lancer **calcsolde**. Les plus rusés d'entre vous auront compris qu'il nous faut activer cette procédure aux mêmes endroits ou l'on a activé le rafraîchissement de la table mémoire. Je vous laisse modifier les zones de code en conséquence (dans le **code** de la fenêtre **départ**). N'oubliez pas de tester la cohérence de votre projet.

Il ne nous reste plus qu'à fabriquer l'état de sortie. Choisissez **Fichier / Nouveau / Etat**. Choisissez un état de type **Tableau**. En cliquant sur **Suivant** WinDev vous demande la source de donnée, il vous faut préciser que ce sont des enregistrements provenant d'un fichier Hyper file. Sélectionnez le fichier **Mouvement**, la clé de parcours est un identifiant qui sert pour donner l'ordre de tri. Continuez à appuyer sur **Suivant**, je vous laisse découvrir les questions posées et à vous de prendre les bonnes décisions. Vous savez lire, donc vous prendrez les bonnes décisions ! Je vous demande juste d'appeler l'état **Etatmouv**.

	Remarques : Les détails du choix du style, de la mise en forme sont à votre discrétion, faites comme bon vous semble. N'oubliez pas que le client n'a pas les mêmes goûts graphiques que vous, donc faites sobre. Eviter les styles « Noir sur fond Noir » « Rose sur fond Vert » et autres singularités visuelles qui feront penser à l'utilisateur qu'il devient déficient visuel.
---	--

Une fois votre état fini et enregistré, entrez dans la zone **code** du bouton **Imprime** :

```

iAperçu(i100) // On enverra l'état à l'écran avec un zoom de 100 %
iImprimeEtat(Etatmouv) // Génération de l'état et utilisation des paramètres définis par iAperçu
    
```

	Remarques : Les Commandes WinDev sont classées. Celles qui commencent par H sont des commandes d'accès aux fichiers, celles qui commencent par i sont des commandes de pilotage d'état.
---	---

	N'oubliez pas de refaire ce TP plusieurs fois , le but étant de se passer du guide papier et d'apprécier la facilité avec laquelle on peut travailler avec WinDev.
---	---

Exercices applicatifs

Répertoire d'amis

Énoncé :

Ecrire un programme gérant un fichier des amis. Pour chacun d'eux, on souhaite connaître le nom, le prénom et le numéro de Téléphone.

Faites en sorte que le programme puisse permettre à l'utilisateur de :

- ☒☒ Saisir un nouvel ami.
- ☒☒ Supprimer l'ami qui vous a fait une embrouille.
- ☒☒ Modifier le numéro de téléphone d'un ami.
- ☒☒ Imprimer la liste des amis.
- ☒☒ Afficher le nombre d'amis.
- ☒☒ Quitter.

La Vidéothèque

Énoncé :

Un proche vous demande de lui concevoir un programme d'archivage de ses cassettes vidéo. Il veut connaître le titre du film, l'année de sortie, l'acteur masculin principal, l'actrice féminine principale et le genre du film (comédie, dramatique, policier, western, enfants, adultes....) Il vous demande aussi des états imprimés triés soit par année de sortie, soit par genre, soit avec aucun tri.

Comme vous êtes un développeur consciencieux, vous allez faire un programme qui apporte toutes satisfactions à votre ami.