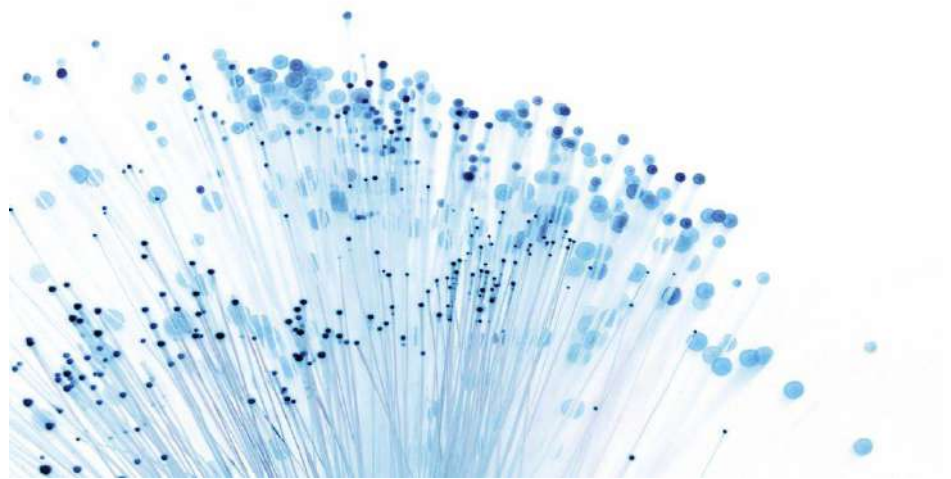


COURS WINDEV NUMERO 3



01/02/2015

Travailler avec un fichier de données

Etude du gestionnaire d'analyse,

Manipulation des tables mémoires,

Manipulation de données,

Création d'états,

...

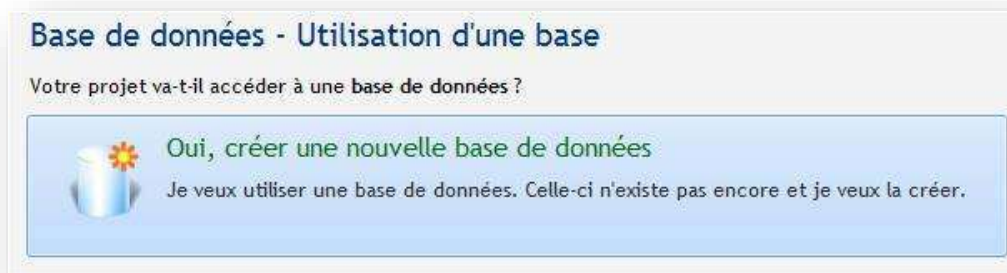
Pré requis : Cours WinDev Numéro 1 et 2

Cours WinDev Numéro 3

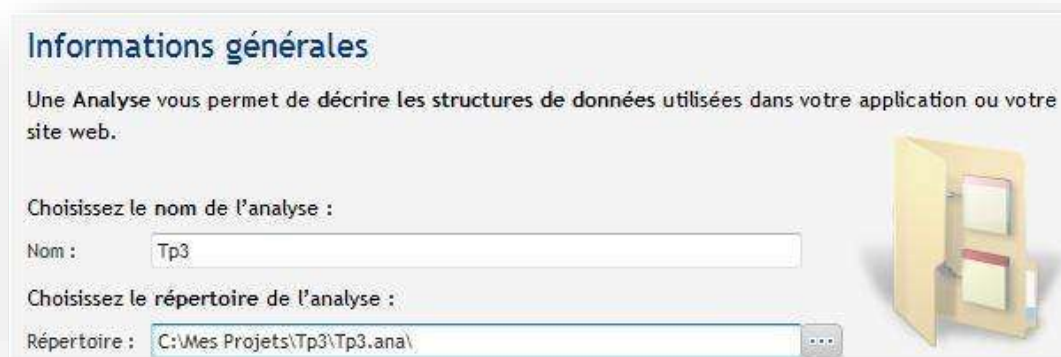
VERSION 19

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 sélectionnez votre thème préféré, confirmez les choix par défauts jusqu'à ce point :



Ici, il est impératif de sélectionner le choix « **Oui, créer une nouvelle base de données** ». Cliquez sur **Suivant**.



Par défaut, les informations sont correctes, vous pouvez cliquer sur suivant.



Comme vous pouvez le constater, le choix des types de bases de données est vaste ! Nous allons travailler avec le format natif de WinDev HyperFileSQL Classic. Nous pouvons passer à l'écran suivant.

Vous avez décrit les informations nécessaires à la création de l'analyse.
Cliquez sur le bouton de validation pour créer l'analyse.

Voilà, la description de l'analyse est finie !

Un assistant de création de fichier va prendre le relais :

Mode de création du fichier de données



Créer une nouvelle description d'un fichier de données

Vous choisissez le nom et le type du fichier de données à créer.



Sélectionner une description parmi des fichiers de données prédéfinis

Vous choisissez parmi plusieurs thèmes qui proposent des fichiers de données prédéfinis que vous pourrez modifier par la suite.



Utiliser des fichiers de données d'une base existante

Vous pouvez importer des descriptions de fichiers de données depuis différents formats (HFSQL Client/Serveur, Oracle, AS/400, etc.) avec ou sans leurs données.

Ici, nous laissons le choix par défaut.

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.

Voici la fenêtre suivante

Paramètres généraux

Nom du fichier de données _____

Nom :

Libellé :

Un enregistrement représente :

Identifiant automatique _____

Identifiant automatique sur 8 octets (obligatoire si vous faites de la réplication serveur)

Identifiant automatique sur 4 octets

Aucun

Dans la zone **Nom**, vous allez indiquer le nom suivant : **Mouvement**. Remarquez les champs suivants qui se remplissent automatiquement. Vérifiez que la case « Le fichier possède un identifiant automatique » soit cochée. WinDev va ainsi créer un identifiant automatique. L'identifiant automatique est comparable à un compteur, c'est lui qui vous garantit l'unicité de vos enregistrements. Je vous renvoie au cours d'analyse Merise pour les compléments d'informations que vous pourriez avoir sur le rôle des identifiants ! (A ce sujet, j'ai écrit un bon livre la dessus – Voir chez l'éditeur ENI)

On clique sur Suivant ?



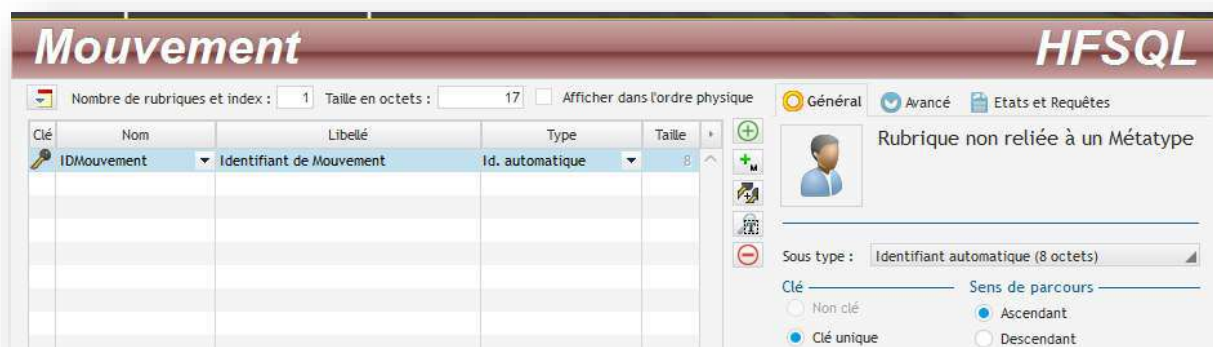
La nouvelle fenêtre vous demande de confirmer le fait que vous voulez travailler avec des fichiers de type Hyper file (Format propriétaire WinDev).

C'est terminé !

Vous avez décrit les informations nécessaires à la création du fichier de données.

Cliquez sur le bouton de validation pour créer le fichier dans l'analyse.

Et voilà, la description du type de fichier est finie. Cliquez sur le bouton vert pour terminer. L'assistant se ferme et vous ouvre la fiche de description du fichier.



C'est ici que nous allons déterminer la composition de chaque attribut de notre fichier mouvement.

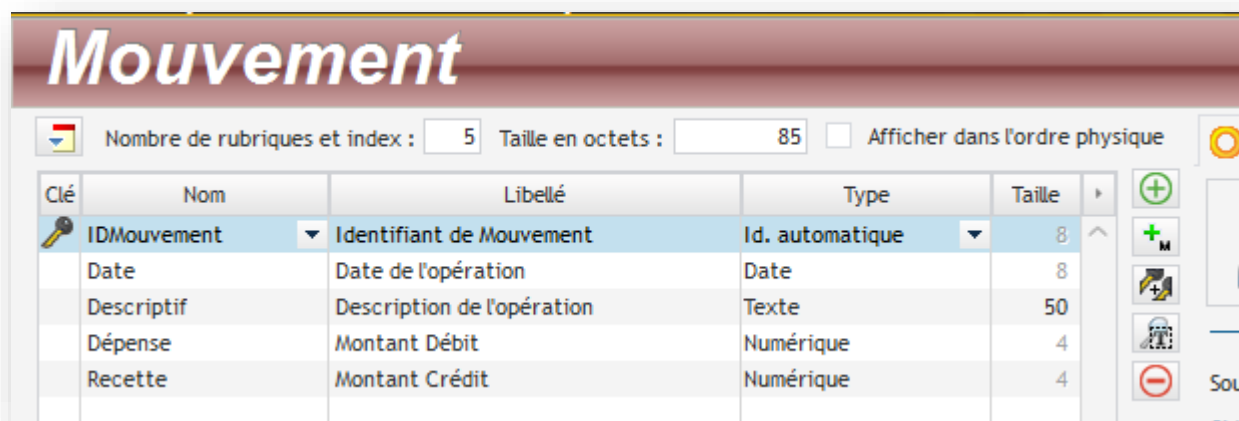
Vous pouvez remarquer que notre identifiant est déjà créé « Idmouvement » et vous voyez la clé jaune à gauche qui symbolise l'identifiant. Nous allons insérer les rubriques suivantes :

Nom	Libellé	Type	Taille
Date	Date de l'opération	Date	Sera rempli automatiquement
Descriptif	Description de l'opération	Texte	50
Dépense	Montant Débit	Réel (Format $\pm 99\,999,99$)	Sera rempli automatiquement
Recette	Montant Crédit	Idem	Idem

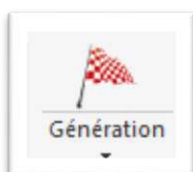
Pour insérer, cliquez sur une ligne vide dans la rubrique choisie. A vous de bien remplir vos champs, cela ne devrait vous poser aucun problème. Une fois la saisie terminée, cliquez sur **Valider**.

L'assistant vous propose de retourner sous l'éditeur de fenêtre. Cliquez sur Fermer pour rester sous l'analyse.

Double-cliquez sur la table pour faire apparaître la description, vous devriez avoir le résultat suivant :

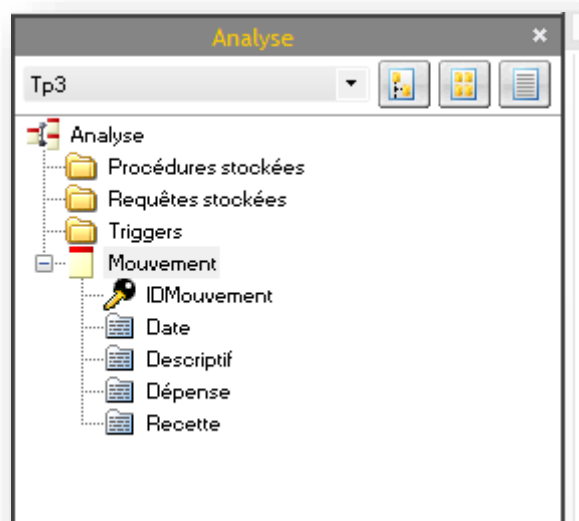


Pour que cette description de fichier soit prise en compte dans le projet nous devons générer l'analyse. Pour cela cliquez sur le drapeau

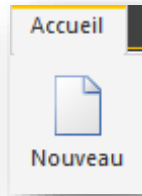


Remarques : Ce drapeau sert à lancer la génération de l'analyse. En effet, chaque fois que vous créez ou modifiez une structure de fichier, vous devez l'activer pour synchroniser l'analyse et les champs des fenêtres qui accèdent aux données.

Comme vous pouvez le constater dans le volet droit de l'Analyse nous retrouvons notre fichier :



Nous allons maintenant fabriquer les fenêtres de notre application. Cliquez sur l'icône **Nouveau**



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**

Enregistrez les modifications.

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 par programmation. Comme vous pouvez le constater, nous aurions pu prendre le choix par défaut (Afficher des données d'un fichier ou d'une requête existante). Ce choix par défaut paraît sensé, mais je préfère vous expliquer comment faire les choses manuellement. Vous aurez tout le temps ensuite de le faire faire de façon automatique.



Cliquez sur **Suivant**.

Paramètres supplémentaires

Mode de saisie

Table en affichage

Table avec saisie

Autoriser la saisie en cascade

Bloquer l'enregistrement dès l'entrée en saisie

Bandeau de sélection

Lignes : Sélection simple

Colonnes : Sans sélection

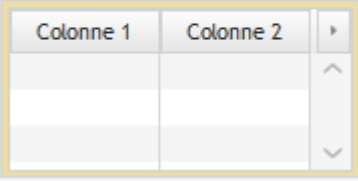
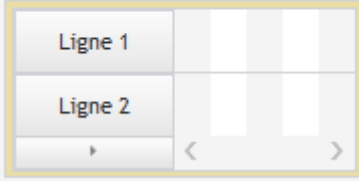
Cellules : Sans sélection

Choisissez Type de « **Table en affichage** » puis **Suivant**.

Orientation du champ table

Vertical

Horizontal

Nous allons choisir un type de présentation des données vertical. Cliquez sur **Suivant**.

Définition des colonnes

Nombre de colonnes : 5

Nombre de lignes affichées : 10

Colonne1	Colonne2	Colonne3	Colonne4	Colonne5
Texte	Texte	Texte	Texte	Texte

Cliquez sur la colonne pour modifier ses caractéristiques.

Titre : Colonne1

Type : Visible

Largeur : 100 pixels

Cet écran nous permet de prédéfinir le nombre de colonnes de la table. Indiquez qu'il y aura 5 colonnes. Cliquez ensuite sur **Suivant**.



Donnez « **Tmouv** » comme nom de la table. Et cliquez sur le bouton **Terminer**. La table est maintenant prédéfinie. Vous devriez voir ceci :



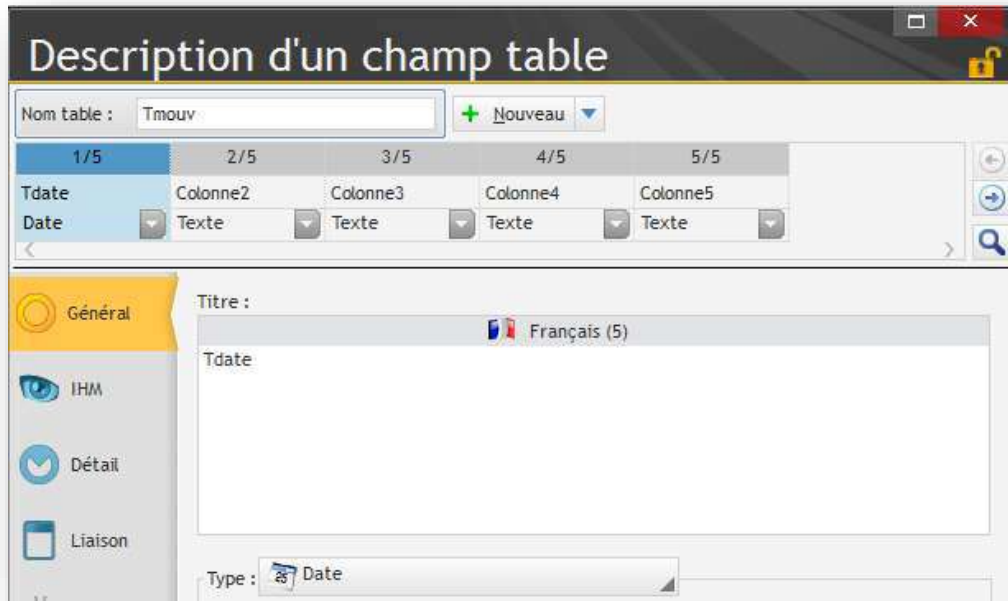
Comme vous le voyez le nom des colonnes, le type des données n'est pas défini, faites un clic droit sur la table et activez le menu **Description**. Vous allez suivre les instructions suivantes pour tout modifier :

Nom de la colonne 1 : Tdate

Type de la colonne 1 : Date

Titre de la colonne 1 : Date

Voici ce que vous devriez voir à ce stade :



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 (Réel ayant un masque identique au fichier +99 999,99)

Titre de la colonne 3 : Dépense

Nom de la colonne 4 : Tcrédit

Type de la colonne 4 : Numérique (Réel ayant un masque identique au fichier +99 999,99)

Titre de la colonne 4 : Recette

Nom de la colonne 5 : clé

Type de la colonne 5 : Numérique

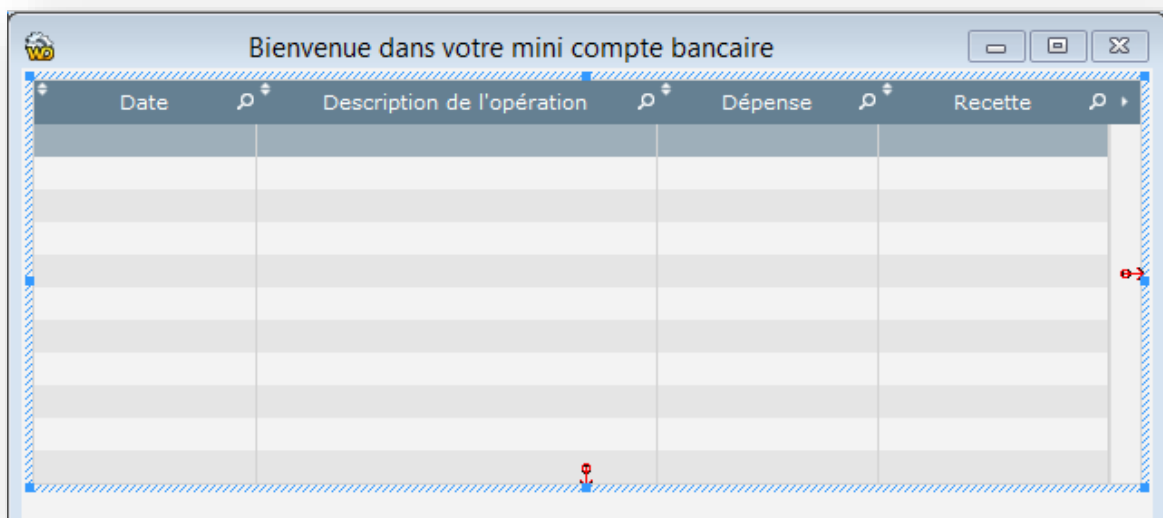
Masque de saisie : 999 999 999.

Dans l'onglet **IHM** 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.



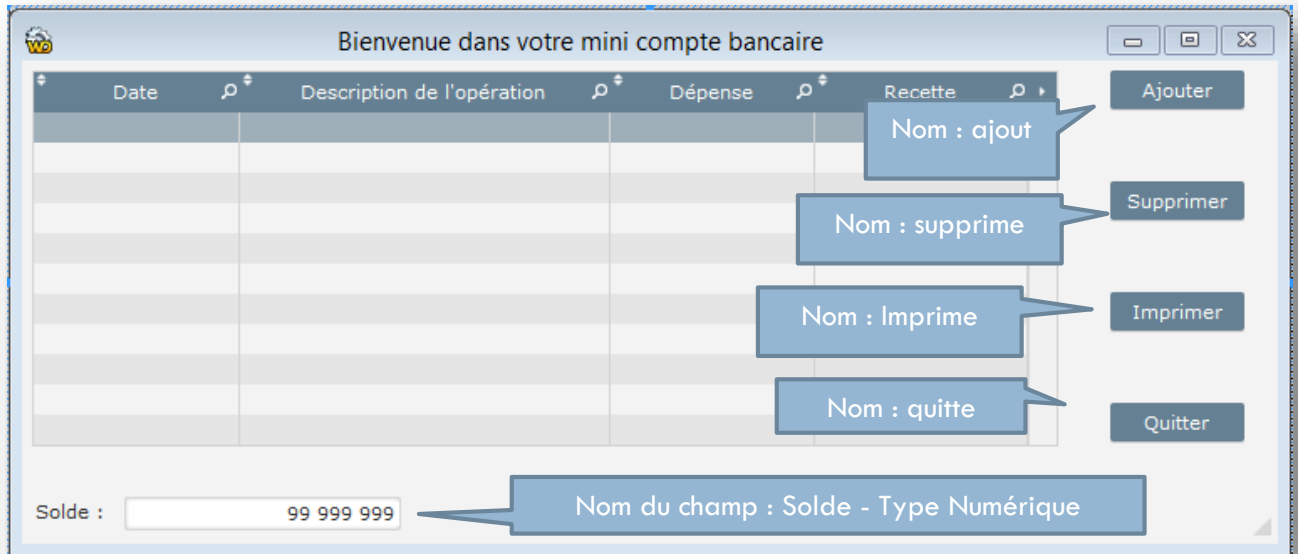
Une fois tous les champs renseignés, cliquez sur **Appliquer** puis **OK**

Vous devriez voir ceci :



Vous pouvez redimensionner les largeurs des colonnes avec la souris si certains libellés sont tronqués.

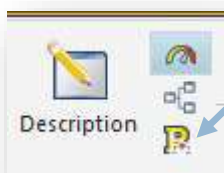
Voici maintenant, ce qu'il vous reste à concevoir pour terminer notre projet :



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.



Dans la zone **Initialisation de TP3** inscrivez le code suivant :

```
! Initialisation de Tp3
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 conçoit. 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 :



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 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 :

```

Déclarations globales de Depart *
PROCEDURE Depart()
POUR TOUT Mouvement
    TableAjouteLigne(Tmouv,Mouvement.Date,Mouvement.Descriptif,Mouvement.Dépense,Mouvement.Recette,Mouvement.IDMouvement)
FIN

```

Explication du code :

Pour tout Mouvement // Cette ligne met en place une boucle qui ordonne à WinDev d'ouvrir le fichier Mouvement et de lire la première ligne en plaçant les champs correspondants en mémoire.

La séquence tableajouteligne est connue, elle permet de positionner du texte - ici les rubriques du fichier - à l'intérieur de la table mémoire.

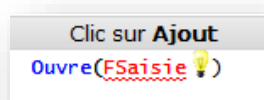
Fin // indique la fin de la boucle.

En résumé, nous pourrions lire ce code de la façon suivante :

Pour toutes les lignes du fichier « **Mouvement** »,

Ajoute les enregistrements physiques dans la table « **Tmouv** »

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



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

Dans la description de la fenêtre :

Nom logique : Fsaisie

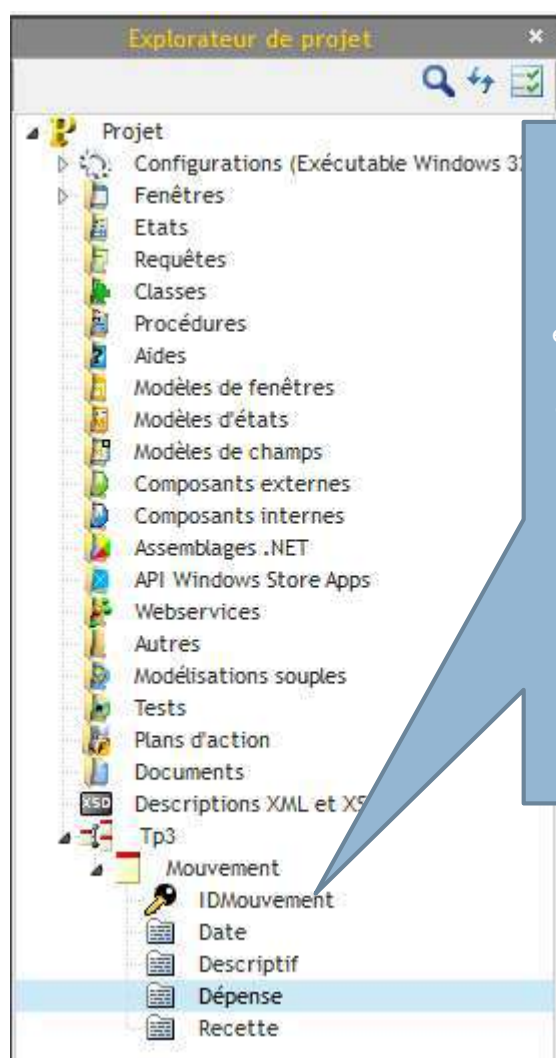
Description : Fenêtre de saisie

Titre : Saisissez votre opération



Il me semble que ça fait longtemps que vous n'avez pas sauvegardé votre projet !

Nous allons placer les champs nécessaires sur la fenêtre « Fsaisie »



Cliquez sur un champ en tenant le bouton gauche de la souris enfoncé, faites 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 garantit que WinDev peut faire l'association entre le champ sur la fenêtre et la rubrique du fichier.

Vous pouvez sélectionner plusieurs champs en maintenant la touche Ctrl

Faites glisser les champs **Date**, **Descriptif**, **Dépense**, **Recette** sur la fenêtre **Fsaisie**.

Voici le résultat :

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(Fsaisie) // Fsaisie 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 ajouter un ordre de fermeture de la fenêtre.

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

```

Clic sur Valider
EcranVersFichier(FSaisie)
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 :

```

Clic sur Valider
SI Date="" OU Descriptif="" OU (Dépense=0 ET Recette=0) ALORS
    Info("L'un des champs n'a pas été rempli!")
    RETOUR
FIN
SI Dépense <> 0 ET Recette <> 0 ALORS
    Info("Vous ne pouvez saisir qu'un montant au débit ou alors au crédit!")
    RETOUR
FIN
EcranVersFichier(FSaisie)
HAjoute(Mouvement)
Ferme

```

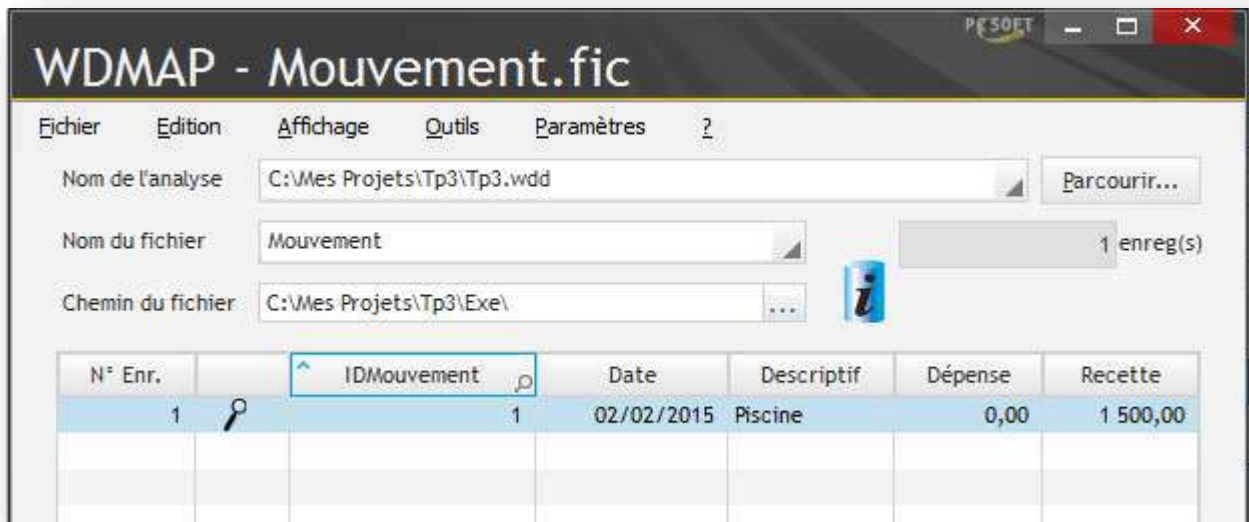
Testez votre fenêtre avec le bouton  ou en cliquant à droite de l'écran sur la fenêtre avec le bouton droit pour choisir **Tester**.

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**



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 :



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 :

```

Clic sur Supprime
HLitRecherchePremier (Mouvement, IDMouvement, Tmouv.Clé)
SI HTrouve(Mouvement) ALORS
    HSupprime(Mouvement)
    Info("La suppression est effective")
SINON
    Info("Grave problème de l'application")
FIN

```

Explications :

La première ligne fait rechercher à l'identique 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)

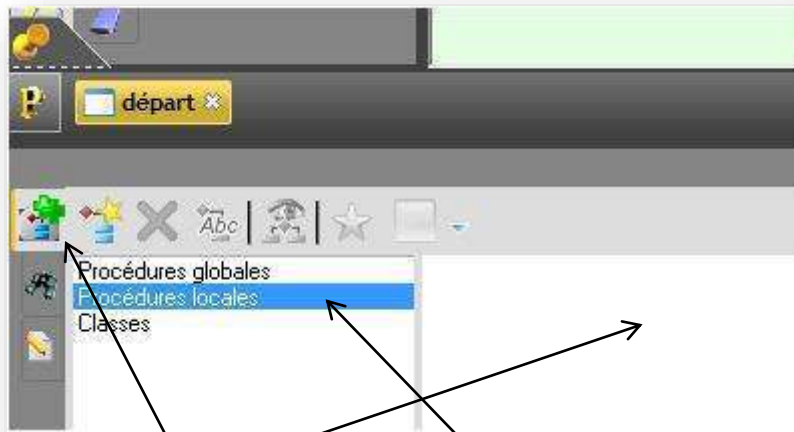
Cette commande fait rejouer un traitement existant.

Testez et vérifiez la cohérence de votre projet.

LES PROCEDURES

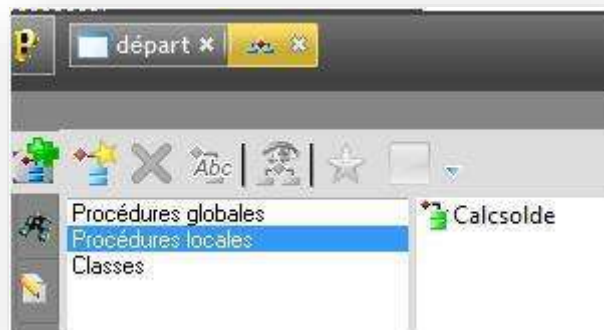
Maintenant **nous avons à calculer le solde** (Débit – Crédit), pour ce faire nous allons créer une procédure qui scanner le fichier et fera les calculs pour nous.

Pour créer une procédure



Recherchez en bas de l'écran cet onglet, Choisissez « **Procédures locales** », et faite un clic droit dans cette zone. Dans le menu contextuel choisissez « **Nouvelle procédure locale** ».

Nommez cette procédure **Calcsolde**.



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**. (Clic droit et ensuite l'option « **Code** »).

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 :

```

Procédure locale Calcsolde *
PROCEDURE Calcsolde()
  sdebit,scredit sont des réels=0 // on affecte la valeur 0 aux deux variables

POUR TOUT Mouvement
  sdebit+=Mouvement.Dépense // += signifie Sdebit=Sdebit+Mouvement.Dépense
  scredit+=Mouvement.Recette
FIN

Solde=sdebit-scredit // Solde représente le champ de la fenêtre

```

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 où 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.

```

Initialisation de départ * Si Erreur : par programme Quand Exception : pai
POUR TOUT Mouvement
  TableAjouteLigne(Tmouv,Mouvement.Date,Mouvement.Descriptif,Mouvement.Dépense,Mouvement.Recette,Mouvement.IDMouvement)
FIN
Calcsolde()

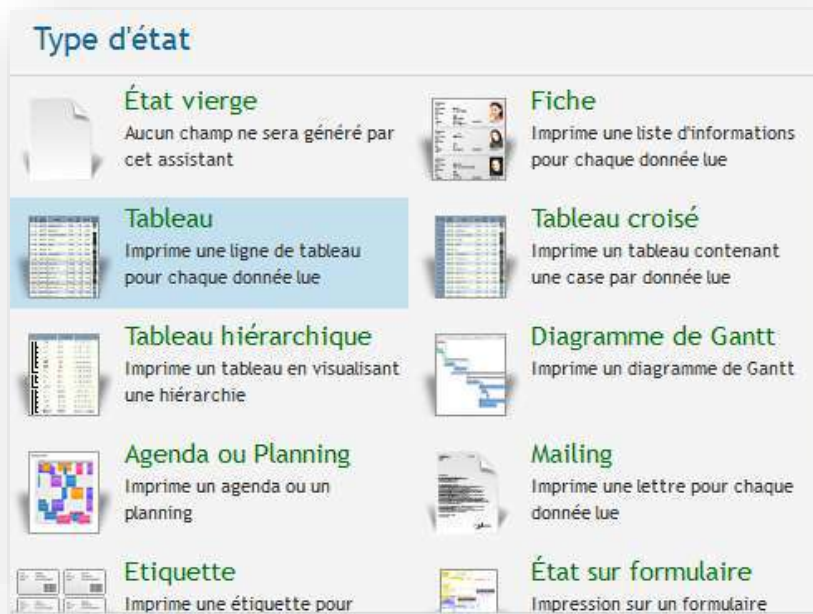
Fermeture de départ * Si Erreur : par programme Quand Exception : pai

Prise de focus de départ * Si Erreur : par programme Quand Exception : pai
TableSupprimeTout(Tmouv) // Efface la table mémoire pour éviter d'insérer les enregistrements à la suite des précédents
POUR TOUT Mouvement
  TableAjouteLigne(Tmouv,Mouvement.Date,Mouvement.Descriptif,Mouvement.Dépense,Mouvement.Recette,Mouvement.IDMouvement)
FIN
Calcsolde()

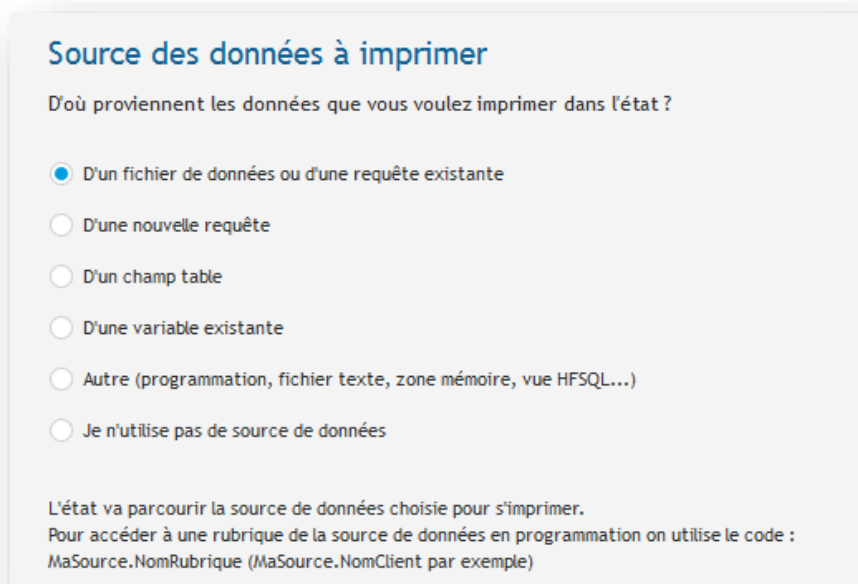
```

LES IMPRESSIONS

Il ne nous reste plus qu'à fabriquer l'état de sortie. Choisissez **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** :

<http://www.btsig.org/>

Jean-Luc Baptiste

Page 22 sur


```
Clic sur Imprimer * Si Erreur ; par program  
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

Enoncé :

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

Enoncé :

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.