

COURS WINDEV NUMERO 4



14/02/2015

Travailler avec plusieurs fichiers de données

Création d'un MCD,
Gestion des champs indexés,
Manipulation de données,
Liaison des données,
...

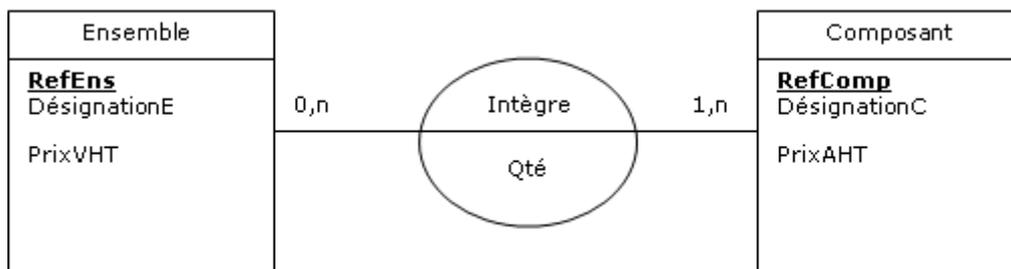
Pré requis : Cours WinDev Numéro 1,2 et 3

Cours WinDev Numéro 4

VERSION 19

L'objectif de ce cours est de vous familiariser avec l'utilisation des fichiers liés.

Nous allons créer une mini GPAO (**G**estion de **P**roduction **A**ssistée par **O**rdinateur). Vous travaillez pour un assembleur informatique, son processus de production est le suivant : Il reçoit les différentes pièces détachées (disque dur, mémoires, cartes mères...) et assemble ces différentes pièces pour en faire un modèle fini. Comme vous pouvez le percevoir, le modèle conceptuel travaillera avec 2 entités (Ensemble fini et composants). Voici une représentation du MCD :



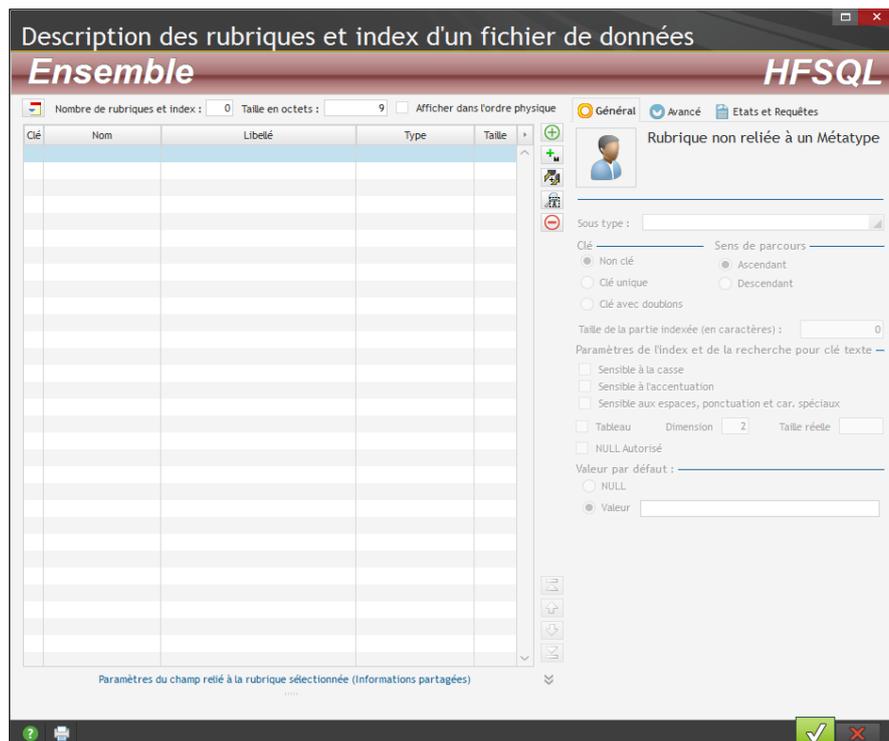
Remarque : Je n'ai pas besoin de vous rappeler que les propriétés soulignées et en gras désignent les identifiants de l'entité, donc je ne le fais pas.

Vous allez commencer par créer un nouveau projet nommé TP4. Dans l'Assistant, vous sélectionnez votre thème préféré, confirmez le choix de création d'analyse et finirez le processus de l'assistant. Un nouvel assistant de **Création d'analyse** va apparaître. Vous pouvez ensuite valider les différentes fenêtres de l'Assistant jusqu'à arriver à l'Assistant de **Création d'un fichier de données**.

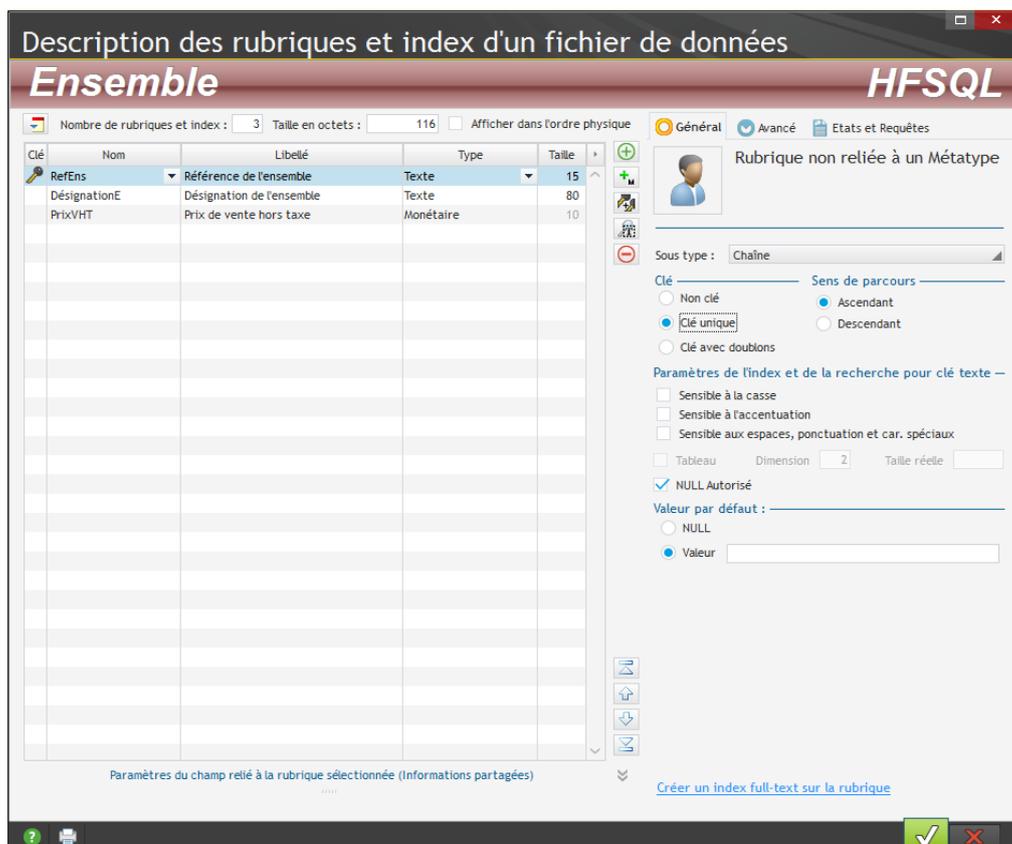
Restez sur **Créer une nouvelle description d'un fichier de données**, dans le plan suivant donnez **Ensemble** comme Nom et **Ordinateurs assemblés** comme Libellé. Pensez à choisir **Aucun** dans la zone **Identifiant automatique** puisque nous avons **Refens** qui sera notre identifiant.

Gardez **HFSQL Classic** comme **Type de base de données**. L'assistant va se terminer

Maintenant la fenêtre suivante apparaît



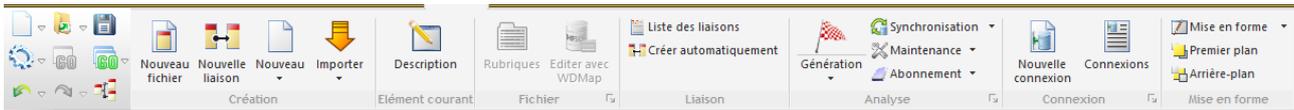
C'est elle qui va vous permettre de saisir toutes les propriétés de votre entité. Remplissez-la pour qu'elle ressemble à ceci et n'oubliez pas de mettre **RefEns** en clé unique (notre identifiant):



Une fois remplie, cliquez sur **le jet vert** et retournez sous l'éditeur de Windev. Enregistrez le projet.

Attention un écran va vous demander si vous voulez créer une fenêtre ou allez sous l'éditeur, cliquez sur **Editeur de Windev** pour rester dans l'analyse.

Si vous regardez bien votre éditeur d'analyse en haut vous devez découvrir une barre d'outils comme celle-ci :



Voyez les utilisations des différents icônes en passant le curseur de la souris dessus.

Comme vous êtes dégourdis et intelligents je vous laisse finir de concevoir le modèle !

Voilà à quoi vous devez arriver :



Vous venez donc de créer le **Modèle Logique de Données (MLD)**. Enregistrez.

Attention : N'oubliez pas que ce n'est pas parce que l'analyse est créée que les fichiers sont physiquement créés sur votre disque dur... Allez dans le code du projet « Projet / Code du projet » et inscrivez la séquence suivante dans la zone « Initialisation de TP4 » :

```
HCréationSiInexistant(Composant)
HCréationSiInexistant(Ensemble)
HCréationSiInexistant(Intègre)
HGèreIntégrité("*", "*", hCardinalité+hEnModification+hEnSuppression, Faux)
```

La dernière ligne indique à WinDev de ne pas se soucier de gérer l'intégrité référentielle, nous le ferons nous-même.

Maintenant que l'analyse est créée, nous allons commencer à construire notre application. Choisissez **Accueil / Nouveau / Fenêtre**.

Créez une fenêtre nommée **Départ** qui sera la première fenêtre du projet. Faites en sorte qu'elle ait les caractéristiques suivantes (dans sa description) :

Onglet Général :

Nom logique : départ

Description : Première fenêtre du projet

Titre : Bienvenue dans la Mini Gp

Onglet IHM :

Taille : Taille 640*480

Onglet Détail :

Type de fenêtre : Mère Mdi



Remarque : Une fenêtre mère MDI est obligatoirement la première fenêtre d'un projet WinDev. Cette fenêtre permet d'afficher toutes les fenêtres de l'application.

Une fenêtre mère MDI a les caractéristiques suivantes (non modifiables) : bords modifiables, bouton d'icônisation, d'agrandissement, menu système, barre de titre, barre d'icônes, possibilité d'avoir un menu déroulant et des barres outils.

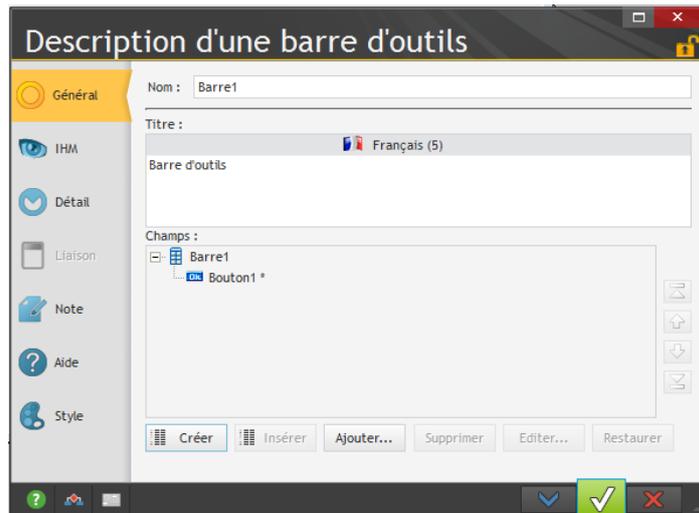
Dans une fenêtre mère MDI, seuls les champs situés dans la zone "barre d'icônes" peuvent être en saisie. En dehors de cette zone, aucun champ ne peut être en saisie, aucun clic souris n'est actif.

Nous allons placer une barre d'outils dans cette fenêtre **départ**, pour cela cliquez sur l'objet



Créer une barre d'outils (Cliquez sur le menu Fenêtre pour voir l'icône) Barre d'outils et placez le dans le bord supérieur gauche de la fenêtre. Faites un **clic droit** sur ce nouvel objet et allez dans **Description**. Nous allons placer 5 boutons dans cette barre d'outils (un pour saisir les nouveaux composants, un pour créer des ensembles, un pour imprimer la liste des composants, un pour imprimer l'ensemble et ses composants, le dernier pour quitter l'application).

Vous allez donc cliquer sur le bouton **Créer** puis choisissez **Bouton**. Le résultat obtenu devrait ressembler à ceci :



Cliquez sur **Bouton1**, puis sur le bouton **Editer**, dans cette fenêtre **Description d'un bouton** saisissez **nc** dans la zone **Nom**, supprimez le libellé par défaut, cliquez ensuite sur **Catalogue...** pour la zone **Image**.

Le catalogue apparaît, sur la gauche choisissez seulement **16*16** (nous n'aurons que des petites images), dans la zone **Recherche** frappez « **dossier** » puis appuyez sur la loupe.

Prenez le 4^{ème} bouton de la 4^{ème} ligne puis validez. Dans l'onglet **Aide** de la fenêtre description, dans la zone **Bulle d'aide** inscrivez **Saisie d'un nouveau composant**.

Pour les boutons suivants, faites de même en suivant les instructions suivantes :

Boutons	Nom	BULLE D'AIDE	MOT CLE (RECHERCHE ICONE)
Bouton2	Créer	Créer des ensembles	« Créer »
Bouton3	ImprimeC	Imprimer les composants	« Imprimer »
Bouton4	ImprimeE	Imprimer les ensembles	« Imprimer »
Bouton5	Fermer	Quitter l'application	« Annuler »

Vous choisirez les icônes les plus en phase avec le but du bouton.



Remarque : Si vous avez des difficultés pour placer les boutons (chevauchement par exemple), allez dans leur onglet **Détail** et affectez leurs les dimensions suivantes : largeur 32, hauteur 24.

Voici un exemple de ce que devrait être la barre d'outils :



Nous allons créer les 4 fenêtres filles nécessaires pour faire fonctionner notre application (fenêtre filles car elles doivent s'exécuter à l'intérieur de la fenêtre mère, logique ! Non ?). Pour le bouton **Quitter**, je vous laisse mettre le code correspondant, vous n'avez plus besoin de mon aide, sinon retour aux TP précédents.



Pour la première fenêtre associée au bouton , respectez les consignes ci-dessous :

Onglet Général de la description

Nom logique : Gcompo

Description : Gestion des composants

Titre : Gestion des composants

Onglet **IHM**

Largeur : 500

Hauteur : 350

Position : Relatif à la mère

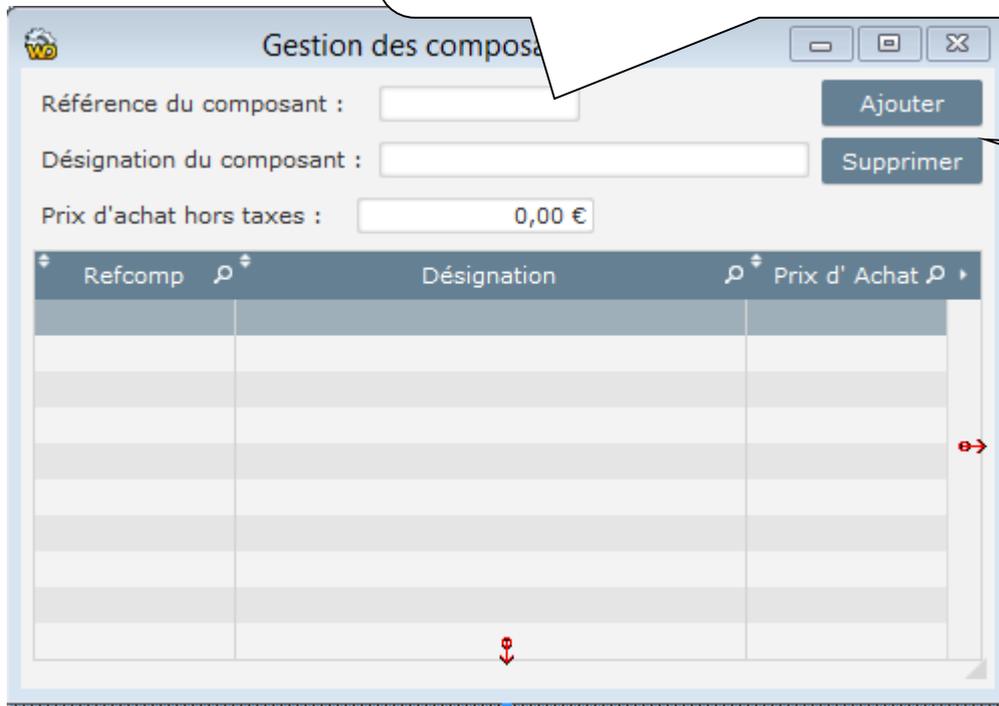
Onglet Détail

Type de fenêtre : Fille Mdi

Le code d'ouverture de la fenêtre fille sera :
OuvreFille(Gcompo)

Voici à quoi elle devrait ressembler :

Ces 3 champs sont liés avec les champs du fichier Composant (faites-les glisser depuis l'onglet  à droite de l'écran). Si certains ont des difficultés : retour au TP3. Faites en sorte que RefComp ait un masque de saisie **TOUT EN MAJUSCULES**.



Créez 2 boutons : **BAjout** et **BSup**

Nom de la table mémoire : **Table1**

RefComp : type texte

Désignation : type texte

PrixAchat : type numérique (Titre : Prix d'Achat)

Choisir dans l'assistant **Remplir la table par programmation**

Dans l'onglet **IHM**, dans la zone **Etat initial**, cliquez sur **Sélection (sans saisie)**.

Pour ceux qui auraient encore des difficultés : voir TP3

Maintenant analysons le comportement que devra avoir cette fenêtre à l'ouverture : Elle devra parcourir le fichier Composant et afficher les tuples dans la table mémoire s'il y en a.

Le bouton **Ajouter** devra :

- . Inscrire la valeur des 3 champs dans le fichier correspondant en interceptant une erreur de doublonnage si l'utilisateur saisi 2 fois une référence existante,
- . Il devra aussi mettre à blanc les 3 champs pour préparer une nouvelle saisie,
- . Réafficher la table mémoire (supprimer le contenu existant et réécrire avec le contenu du fichier),
- . Et enfin avertir l'utilisateur que tout s'est bien passé.

Le bouton **Supprimer** devra :

- . Supprimer physiquement la ligne pointée dans la table mémoire
- . Réafficher la table mémoire

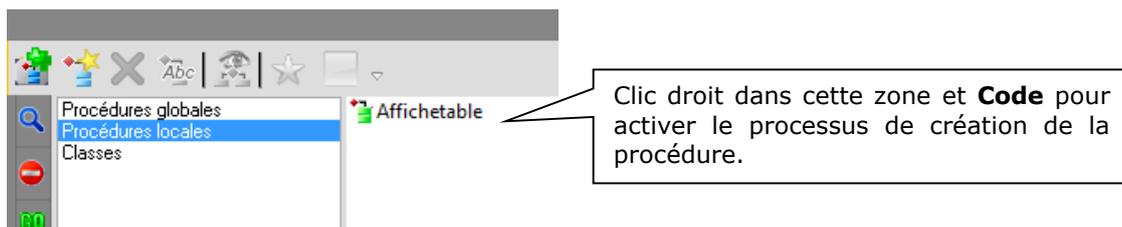
Voici les différents codes (je vous rappelle qu'ils ont déjà été étudiés à la leçon 3) :

Nous allons créer une procédure locale **affichetable** qui aura pour rôle de réafficher la table mémoire (supprimer le contenu existant et re-écrire avec le contenu du fichier) :

```
TableSupprimeTout(Table1)
POUR TOUT Composant
    TableAjouteLigne(Table1,Composant.RefComp,Composant.DésignationC,Compo
sant.PrixAHT)
FIN
```

Cette séquence devrait vous être familière donc je ne vous la commente pas, sinon retour au cours 3.

Rappel pour faire une procédure locale cliquez sur « Procédures locales » en bas de l'écran :



Déclarations globales de Gcompo (la fenêtre) :

```
// Appel de la procédure affichetable
affichetable()
```

Maintenant on est sûr que dès que cette fenêtre s'ouvrira, la table mémoire sera en phase avec le contenu du fichier.



Remarque : Certains d'entre vous se demandent peut être pourquoi utiliser des tables mémoire pour afficher le contenu des fichiers plutôt que des tables fichiers qui semblent toutes destinées à cette utilisation. Ma réponse est simple : la table mémoire vous apportera plus de souplesse et une gestion plus fine des enregistrements et des lignes dans certains cas. Et puis qui peut le plus peut le moins ;-))

Voici le code du bouton **BAjout** :

```
EcranVersFichier() // Transfère les valeurs contenues dans les champs de la fenêtre
// dans la zone de structure du fichier
HAjoute(Composant) // Passe la structure physiquement dans le fichier
RAZ(Vrai)// Efface les valeurs contenues dans les champs liés pour saisir un nouveau
composant
affichetable()
```

Voici le code du bouton **BSup** :

```

HLitRecherchePremier(Composant,RefComp,Table1.Refcomp) // Recherche dans la
table Composant,
// sur la rubrique RefComp, la valeur contenue dans RefComp de Table1
SI HTrouve ALORS // Si la valeur est trouvée alors
    HSupprime(Composant) // On supprime physiquement la ligne dans le fichier
Composant
    Info("Suppression réussie")// On informe l'utilisateur de la réussite de la
suppression
    Affichetable()
SINON // Sinon, on n'a pas trouvé ?!!
    Info("Il y a un boulon dans le potage") // Bizarre, bizarre !
FIN

```

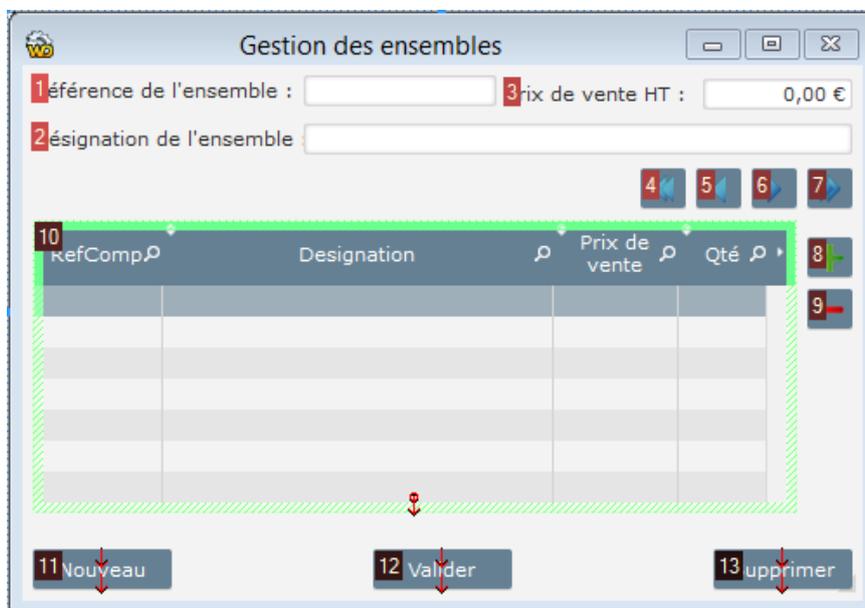
Enregistrez la fenêtre et testez votre projet sans oublier de définir qu'elle est la 1^{ère} fenêtre du projet (**départ**), normalement les comportements doivent être cohérents. Dans le cas contraire, reprenez le support et vérifiez que vous n'avez pas oublié quelques instructions.

Nous allons maintenant faire la fenêtre la plus intéressante au niveau intellectuel, celle qui gère l'assemblage d'un ordinateur.

Créez une nouvelle fenêtre (Accueil / Nouveau / Fenêtre). Prenez une fenêtre vierge et enregistrez la sous **Gensemble**.

N'oubliez pas d'en faire une fenêtre fille (cf. : étapes identiques à la fenêtre **Gcompo**) et de mettre le code d'ouverture de cette fenêtre dans le deuxième bouton de la barre d'outils de la fenêtre **départ**.

Voici la fenêtre telle qu'elle devrait être :



Objet N°	Nom	Rôle
1	RefEns	Lié à Ensemble.RefEns (TOUT EN MAJUSCULES)
2	PrixVHT	Lié à Ensemble.PrixVHT
3	Désignation	Lié à Ensemble.Désignation
4	BPremier	Bouton nous positionnant sur le premier enregistrement du fichier Ensemble
5	BPrecedent	Bouton nous positionnant sur l'enregistrement précédent
6	BSuivant	Bouton nous positionnant sur l'enregistrement suivant
7	BDernier	Bouton nous positionnant sur le dernier enregistrement du fichier Ensemble
8	Table1	Table mémoire contenant des rubriques similaires à la structure de Composant
9	BAjout	Bouton nous permettant d'ajouter un composant à l'ensemble
10	BSupprime	Bouton nous permettant de supprimer un composant à l'ensemble
11	BNouveau	Bouton nous permettant de créer un nouvel ensemble
12	BValide	Bouton nous permettant de valider un ensemble
13	BSupp	Bouton nous permettant de supprimer un ensemble

Analyse des éléments de la fenêtre **Gensemble**.

A l'ouverture de la fenêtre dans le code d'initialisation, nous devons nous positionner sur le premier ensemble (s'il existe) et rechercher tous ces composants constitutifs.

Voici comment faire :

Il nous faut lire la référence de l'ensemble (RefEns), parcourir le fichier de liaison (Intègre) sur la clé RefEns, tant que l'on trouve Ensemble.RefEns=Intègre.RefEns il nous faut chercher dans la table composant la clé de liaison (Intègre.RefComp = Composant.RefComp) et récupérer Intègre.Qté. C'est clair ?

Non !! Alors relisez en ayant l'analyse en tête.

Nous allons créer une procédure locale qui, prenant en paramètre le code RefEns, nous remplit la table. Facile et efficace.

Notez qu'une requête SQL nous aurait évité pas mal de lignes, à vous de juger.

```

PROCEDURE remplir (code)
TableSupprimeTout(Table1)
FichierVersEcran(Gensemble) // on affecte aux champs de la fenêtre les données liées

POUR TOUT Intègre AVEC RefEns = code
    HLitRecherche(Composant,RefComp,Intègre.RefComp) // A vous de trouver
    TableAjouteLigne(Table1,Composant.RefComp,Composant.DésignationC,Compo
sant.PrixAHT,Intègre.Qté)
FIN

```

En fait, comme vous pouvez le voir, nous venons de « jongler » avec 3 fichiers sans trop de difficultés. Regardons le code d'initialisation de la fenêtre (Déclarations globales de la fenêtre Gensemble)

```

HLitPremier(Ensemble,RefEns)
SI HNbEnr(Ensemble)>0 ALORS // si le fichier Ensemble contient au moins un
enregistrement
    remplir(Ensemble.RefEns)
SINON
    Info("Le fichier est vide")
FIN

```

Voyons le code nécessaire aux boutons 4, 5, 6 et 7 (pour les icônes, faites un tour du côté des flèches en 16x16).

Je vous donne le code du bouton 4, vous êtes suffisamment aguerri pour pouvoir créer le code des boutons 5, 6 et 7.

Comment ça ? Réfléchissez un peu et vous y arriverez ! Non mais, il faudrait tout leur faire...

(Si vous ne savez pas... hop, au TP 3).

D'ailleurs, je ne vous le donne pas, non je ne suis pas méchant, j'ai seulement la flemme de vous réécrire la séquence de code du dessus. Eh oui, c'est ce bout de code, analysez le et regardez à quel point il est idéal pour notre bouton 4.

Nous allons traiter le bouton 9. Ce bouton doit nous permettre de sélectionner dans le fichier **Composant** le composant que nous voulons insérer dans la table mémoire. Nous allons voir si WinDev peut nous générer la fenêtre qui va bien. Allez sur **Accueil / Nouveau / Fenêtre**. Cherchez une fenêtre nommée **Vision** et validez.

Dans l'Assistant qui arrive, cliquez sur **Composant**, cliquez encore sur **Suivant**, vérifiez que tous les champs soient cochés, cliquez encore sur **Suivant** 2 fois. Interdisez la saisie dans le champ table. Puis saisissez **Selection** comme **Nom**. Enfin cliquez sur **Terminer**.

Vous avez maintenant une fenêtre fonctionnelle dans laquelle on peut supprimer le bouton **Nouveau** (on peut utiliser la création de composants avec notre 1^{er} bouton du menu). Réduisez-la pour qu'elle soit plus petite. Voici à quoi elle pourrait ressembler :



Regardons comment nous pourrions modifier le code pour qu'il nous soit utile. Allez dans le code du bouton **OK**. Vous pouvez voir que la fenêtre retourne **vrai** si on clique sur **OK**, faux sur **Annuler**.

Qui dit bandeau déplacé dit structure mémoire du fichier contenant les valeurs recherchées ! Donc, nous n'aurons qu'à lancer la fenêtre et tester si elle retourne **vrai**.

Fermez-la et retournez sur **Gensemble** dans le code du bouton **BAjout**. Saisissez la séquence de code suivante :

```

resultat est un booléen=Faux
resultat=Ouvre(selection)
SI resultat=Vrai ALORS // la fenêtre nous a renvoyé vrai, donc le bandeau a été déplacé
    TableAjouteLigne(Table1,Composant.RefComp,Composant.DésignationC,Compo
sant.PrixAHT)
FIN

```

Je vous laisse méditer sur la différence entre **tableajoute** et **tableajouteligne** (n'oubliez pas d'utiliser l'aide, touche F1).

Sauvegardez et exécutez la fenêtre, normalement le clic sur le bouton **BAjout** lance bien la fenêtre **selection** qui nous retourne vrai ou faux selon le bouton cliqué.

Pour saisir la **Qté**, cliquez dans la colonne pour saisir le nombre souhaité. N'oubliez pas de vérifier que la Table est **En saisie** et d'avoir seule la rubrique Qté **En saisie / actif**, les autres restant en mode **Affichage seulement**.

Pour le bouton 10 (**BSupprime**), la séquence de code a déjà été étudiée lors du TP3. Pour le bouton Nouveau (**BNouveau**), il vous faut programmer un traitement qui efface les champs 1, 2, 3 et la table mémoire, vous savez faire, donc je passe.

Le bouton Supprime (**BSupp**) va nous obliger à scanner 2 tables et supprimer les références « Ensemble » recherchées. Il faut donc prendre la valeur du champ 1 (RefEns), parcourir la table Intègre sur le champ RefEns et supprimer tous ceux égaux au champ RefEns. Et ensuite le supprimer dans la table Ensemble.

Voici une séquence de code qui devrait être efficace :

```

SI OuiNon("Voulez vous vraiment supprimer cet enregistrement ?")=Oui ALORS
    POUR TOUT Intègre AVEC Intègre.RefEns = RefEns

```

```

        HSupprime(Intègre)
    FIN
    HLitRecherchePremier(Ensemble,RefEns,RefEns)
    SI HTrouve ALORS
        HSupprime(Ensemble)
        Info("La suppression est réussie")
        ExécuteTraitement(Gensemble,trtInit)
    FIN
FIN

```

Vous n'avez pas besoin que je vous commente ce code, il est limpide.

La séquence du bouton **Valider** va être un peu plus rude, mais guère plus. En fait, lorsqu'un nouvel ensemble est créé ou modifié par l'adjonction d'un nouveau composant, il va nous falloir faire plusieurs actions différentes.

Il faut supprimer l'ensemble existant dans le fichier pour le recréer avec ses nouveaux composants, donc on commence par une phase de suppression (voir code ci-dessus) et on achève par une phase d'insertion dans le fichier.

Le code est long mais n'est pas complexe, prenez le temps de l'étudier.

```

i est un entier

HLitRecherchePremier(Ensemble,RefEns,RefEns)
SI HTrouve(Ensemble) ALORS // l'enregistrement existait dans le fichier, on va le
supprimer
    HLitRecherche(Intègre, RefEns, RefEns)
    POUR TOUT Intègre AVEC Intègre.RefEns = RefEns
        HSupprime(Intègre)
    FIN
    HSupprime(Ensemble)
FIN
// Maintenant inscrivons l'ensemble dans le fichier
Ensemble.RefEns=RefEns
Ensemble.DésignationE=DésignationE
Ensemble.PrixVHT=PrixVHT
HAjoute(Ensemble)

// Remplissons le fichier Intègre
POUR i=1 A TableOccurrence(Table1) // Pour i=1 aux max de la table mémoire
    Intègre.RefComp=Table1.RefComp[i] // [i] indique l'indice de la ligne
    Intègre.RefEns=RefEns
    Intègre.Qté=Table1.Qté[i]
    HAjoute(Intègre)
FIN
Info("L'ajout s'est bien passé")

```

Voilà, la super fenêtre de gestion des ensembles est finie... N'oubliez pas de sauvegarder votre œuvre.

Le troisième bouton de la barre d'outil sert à imprimer la liste des composants, vous avez déjà étudié la procédure dans le TP 3 donc je n'y reviens pas.

Voyons comment imprimer les ensembles et leurs composants (bouton 4)

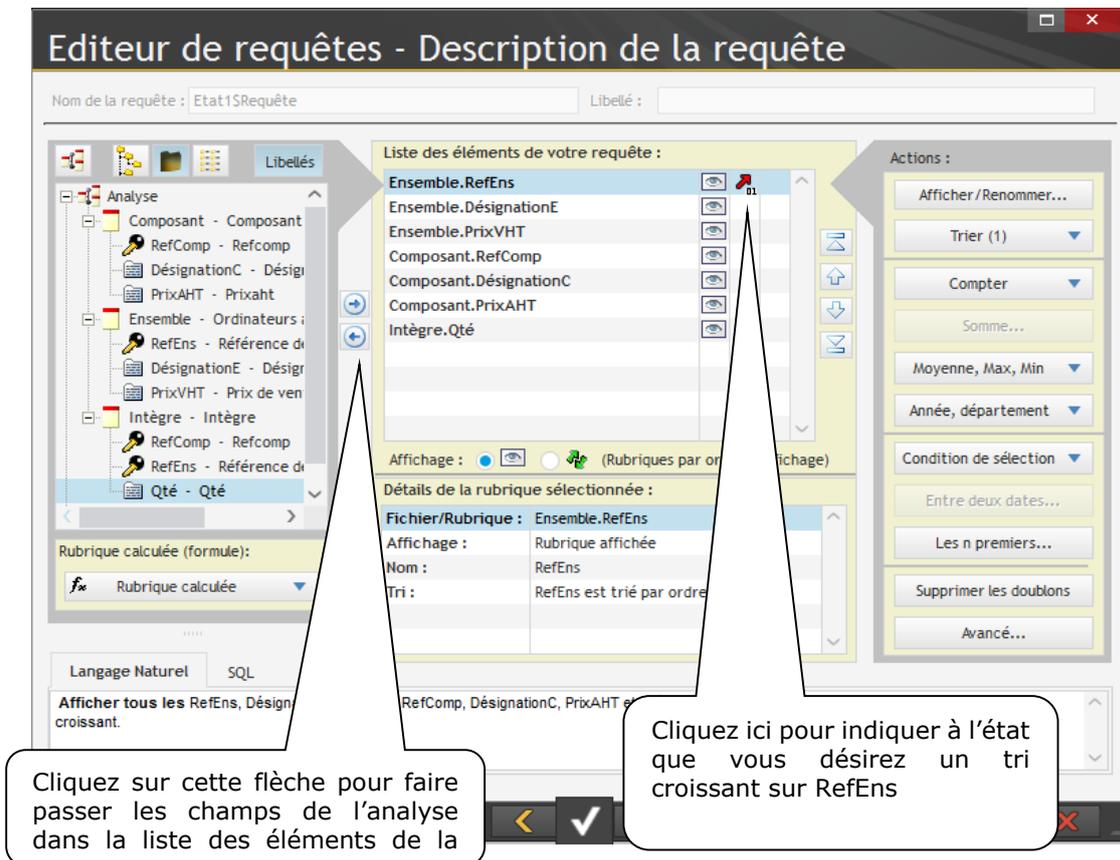
Nous allons nous laisser guider par l'Assistant pour faire un état avec des données provenant des 3 tables. Allez dans **Accueil / Nouveau / Etat**.

Choisissez ensuite un type d'état tableau, cliquez sur Suivant.

Vérifiez que les données proviennent bien d'une requête et cliquez sur Suivant.

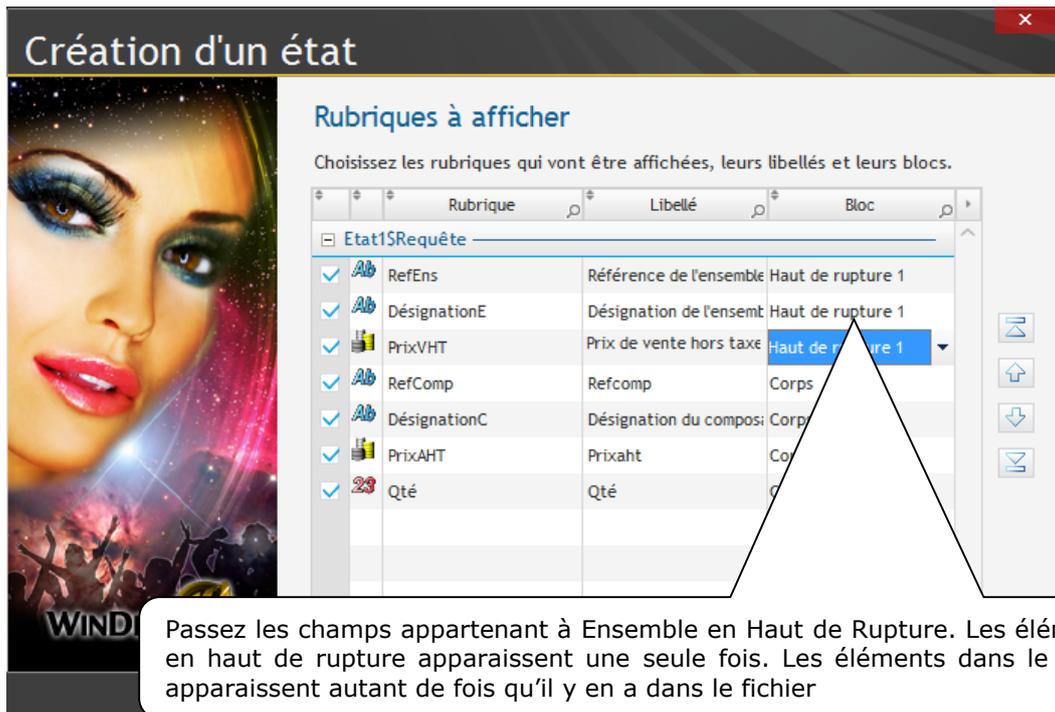
Faites-lui comprendre que les données proviennent d'une nouvelle requête que vous voulez définir maintenant, cliquez sur Suivant.

Maintenant vous devez avoir cette fenêtre à l'écran :



Cliquez ensuite sur **Suivant**. A la question « Voulez-vous des ruptures dans l'état » répondez **Oui** et cliquez sur **Suivant**.

La fenêtre suivante vous demande de sélectionner la rupture, vérifiez que **Référence de l'ensemble** soit bien coché, cliquez sur **Suivant**.



Création d'un état

Rubriques à afficher

Choisissez les rubriques qui vont être affichées, leurs libellés et leurs blocs.

Rubrique	Libellé	Bloc
Etat1SRequête		
<input checked="" type="checkbox"/> Ab RefEns	Référence de l'ensemble	Haut de rupture 1
<input checked="" type="checkbox"/> Ab DésignationE	Désignation de l'ensem	Haut de rupture 1
<input checked="" type="checkbox"/> PrixVHT	Prix de vente hors taxe	Haut de rupture 1
<input checked="" type="checkbox"/> Ab RefComp	Refcomp	Corps
<input checked="" type="checkbox"/> Ab DésignationC	Désignation du composi	Corps
<input checked="" type="checkbox"/> PrixAHT	Prixah	Co
<input checked="" type="checkbox"/> 23 Qté	Qté	Co

Passez les champs appartenant à Ensemble en Haut de Rupture. Les éléments en haut de rupture apparaissent une seule fois. Les éléments dans le corps apparaissent autant de fois qu'il y en a dans le fichier

Cliquez ensuite sur **Suivant**. La fenêtre suivante vous demande quels sont les champs numériques sur lesquels vous voulez effectuer un calcul, essayer de faire en sorte qu'elle ressemble à ceci :



Création d'un état

Compteurs, sommes et moyennes

Pour chaque rubrique de l'état, vous pouvez imprimer un compteur, une somme ou une moyenne. Sélectionnez le type de calcul souhaité dans la colonne de droite de la table ci-dessous. Sélectionnez 'Aucun calcul' pour ne pas imprimer de calcul.

Rubrique	Libellé	Calcul
PrixVHT	Prix de vente hors taxe	Aucun calcul
PrixAHT	Prixah	Somme sur rupture
Qté	Qté	Aucun calcul

Aucun calcul

Cliquez sur **Suivant**. Les choix suivants concernent les formats du papier, laissez-les par défaut sauf l'orientation que vous prendrez en **Paysage** et cliquez sur **Suivant**.

Choisissez un gabarit et **Suivant**. Nommez cet état **Ensemble**, comme titre « Etat des ensembles ». Ouf ! Vous pouvez enfin cliquer sur **Terminer**.

Comme votre état est plus large que la feuille, passez en mode paysage. Votre état apparaît maintenant dans l'éditeur d'état. Si vous voulez faire quelques retouches, c'est maintenant.

Il ne vous reste plus qu'à activer l'état dans le bouton de la barre d'outils avec la séquence de code suivante :

```
iAperçu(i100)  
iImprimeEtat(Iensemble)
```

Voilà, ce modeste "mini-lab" est fini. Vérifiez que tous les boutons fonctionnent, que les traitements sont cohérents. Il y aurait des améliorations possibles (concevoir un compteur qui totalise les prix HT des composants entrant dans la composition d'un ensemble...). Mais je considère que vous êtes apte à le faire vous-même.

Vous avez mérité la pause syndicale...

Exercice Pratique

Enoncé :

La municipalité de votre lieu de résidence désire ouvrir une bibliothèque municipale. Connaissant vos nombreuses compétences informatiques, l'adjoint au maire chargé de la culture vous demande de concevoir le logiciel chargé de la gestion des prêts de livre et des achats.

Voici quelques éléments de réflexion :

Chaque livre est référencé par un numéro ISBN qui sert d'identifiant et qui est défini de façon internationale, par exemple le livre «WinDev, WebDev, WinDev Mobile» écrit par Jean-Luc Baptiste aux éditions ENI a le numéro ISBN 978-2-7460-5898-9.

Le directeur de la bibliothèque voudrait pouvoir saisir les références complètes du livre (la date d'achat, son prix d'achat, la quantité achetée, le fournisseur, sa classification : policier, aventure...).

Cette saisie doit lui permettre de connaître le nombre total de livre en stock, le coût du stock, le nombre et le coût par type de livre, le nombre de livre par auteur avec le titre et la quantité en stock. Si vous avez d'autres statistiques à lui proposer, elles seront les bienvenues.

Un client est référencé par un numéro aléatoire généré par la base de données. Les emprunteurs doivent pouvoir être ajoutés, modifiés, supprimés.

Les règles de gestion sont les suivantes :

- Une personne ne peut pas emprunter plus de 3 livres de la bibliothèque. Elle peut en emprunter 3 d'un coup ou les uns après les autres ;
- Elle doit ramener un livre 15 jours après son emprunt. Le système doit pouvoir faire un état des livres empruntés non rentrés et éditer un listing avec les noms, prénoms, adresses et N° de téléphone des emprunteurs et la liste des livres qu'ils auraient dû rendre. Ainsi, le bibliothécaire pourra téléphoner à la personne pour lui demander de rendre les ouvrages.

On doit pouvoir imprimer ou visualiser :

- La liste complète des livres ainsi que leur quantité en stock ;
- La liste des livres en prêt ;
- La liste des inscrits n'ayant jamais emprunté un livre.