TABLEAUX D'OBJETS ET DATABINDING

L'AUTEUR DE CE TUTORIEL

Dominique DAUSSY
3 Place de la Galaxie
76400 TOUSSAINT

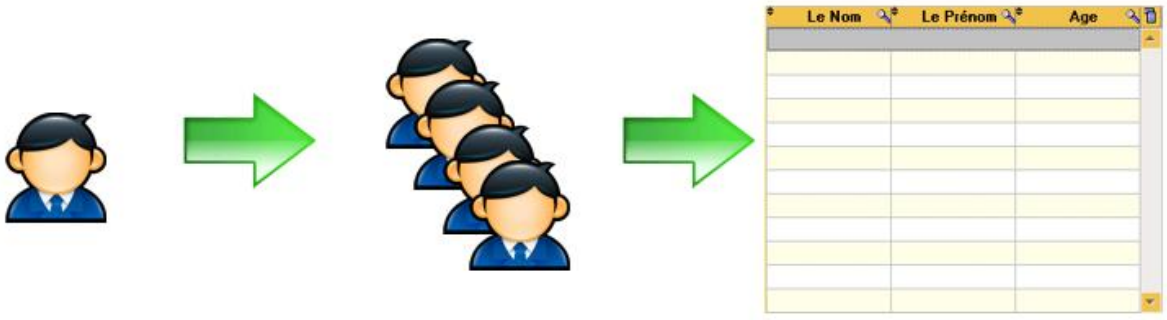
Mail : daussy.dominique@orange.fr
Web : http://dominique-daussy.fr

Je suis développeur et j'utilise les produits pc-soft depuis Hyper-Screen ! J'ai décidé d'exposer mon savoir faire pour que vous puissiez en profiter. Peut être vous direz vous, je le veux dans mon équipe ! En effet je suis en recherche d'emploi à ce jour (04/01/2013).. Alors mieux qu'un CV, voici la démonstration d'une partie de mon savoir faire.

PRESENTATION DU TUTORIEL

Date : 04 Janvier 2013 - Version : 1.00A

Ce tutoriel vous montre comment créer des objets, les utiliser en tableaux mémoire pour pouvoir ensuite afficher les données dans une table écran windev (table liée à une variable de type tableau).



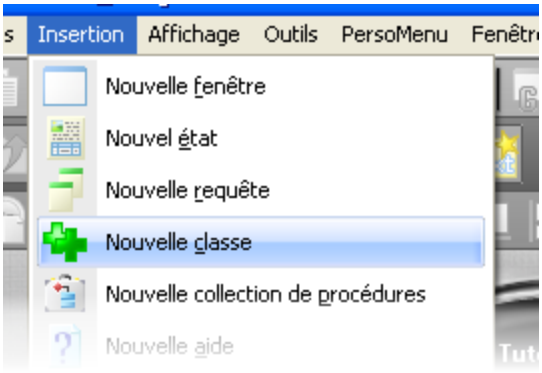
Vous devez avoir un minimum de connaissances en POO (Création d'une classe, ajout de méthodes et propriétés. L'exemple a été développé avec la version 17 de windev. Les manipulations dans windev ne sont pas détaillées, vous êtes censé être un utilisateur courant windev. Certains exemples de codes sont simplifiés et ne sont pas obligatoirement optimisés pour des questions de compréhension. Ce code fonctionne avec windev mais devrai logiquement pouvoir être utilisé avec les autres produits pc-soft (webdev et windev mobile).

CREONS TOUT D'ABORD UN NOUVEAU PROJET

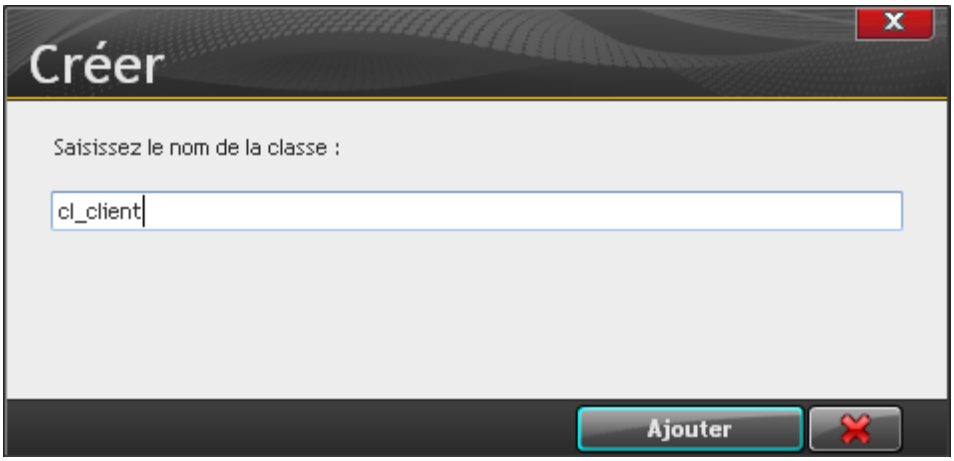
Pour l'exemple, nous allons créer le projet tutoriel_001. Je suppose que vous connaissez la procédure pour créer un projet avec windev ! Créez un projet de type 'Application' sans base de données avec le style que vous souhaitez.

CREATION DE LA CLASSE CL_CLIENT

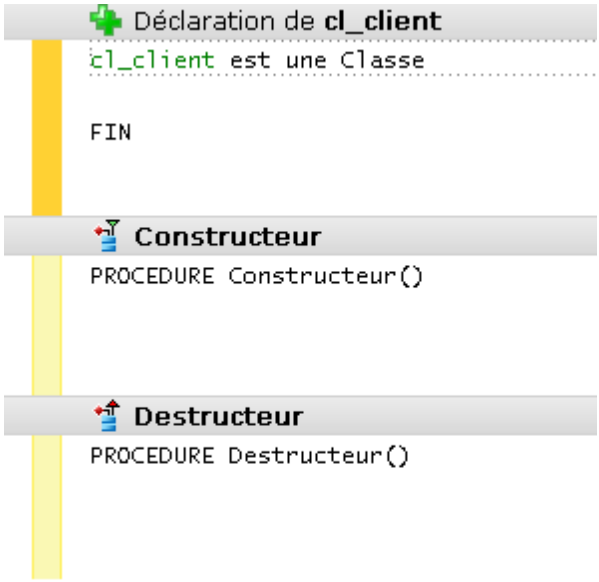
Créez la classe cl_client en utilisant le menu 'Insertion>Nouvelle classe' (vous pouvez également ajouter des classes à divers endroits dans l'interface windev



Saisissez le nom de classe 'cl_client' et validez.

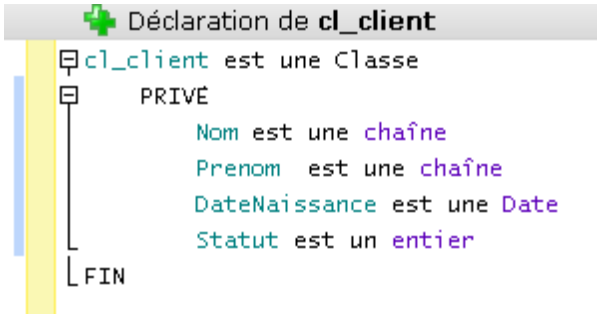


Vous obtenez le squelette de classe suivant



SAISISSONS LES MEMBRES DE LA CLASSE

Un client aura un nom, un prénom, un statut (sans, prospect, litige en cours) et une date de naissance. Il suffit d'ajouter les membres correspondants ce qui donne le code ci-dessous



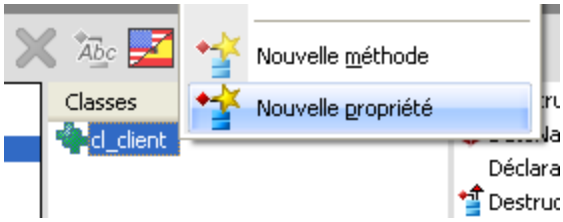
Les membres sont déclarés en PRIVE pour respecter le principe d'encapsulation des données. Seule la classe sera autorisée à modifier ses propres membres.

COMMENT MODIFIER LES INFORMATIONS DE MES OBJETS ?

Puisque les membres de ma classe sont privés, il n'y a actuellement aucun moyen de modifier le nom, le prénom et les autres informations de mes objets. Les objets resteront désespérément 'vides' et ne serviront dont à rien !

Pour initialiser nos objets, nous allons ajouter des propriétés (les getter et setter en POO) pour tous les membres. Il s'agit ici de donner la possibilité de pouvoir **hydrater** nos objets en terme POO.

Pour créer une propriété, il suffit de sélectionner la classe et de faire un clic droit sur son nom. Dans le menu contextuel qui apparait, sélectionnez l'option 'Nouvelle propriété'.



Dans la fenêtre qui apparait, saisissez 'p_nom' et cliquez sur le bouton 'Ajouter'. Vous apercevez alors 2 zones de code dans l'éditeur windev ! Une qui porte le nom de 'Récupération' et une autre qui se nomme 'Affectation'.

Les propriétés d'un objet permettent en générale de connaitre OU modifier un état de vos objets. C'est pour cette raison que windev propose deux zones de code indépendantes mais avec le même nom. Une des zones servira à retourner l'état (Récupération) et l'autre zone servira à initialiser un état (Affectation).

Dans le code d'une de vos fenêtres, vous pourrez donc voir ceci :

Info(MonClient :p_Nom)

Dans cet exemple, c'est le code de récupération de l'objet qui est sollicité puisqu'on effectue une 'lecture' de la propriété.

MonClient :p_Nom='DUPONT'

Dans cet autre code c'est le code d'affectation qui est sollicité puisque nous assignons une nouvelle valeur à la propriété.

Vous comprenez en fait qu'une propriété fonctionne a peu prêt comme une variable simple qu'il est possible de lire ou de modifier et que selon l'action (lire/modifier) ce n'est pas le même code qui est sollicité.

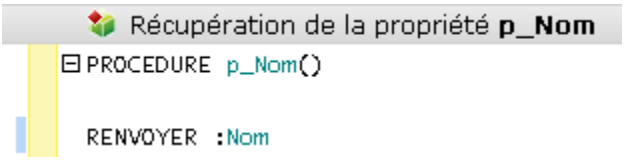
Nous pourrions nous passer des propriétés en créant deux méthodes Set_Nom et Get_Nom dans la classe mais j'y vois deux inconvénients majeurs :

- Deux noms différents Set_ et Get_
- Seules les propriétés peuvent être utilisées avec le DataBinding !

LA PROPRIETE P_NOM

La propriété p_nom est une propriété simple ! Soit elle retourne la valeur du membre :Nom soit elle la met à jour.
Nous souhaitons par contre que le nom des clients soit en majuscules et sans espace devant/Derrière.

Dans le code de récupération de la propriété p_nom saisissez le code suivant :



Dans le code d'affectation de la propriété p_nom, saisissez le code suivant :

Affectation de la propriété **p_Nom**

```
PROCEDURE p_Nom(Valeur)

  :Nom=ChaîneFormate(Valeur,ccSansEspace+ccMajuscule)
```

La première propriété est terminée ! Nous allons la tester.
Créez une nouvelle fenêtre que vous appelez 'Test'. Dans cette fenêtre, créez un bouton et saisissez le code ci-dessous :

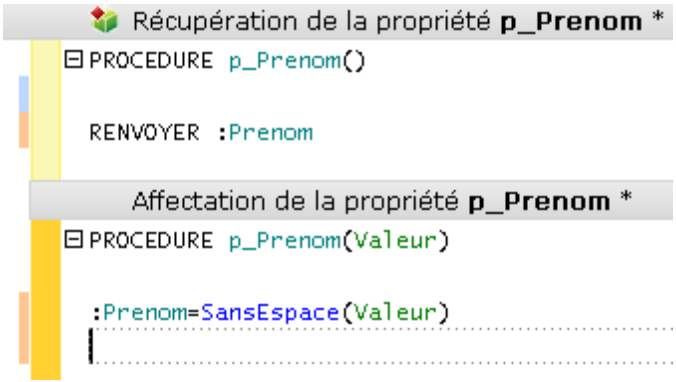
Clic sur **Bouton1**

```
LOC_MonClient est un cl_client()        // Création d'un objet client
LOC_MonClient:p_Nom ="Dupond"           // Affectation de la propriété 'p_nom'
Info(LOC_MonClient:p_Nom)               // Récupération de la propriété 'p_nom'
```

Exécutez la fenêtre en mode test et cliquez sur le bouton. Vous devriez voir une fenêtre info avec le texte « DUPONT » en majuscules. Magique non ? Mais ce n'est encore rien du tout !

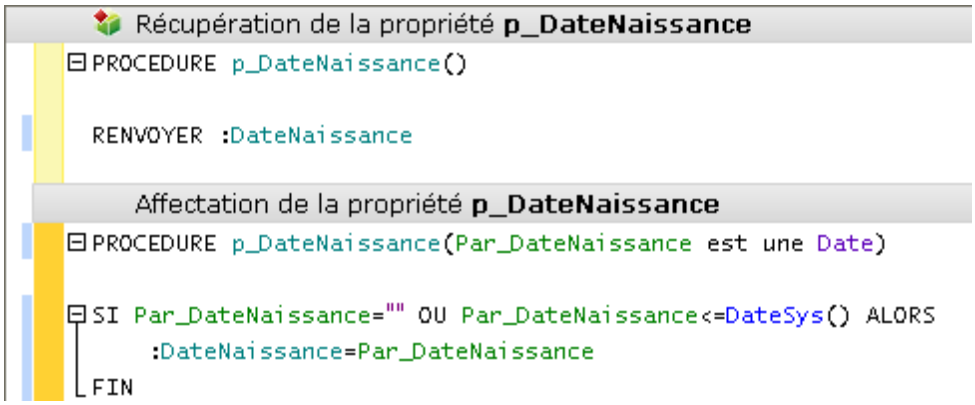
LA PROPRIETE P_PRENOM

La propriété p_Prenom n'a rien de particulier, elle permet de récupérer ou d'initialiser le prénom du client. Ajoutez cette propriété dans votre classe comme vous venez de le faire pour p_Nom et saisissez le code ci-dessous.



LA PROPRIETE P_DATENAISSANCE

La propriété p_DateNaissance permet de récupérer et d'initialiser la date de naissance du client. On peut ajouter un contrôle pour refuser une date si elle est supérieure à la date du jour ! Les dates 'null' sont acceptées. Ajoutez cette propriété dans votre classe et saisissez le code ci-dessous.



LA PROPRIETE P_STATUT

La propriété p_Statut permet de récupérer et d'initialiser le code statut du client. Un contrôle doit être effectué pour affecter un code statut valide.

Nous allons ajouter des constantes à notre classe pour les 4 statuts suivant : Sans, Prospect, Client, Perdu. Votre déclaration de classe doit ressembler à ceci

Déclaration de **cl_client**

```
CONSTANTE
    STATUT_CLIENT_SANS = 0
    STATUT_CLIENT_PROSPECT = 1
    STATUT_CLIENT_CLIENT = 2
    STATUT_CLIENT_PERDU = 3
FIN

cl_client est une Classe
    PRIVE
        Nom est une chaîne
        Prenom  est une chaîne
        DateNaissance est une Date
        Statut est un entier
FIN
```

Et le code de la propriété p_Statut qui utilise ces constantes pour contrôler la valeur du code à l'affectation.

Récupération de la propriété **p_Statut**

```
PROCEDURE p_Statut()

RENVOYER :Statut
```

Affectation de la propriété **p_Statut**

```
PROCEDURE p_Statut(Par_CodeStatut est un entier)

// On autorise l'affectation seulement si le code est connu
SI Par_CodeStatut DANS (::STATUT_CLIENT_SANS,::STATUT_CLIENT_CLIENT,::STATUT_CLIENT_PROSPECT,::STATUT_CLIENT_PERDU) ALORS
    :Statut=Par_CodeStatut
FIN
```

Les propriétés de bases sont terminées. Pour changer, nous allons jouer avec cet objet et créer une table pour afficher et modifier un tableau de clients.

DATABINDING !

Nous allons maintenant créer une fenêtre avec une table écran qui affichera les données d'un tableau d'objets cl_client. Nous pourrons ajouter de nouveaux clients et modifier directement les informations dans la table. Vous constaterez que le code dans l'interface utilisateur sera vraiment minime.

Créez une fenêtre vierge : F001_TableEtModification

Pour que le DataBinding puisse fonctionner, il faut déclarer un tableau d'objets dans le code d'initialisation de la fenêtre. Ce tableau sera donc global à la fenêtre et il pourra donc être utilisé comme 'source de données' des composants visuels de windev. (Table, ZoneRepetee, liste et autres…)

Saisissez le code ci-dessous dans le code d'initialisation de votre fenêtre

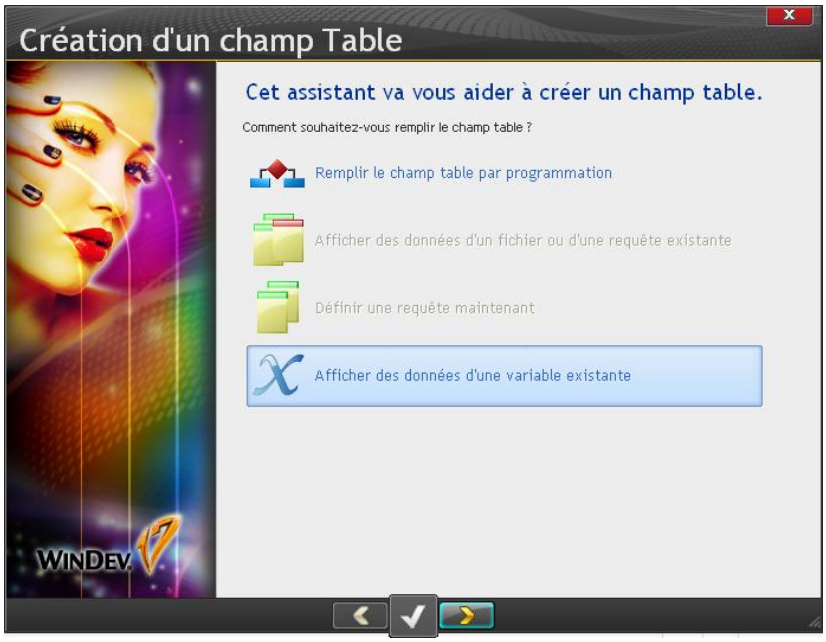
Déclarations globales de **F001_TableEtModification**

```
Gbw_TableauClients est un tableau de cl_client()
```

Notes : Tous les tableaux globaux accessible (fenêtre, projet, autres objets etc..) peuvent être utilisés comme 'source de données' pour un composant visuel windev (table, liste zone répétées etc..).

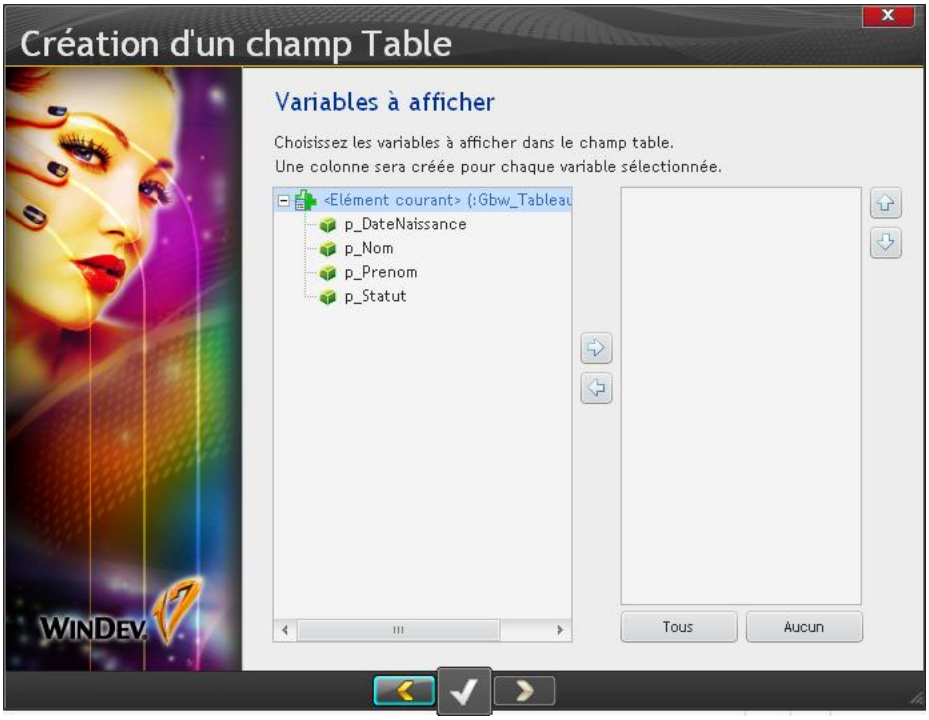
CREER UNE TABLE WINDEV CONNECTEE AU TABLEAU D'OBJETS

Déposez une table windev dans votre fenêtre.



Sélectionnez l'option 'Afficher des données d'une variable existante' et cliquez sur la flèche pour continuer.

Windev affiche les variables accessibles pour votre fenêtre. Ici le tableau que vous avez déclaré dans le code d'initialisation de votre fenêtre 'Gbw_TableauClients'. Sélectionnez-le et cliquez sur la flèche pour continuer.



Ici ca devient intéressant ! Windev a bien détecté que le tableau mémoire était un tableau d'objet cl_Client. Il nous propose donc de créer des colonnes de table et de les lier aux propriétés de ces objets. Ajoutez toutes les propriétés dans l'ordre que vous souhaitez. Cliquez ensuite sur le bouton de validation pour valider et terminer l'assistant (il n'y a rien à modifier dans les autres panneaux de l'assistant).

Windev a générer la table automatiquement. Modifiez manuellement les informations de la table pour être en phase avec ce tutoriel.

Nom Colonne	Titre	Type
CCOL_Nom	Nom	Texte
CCOL_Prenom	Prénom	Texte
CCOL_DateNaissance	Date naissance	Date
CCOL_Statut	Statut	Entier

Adaptez la taille de la table et des colonnes à votre convenance.
C'est terminé mais votre table reste désespérément vide !!

BOUTON 'AJOUTER UN CLIENT'

Ajoutez un bouton dans votre fenêtre.
Le code à écrire doit faire plusieurs choses.

1) Ajouter un objet cl_client dans le tableau mémoire
2) Afficher la table windev liée à ce tableau

Clic sur **Ajouter_un_client**

```
MonNouveauClient est un cl_client()                    // Création d'un objet client
MonNouveauClient:p_Nom="Nouveau"                       // On initialise le nom
TableauAjoute(Gbw_TableauClients,MonNouveauClient)     // On ajoute l'objet dans le tableau en mémoire.
TableAffiche(Table_Gbw_TableauClients,taInit)          // On demande à la table windev de se réafficher.
```

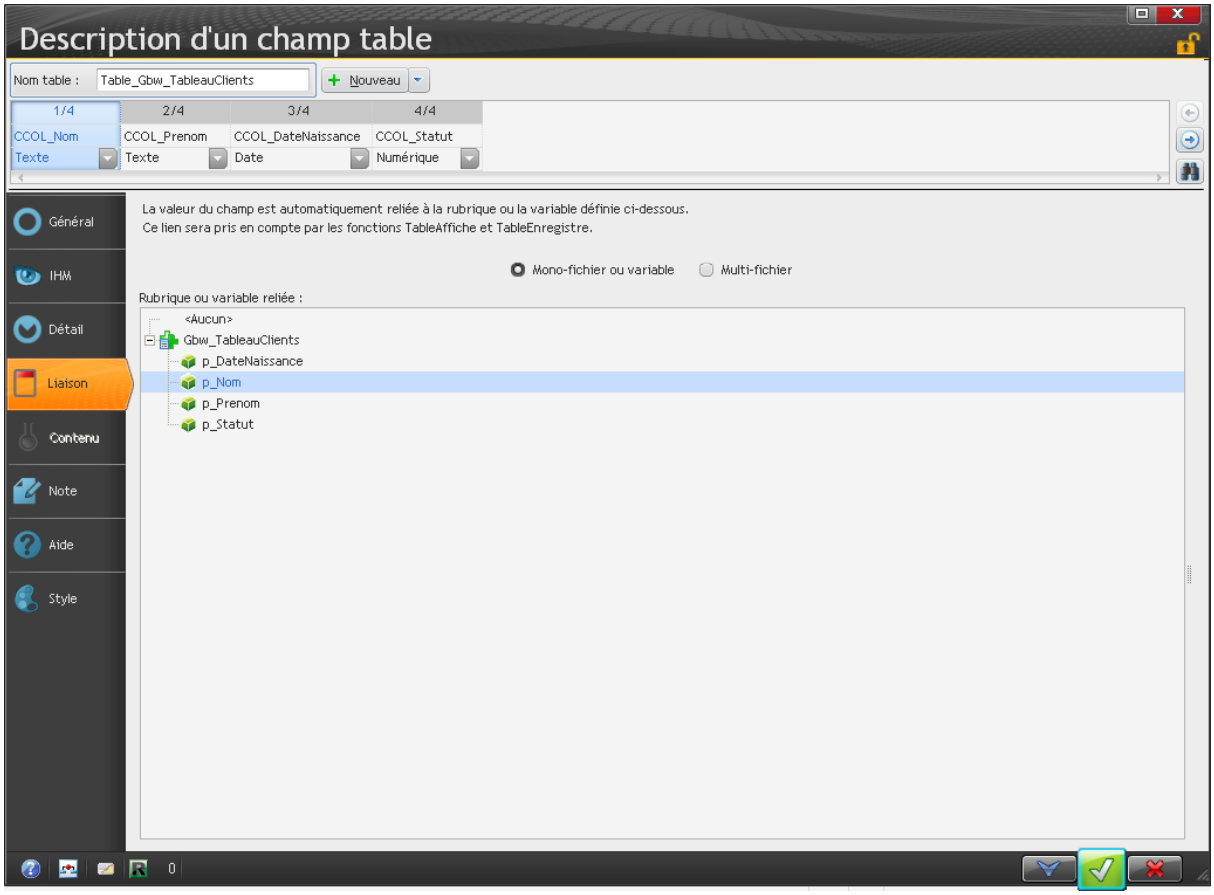
Ce code n'est pas la meilleure solution mais la plus compréhensible.

Lancez votre fenêtre en mode test et cliquez sur le bouton autant de fois que vous le souhaitez. Vous devez voir la table se remplir de lignes.

A ce stade vous avez vu comment ajouter des objets dans un tableau et comment afficher ce tableau dans une table windev.

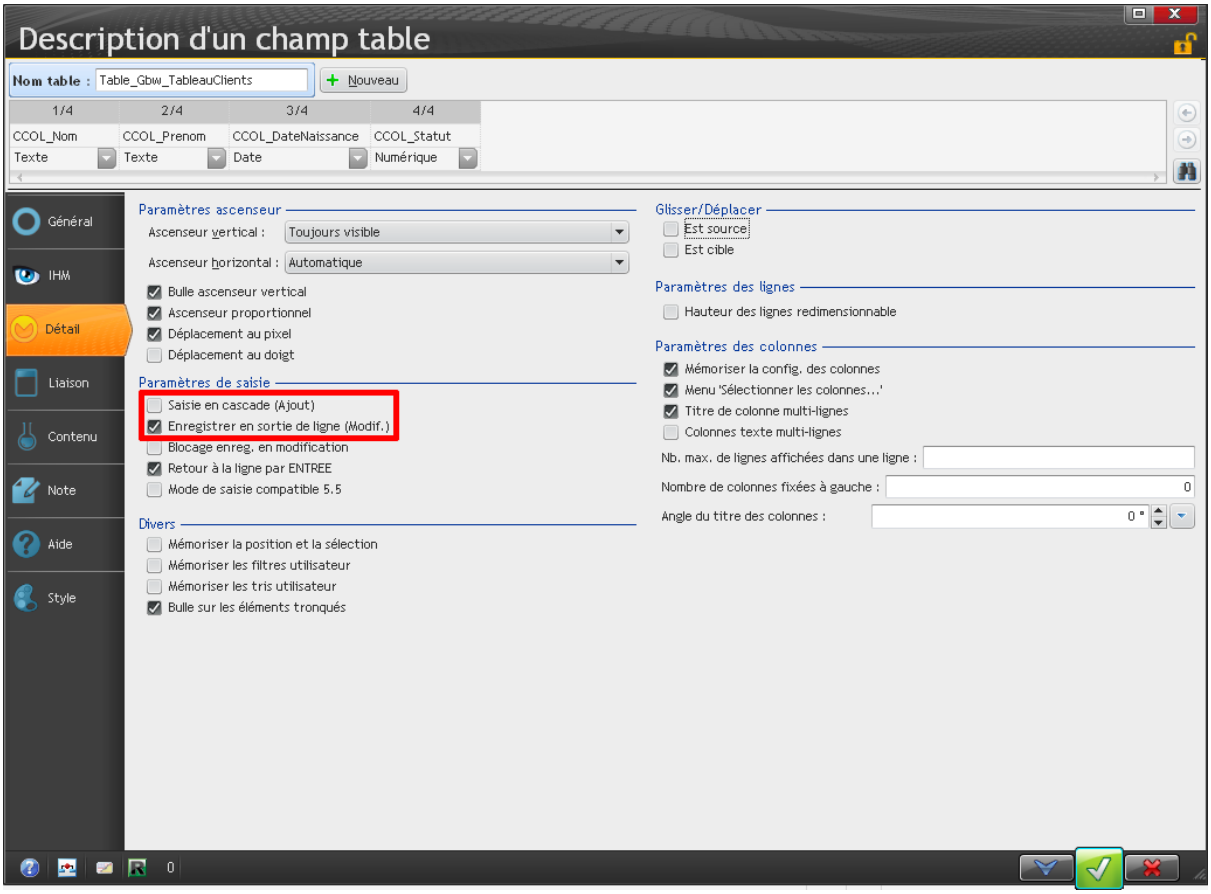
MODIFIER LES PROPRIETES DES OBJETS DEPUIS LA TABLE WINDEV

Pour modifier les propriétés des objets ce n'est pas très compliqué. Rappelez-vous, les colonnes sont déjà liées aux propriétés des objets !! La table windev va se charger toute seule d'envoyer les informations saisies dans les cellules (propriétés) des différents objets (1 par ligne de la table).



Par exemple ici nous voyons que la colonne CCOL_Nom est reliée à la propriété p_Nom des objets du tableau Gbw_TableauClients. Lorsque l'utilisateur va sortir d'une ligne, windev va envoyer la valeur de toutes les colonnes vers les propriétés correspondantes de l'objet ET ré-afficher ces valeurs. Il y a un aller retour de l'information. Par exemple pour la colonne CCOL_Nom, si vous saisissez une chaine en minuscules, votre propriété va automatiquement la convertir en majuscules et ré-afficher le résultat lorsque vous changerez de ligne (option Enregistrer en sortie de ligne).

Vérifiez juste que la table est bien paramétrée pour enregistrer les modifications en sortie de ligne et que la table est en saisie. N'autorisez pas la saisie en cascade pour empêcher l'ajout de nouvelles lignes automatiquement par windev.



Lancez la fenêtre et ajoutez des clients avec le bouton. Tentez de modifier un nom en minuscules et de saisir une date de naissance supérieure à la date du jour et passez sur une autre ligne… Vous pourrez constater que vos propriétés jouent bien leur rôle et contrôle bien les données affectées aux colonnes.

AJOUTER DES PROPRIETES ET DES COLONNES DANS LES TABLES

Vous avez peut être l'habitude (mauvaise !) de remplir certaines de vos tables avec un TableAjoute([NomTable],COL1+tab+COL2…) ce mode est très difficile à gérer car si vous remplissez plusieurs fois des tables avec une même procédure il faut intercaler la nouvelle données dans la procédure à la bonne position et il faut modifier toutes les tables et ajouter la nouvelle colonne exactement à la même position sous peine d'avoir des données dans les mauvaises colonnes.. Je suppose que vous connaissez le problème inhérent à cette méthode de gestion de table.

Avec la méthode de tableau et du databinding, il 'suffit' d'ajouter une propriété dans l'objet et de créer une nouvelle colonne liée à cette propriété !

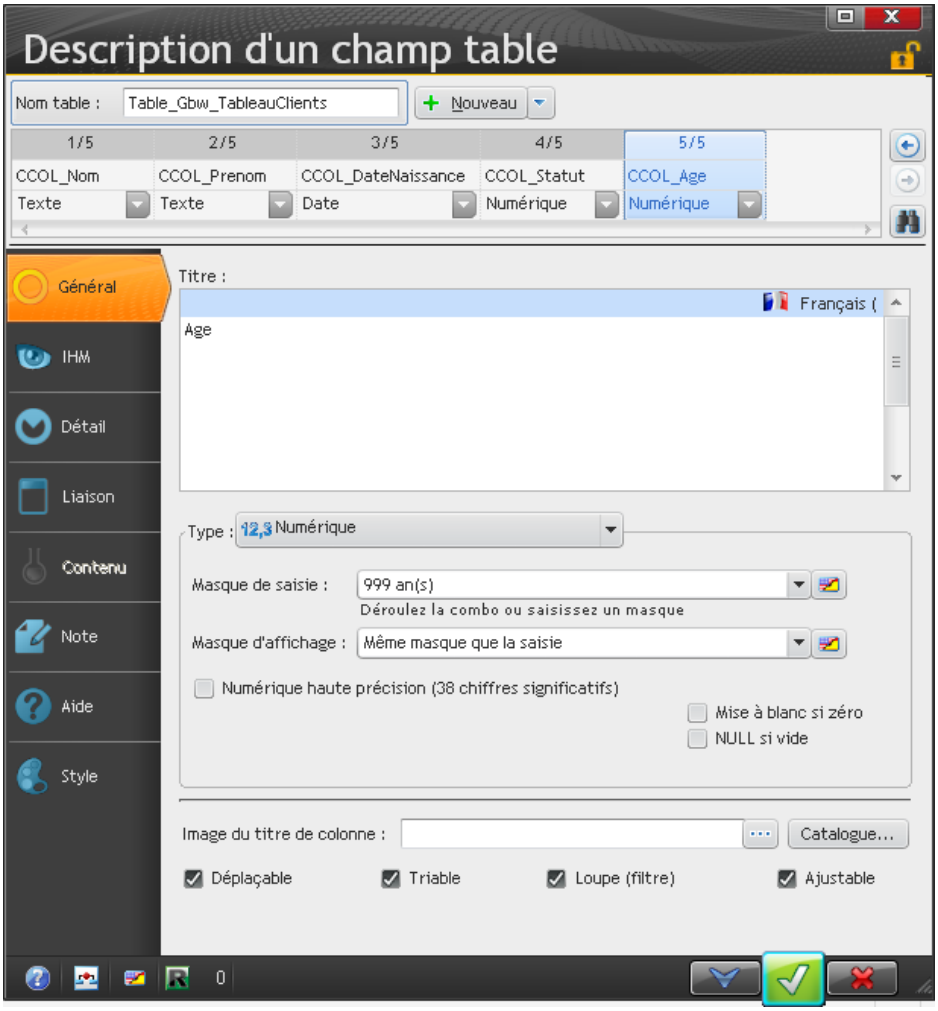
Par exemple , nous souhaitons afficher l'age du client

1) Ajoutez une nouvelle propriété p_Age dans la classe cl_client et saisissez le code comme ci-dessous

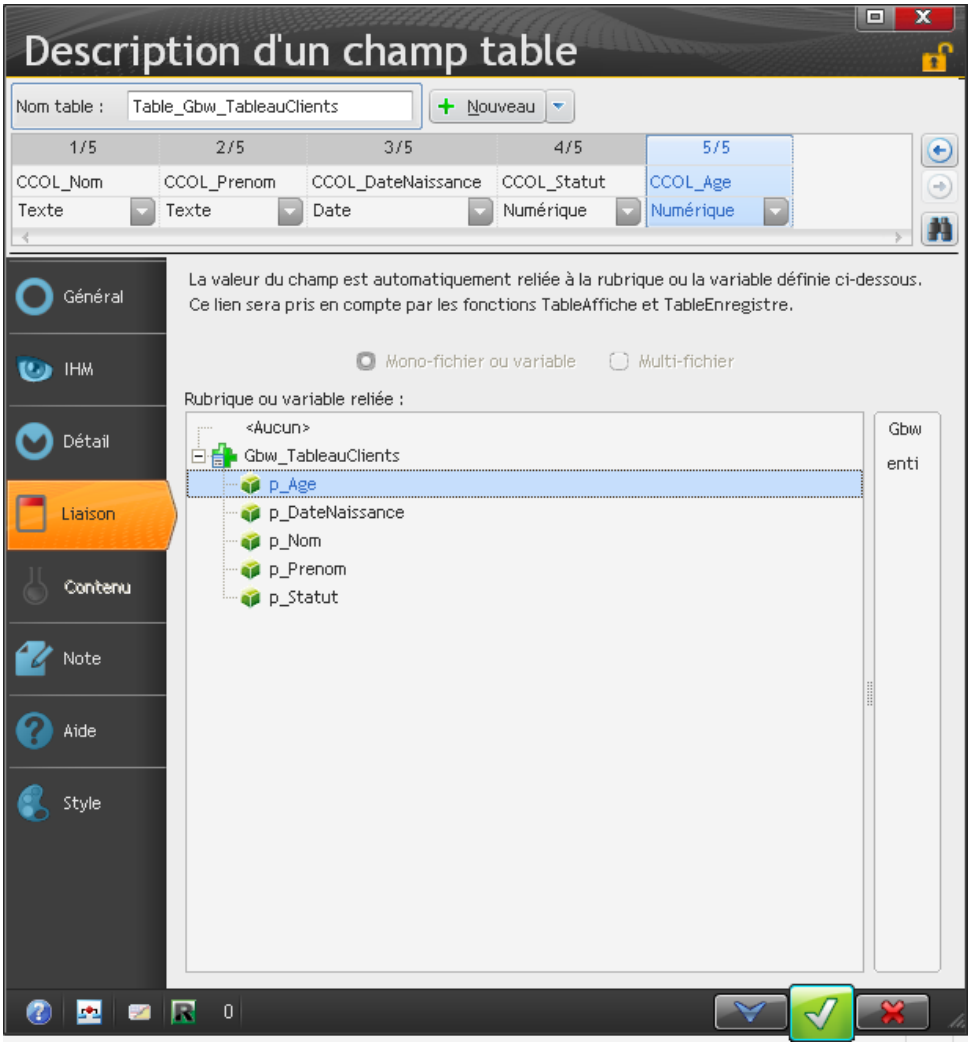
```
Récupération de la propriété p_Age
PROCEDURE p_Age()

  LOC_Age est un entier = Val(Gauche(Age(:p_DateNaissance),4))
  RENVOYER LOC_Age

Affectation de la propriété p_Age
PROCEDURE p_Age(Valeur)
```

La propriété est en lecture seule, il sera possible de connaitre l'âge du client mais il ne sera pas possible de le modifier. Par contre la modification de la date de naissance par la propriété p_DateNaissance reste évidemment possible. Vous pourriez ajouter du code dans l'affectation et tenter de calculer la date de naissance approximative par rapport à un âge.

2) Retournez dans la description de la table écran windev et ajoutez une nouvelle colonne 'CCOL_Age' de type numérique avec le format 999 an(s)



3) Dans l'onglet Liaison de la colonne, sélectionnez la nouvelle propriété



4) Testez votre fenêtre, l'âge doit se calculer et s'afficher lorsque vous modifiez une date de naissance et que vous sortez de la ligne (enregistrement en sortie de ligne)

Voila, vous avez votre objet, un tableau d'objets et vous savez comment l'afficher dans une table.

DIVERSES MANIPULATIONS POUR L'EXEMPLE

FICHIERVERSTABLEAU

Vous connaissez la méthode FichierVersTableau ? Non ou vous ne savez pas comment et pourquoi l'utiliser ! Je vous donne un bout de code que vous devriez tester…

1) Créez un fichier Client dans votre projet avec les rubriques Nom,Prenom,DateNaissance et Statut. L'ordre n'est pas important mais le nom doit correspondre exactement au nom que vous avez dans votre classe cl_client
2) Ajoutez plusieurs clients dans ce fichier avec WdMap ou autre méthode
3) Ajouter un bouton dans votre fenêtre avec le code ci-dessous et tester !

```
Clic sur Charger_le_fichier_dans_le_tableau
FichierVersTableau(Gbw_TableauClients,FIC_Clients)
TableAffiche(CTBL_Clients,taInit)
```

La première ligne charge tout le fichier dans le tableau d'objets ! Windev va automatiquement générer 1 objet par enregistrement et transférer le contenu des rubriques dans les membres des objets. Il se base sur le nom des rubriques et des membres !

La deuxième ligne demande à windev d'afficher la table liée au tableau d'objets. Ces deux lignes peuvent donc désormais être utilisées partout dans votre projet lorsque vous avez besoin d'afficher tous vos clients dans une table, une liste ou une zone répétée !

FichierVersTableau fonctionne également avec une requete ! Vous pouvez donc faire exactement le même code avec une requête créée dans l'éditeur ou une requete SQL directement dans le code.

```
Clic sur Charger_le_fichier_dans_le_tableau *
LOC_SourceDonnees est une Source de Données
SI HExécuteRequêteSQL(LOC_SourceDonnees,hRequêteDéfaut,"SELECT * FROM FIC_Clients WHERE Nom Like '%D%'") ALORS
    FichierVersTableau(Gbw_TableauClients,LOC_SourceDonnees)
    TableAffiche(CTBL_Clients,taInit)
    HAnnuleDéclaration(LOC_SourceDonnees)
SINON
    Erreur(HErreurInfo(hErrMessage))
FIN
```

FONCTIONNALITE POUR SELECTIONNER DES CLIENTS

Lorsqu'on affiche des informations dans une table on veut pouvoir sélectionner des lignes soit par la fonctionnalité proposée par les tables multi-sélection mais également par une case à cocher.

L'approche objet/tableau rend cette possibilité très simple car il suffit d'ajouter une propriété de sélection dans la classe cl_client !

Ajoutez le membre 'Selection' dans votre classe cl_client

```
cl_client est une Classe
    PRIVE
        Nom est une chaîne
        Prenom  est une chaîne
        DateNaissance est une Date
        Statut est un entier
        Selection est un booléen
FIN
```

Ajoutez ensuite la propriété correspondante :p_Selection dans la classe cl_client

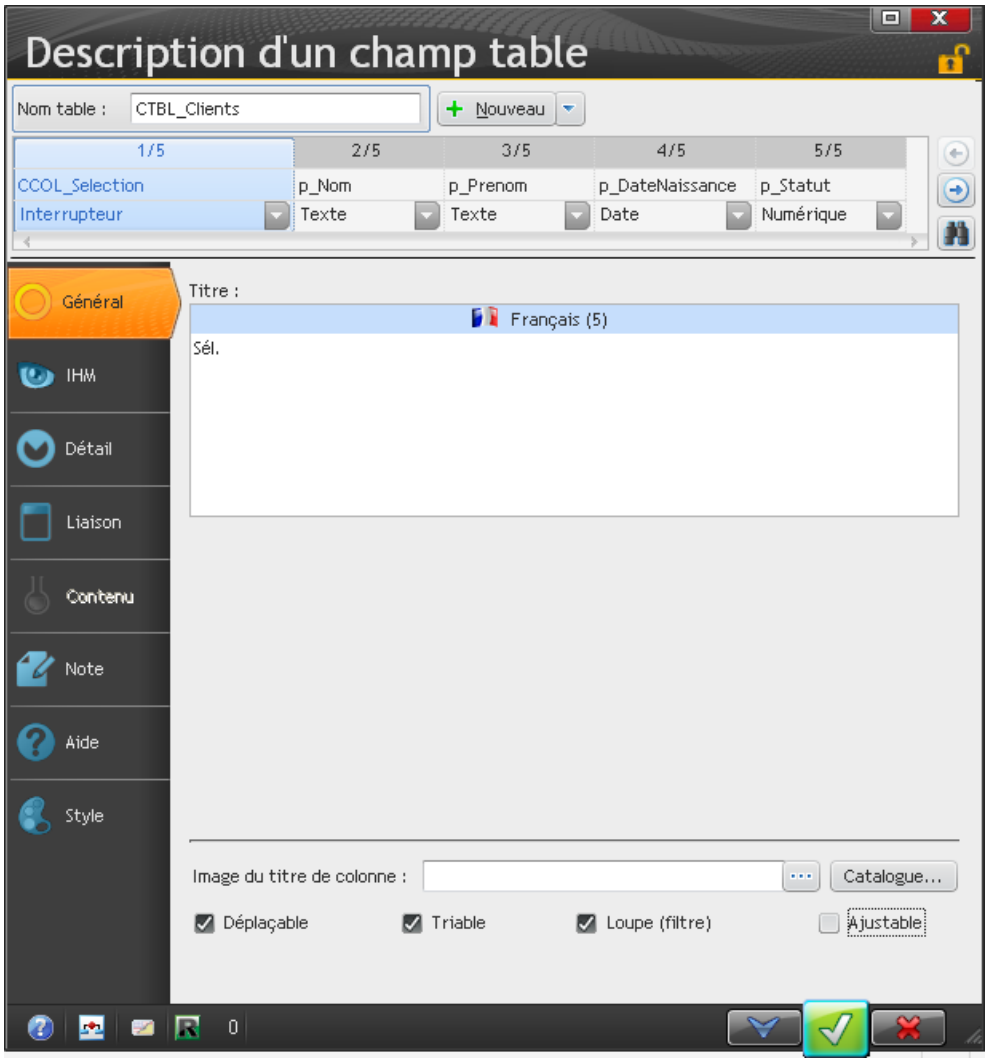
Récupération de la propriété **p_Selection**

```
PROCEDURE p_Selection()

renvoyer :Selection
```

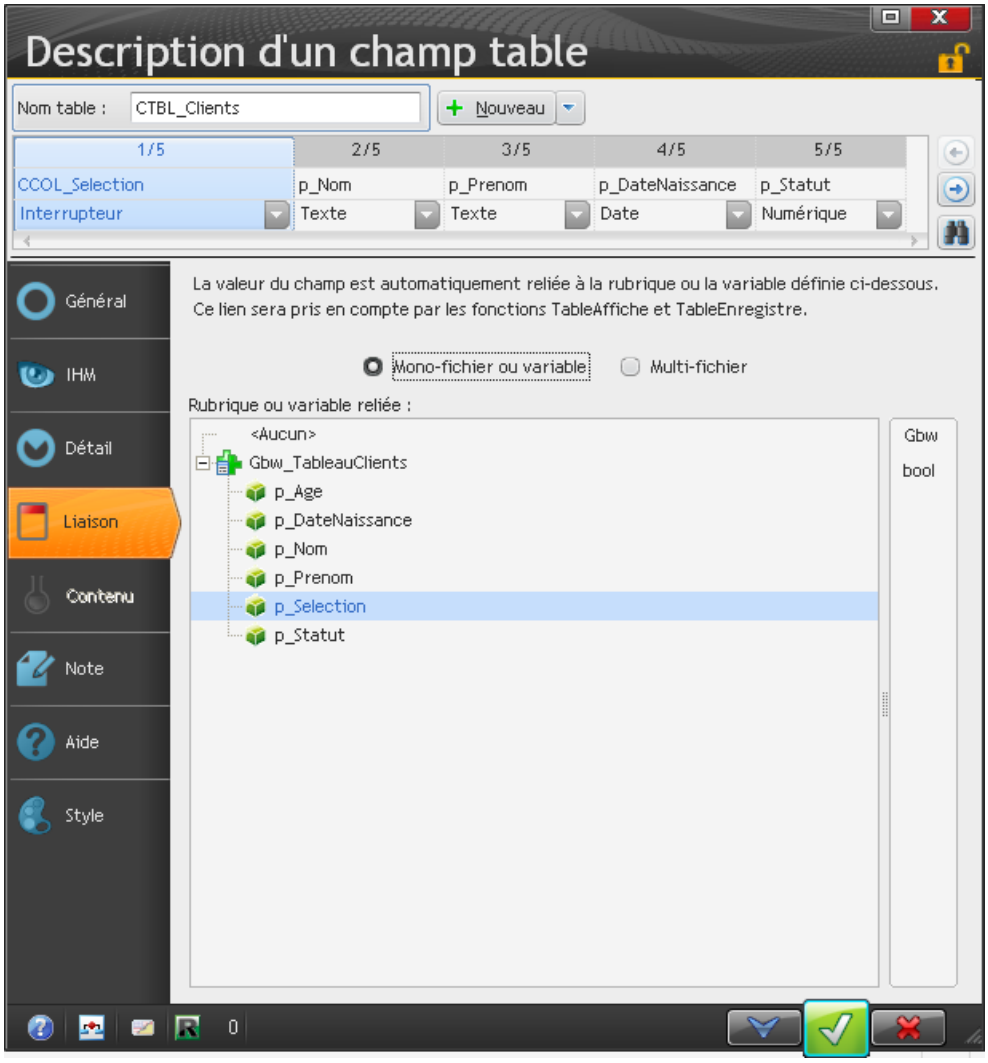
Affectation de la propriété **p_Selection** *

```
PROCEDURE p_Selection(Valeur est un booléen)

:Selection=Valeur
```

Retournez dans la fenêtre et ajoutez une nouvelle colonne dans votre table a la position que vous souhaitez.



Liez cette nouvelle colonne à la nouvelle propriété.



Testez votre fenêtre.. Bon ok ca marche mais on ne se rend pas bien compte si les informations sont mémorisées. Ajoutez un nouveau bouton dans votre fenêtre (CBTN_Selection) avec le code ci-dessous et testez votre fenêtre. Sélectionnez des clients et cliquez sur ce nouveau bouton.

Clic sur **CBTN_NombreSelection**

```
LOC_Texte est une chaîne
POUR TOUT LOC_Client DE Gbw_TableauClients
    SI LOC_Client:p_Selection ALORS
        LOC_Texte+=[RC]+LOC_Client:p_Nom+"-"+LOC_Client:p_Prenom
    FIN
FIN
Info(LOC_Texte)
```