Manuel d'auto-formation
Cyril Beaussier

Passer de Windev à Delphi

Base de données - Débuter avec l'ADO

Version 1.0 – Septembre 2003

COPYRIGHT ET DROIT DE REPRODUCTION

Vous avez acquis un droit d'utilisation de ce manuel dans un cadre privé. Cependant si vous l'utilisez au sein d'une entreprise ou dans un but lucratif, je vous saurai gré de me faire parvenir un chèque de 8,00 € libellé à l'ordre de :

Cyril Beaussier
4, rue de Paris
77200 TORCY – FRANCE

Une facture vous sera envoyée en retour sur simple demande écrite.

Aucune partie de ce support ne peut être reproduite ou transmise à quelque fin ou par quelque moyen que ce soit, électronique ou mécanique, sans la permission expresse et écrite de son auteur.

Si vous souhaitez des améliorations, je suis évidemment ouvert à toute suggestion. Il en est de même si vous constatez une erreur (nul n'est parfait). Pour cela, il suffit de m'écrire avec pour sujet « *Passer de Windev à Delphi / Base de données - Débuter avec l'ADO* » dans la rubrique « *Contact* » de mon site principal :

www.beaussier.com

Les marques et noms de société cités dans ce support sont déposées par leurs propriétaires respectifs. Delphi est la propriété exclusive de BORLAND. Windev est la propriété exclusive de PC SOFT. Windows et ACCESS sont la propriété exclusive de Microsoft Corporation.

Je ne suis lié avec aucun éditeur ou constructeur informatique.

Ce support a été réalisé avec la suite bureautique libre *Open Office* 1.1 (disponible gratuitement sur http://fr.openoffice.org)

Avertissement complémentaire :

Les éléments (données ou formulaires) éventuellement inclus dans ce support vous sont fournis à titre d'exemple uniquement. Leur utilisation peut avoir, dans certains cas, des conséquences matériels et juridiques importantes qui peuvent varier selon le sujet dont ils traitent. Il est recommandé d'être assisté par une personne compétente en informatique ou de consulter un conseiller juridique ou financier avant de les utiliser ou de les adapter à votre activité.

Sommaire

1. Introduction

Ce support traite de l'accès à une base de données MS-Access depuis Delphi via le système ADO. Ayant démarré en janvier 2003 sous cet AGL, j'ai eu tout de suite besoin de développer ce type d'application.

Vous trouverez donc à travers ces pages, un didacticiel rapide traitant de ce sujet et vous allez découvrir qu'avec la puissance des composants ADO, il n'y a *presque* pas besoin de programmer.

Avec ce manuel, vous n'avez pas besoin de connaître le langage Pascal Objet. Néanmoins, je vous recommande la lecture de « *Passer de Windev à Delphi* » (disponible sur ce même site) pour l'utilisation de Delphi lui-même ainsi que pour toute la terminologie par rapport à Windev et au *W-Langage*.

Il y a plus de dix ans, j'ai eu à faire le choix d'un AGL pour le développement d'application tournée vers la gestion. A l'époque Delphi ne proposait que le BDE qui était particulièrement lourd et contraignant en terme d'installation chez le client. Mon choix s'était alors porté sur Windev qui était très simple pour l'accès à des bases SQL.

Avec l'arrivée des versions 6 et 7 de Delphi et ses composants d'accès aux données (ADO et dbExpress), je me suis à nouveau intéressé à cet AGL.

Note :

BORLAND propose Delphi 7 en version d'évaluation. Vous pouvez donc tout à fait expérimenter l'exemple de ce manuel.

2. Concept

Windev propose différents accès aux bases de données tierces (autre que *Hyper File*). Il s'agit des accès ODBC ou par un fournisseur OLE DB[1]. C'est avec ce dernier que nous allons utiliser ce que l'on appelle chez Microsoft l'accès ADO.

2.1. Un mot sur l'ADO

L'ADO qui signifie en clair « ActiveX Data Objects », est un système d'accès aux données purement Microsoft dit de haut niveau. C'est-à-dire que la complexité de la connexion et des opérations de lecture écriture est masquée pour le développeur qui l'utilise.

Le système ADO a donc l'énorme avantage, si vous désirez vous connecter à des bases Microsoft comme Access ou SQL/Server, d'avoir ses bibliothèques en standard sur la plupart des versions récentes de Windows (98/2000/XP). Ce qui facilite le déploiement en clientèle de vos applications.

Note :

Pour des versions Windows comme 95 ou NT, vous aurez probablement besoin d'avoir recours au moteur ADO qui est distribué sous la forme du MDAC (reportez-vous au chapitre *3.2. Connexion*).

2.2. Limite du support

Pour une bonne compréhension de ce support (avec notamment les copies écran), je pars du principe que vous utilisez Delphi dans sa version 7 et MS-Access dans une version 97 minimum.

La version de Windows n'a pas d'importance bien que mon ordinateur soit sous une antique version 98SE. 

En revanche, assurez-vous de disposer d'un paquetage MDAC sur votre ordinateur. Reportez-vous pour cela au chapitre *3.2. Connexion*

1 Je vous renvoie sur la documentation Windev au mot clé ODBC.

3. Présentation

Afin de rester le plus simple possible, je décrirai dans ce document, une architecture en mode client/serveur avec le système de gestion de base de données de Microsoft Access.

En effet, MS-Access est certainement le SGBD le plus simple à mettre en place puisqu'il ne nécessite pas de serveur dédié et qu'il peut être installé sur la même machine que le programme client.

Du côté de Delphi, nous avons à notre disposition un éventail de composants qui vont nous permettre d'utiliser le mode de connexion ADO.

	Composant	Description
	`TADOConnection`	Permet la connexion à la base et assure le support des transactions.
	`TADOCommand`	S'utilise principalement pour exécuter des commandes SQL qui ne renvoie pas d'ensemble de résultats[2].
	`TADODataSet`	Accès aux tables de la base avec retour d'information possible.
	`TADOTable`	Accès physique à une table avec retour d'information possible.
	`TADOQuery`	Permet l'envoi de requête SQL avec retour d'information.
	`TADOStoredProc`	Exécute les procédures stockées d'une base.
	`TRDSConnection`	Permet le transfert de données avec possibilité de triage.

2 La documentation Delphi précise qu'avec la méthode Execute, un `TADOCommand` est capable de renvoyer un ensemble d'enregistrements. Mais pour l'utiliser, vous aurez besoin d'un composant ensemble de données ADO distinct.

4. Partie serveur

Nous allons maintenant procéder au développement d'une petite application. Celle-ci sera une gestion très simple d'un fichier de clients et que nous appellerons... **AdoPrimo**.

L'intégralité des sources est fournie avec ce support dans le même paquetage. Vous avez les sources des programmes, le programme compilé et la base de données Access.

Je vous encourage cependant, à ne vous en servir qu'en dernier recours. 

4.1. Base de données

D'abord, nous allons créer la structure de notre base de données. L'intérêt de MS-Access fait que nous n'avons qu'un seul fichier généré. Les tables comme les index sont en effet intégrés dans le fameux **.MDB**.

Notre base **Client** se compose d'une table unique du même nom et dont la structure est la suivante :

Client		
Id	Numéro automatique	Numéro du client
Nom	Chaîne (50)	Nom du client
Adr	Chaîne (200)	Adresse
Cp	Chaîne (5)	Code postal
Ville	Chaîne (50)	Ville
Tel	Chaîne (25)	Téléphone

Pour bien faire les choses, n'oublions pas les index sur les tables. La colonne `Client.Id` étant déclarée en tant que clé primaire, elle sera ainsi automatiquement indexée[3].

3 La documentation MS-Access précise qu'un index est automatiquement créé sur un champ déclaré en tant que clé primaire.

4.2. Connexion

Avec Windev, si vous programmez avec les ordres `SQL...`, il est courant pour établir un point de connexion sur une base de données à travers l'ODBC, de créer ce que Microsoft appelle un DSN[4].

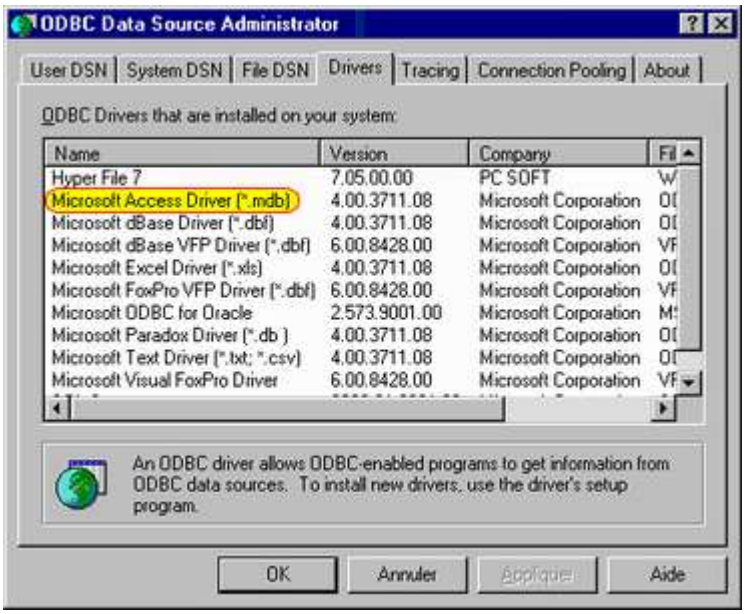
Mais ce n'est pas obligatoire et cela tend à ralentir la connexion. Donc dans ce support nous n'aborderons pas ce principe puisqu'avec Delphi, nous pouvons très simplement nous passer complétement de cette étape.

A noter :

Il ne faut pas confondre le DSN et le DNS (*Domain Name Server*). Ce sont deux choses complètement différentes.

L'utilisation d'un DSN est utile notamment pour permettre à l'utilisateur de modifier lui-même le chemin d'accès à la base MS-Access. Il est cependant contraignant car il vous faudra créer ce DSN sur chaque poste où vous installerez votre application.

Vérifiez que vous disposez bien de l'accès ODBC à MS-Access. Pour cela, allez dans le panneau de configuration Windows et lancez l'icône de l'**Administrateur ODBC**.



Dans l'onglet « Pilote », vérifiez que le pilote ODBC Microsoft Access est installé. S'il n'est pas dans la liste, vous devez **obligatoirement** l'installer.

4 Abréviation anglaise de *Data Source Name*

Reportez-vous sur le site MSDN de Microsoft en tapant le mot clé **MDAC** pour obtenir le paquetage nécessaire.

Attention cependant à la version que vous téléchargez. Le MDAC 2.7 n'inclut pas par exemple, le pilote *OLE DB Microsoft Jet* qui est nécessaire à notre exemple. Préférez plutôt une version 2.1.

A noter :

Si vous n'avez pas d'accès à internet, sachez que PC SOFT fournit sur le CD-ROM d'installation de Windev 7.x ce fameux paquetage MDAC.

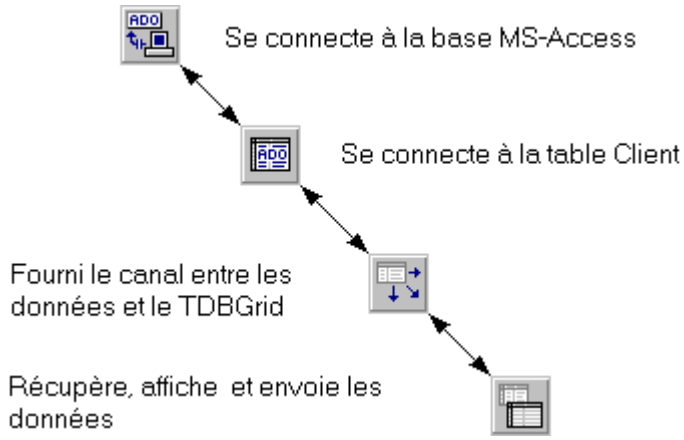
5. Partie client

Nous allons maintenant passer à la programmation de notre application **AdoPrimo**. Créez pour cela un dossier du même nom et dans lequel nous allons mettre l'ensemble des fichiers du projet.

Notre application va se construire en quatre étapes très simples :

- La définition de l'interface utilisateur avec la fiche principale.
- La mise en place des composants ADO pour l'accès aux données.
- La mise en place du composant d'interface entre l'ADO et l'interface utilisateur.
- La mise en place des composants de gestion des données pour l'affichage et les opérations d'écriture des données.

Nous allons ainsi avoir une hiérarchie de composants qui nous permet d'afficher et de modifier les données de notre table Access.



Chaque composant est un objet qui exporte ses propriétés vers un autre objet qui les récupérer. Nous allons voir cela en détail dans les chapitres suivants.

A noter :

Cette notion de composant (apparue sous Windev 7) est la base de toute la philosophie Delphi depuis ses débuts. Reportez-vous au manuel « *Passer de Windev à Delphi* » (disponible sur ce même site) pour plus de détails.

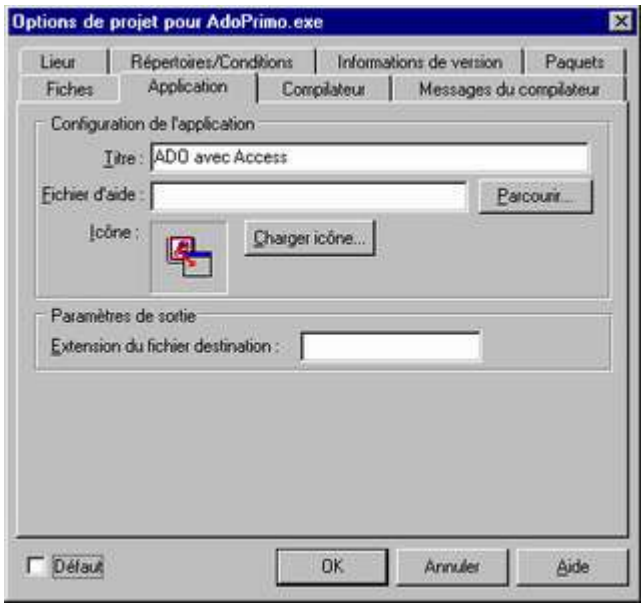
5.1. Structure de l'application

L'application ne comporte qu'une seule fiche que nous appellerons **fPrimo** et **uPrimo** pour son unité Pascal.

Une fois Delphi lancé, enregistrons notre fiche principale avec comme nom d'unité Pascal **uPrimo** et en changeant les propriétés suivantes pour la fiche :

- *Name* pour « **fPrimo** »
- *Caption* pour « **Gestion Client ADO** ».

Enregistrez enfin le projet sous le nom **AdoPrimo**. Dans le menu **Projet / Options**, donnez une jolie icône pour votre application.

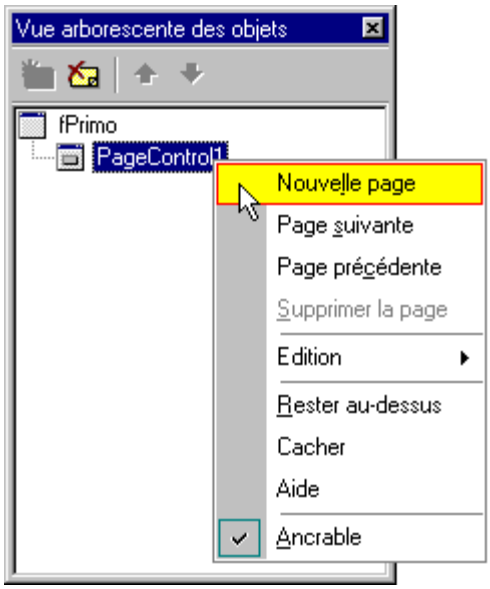


5.2. Interface utilisateur

Pour que notre interface soit agréable pour l'utilisateur, nous allons d'abord placer un composant **PageControl** sur la fiche.

PageControl se trouve sur la palette des composants dans l'onglet *Win32*...

Une fois le composant posé, dimensionnez le pour qu'il tienne sur la totalité de la surface de la fiche. Vous pouvez laisser le nom donné par défaut *PageControl1*, cela n'a pas d'importance.

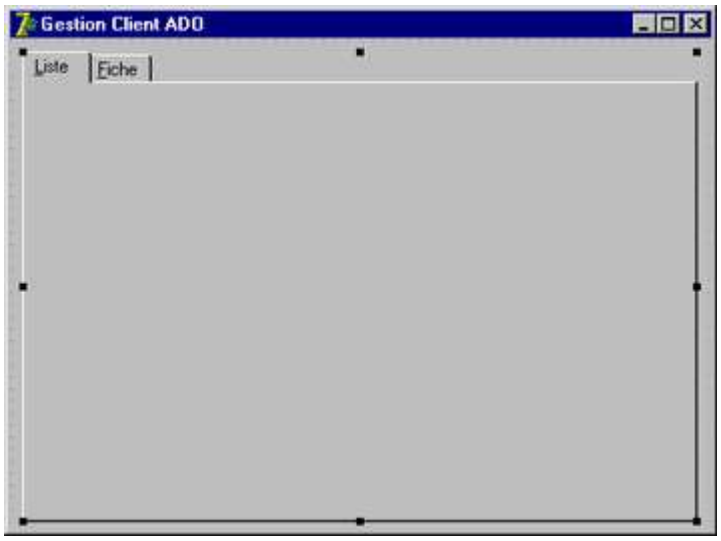


Dans la **Vue arborescente des objets**, faîtes un clic droit sur *PageControl1* et choisissez le menu **Nouvelle page** pour ajouter un onglet de feuille.

Renouvelez l'opération une seconde fois afin d'obtenir deux onglets attachés à *PageControl1* et nommés respectivement *TabSheet1* et *TabSheet2*.

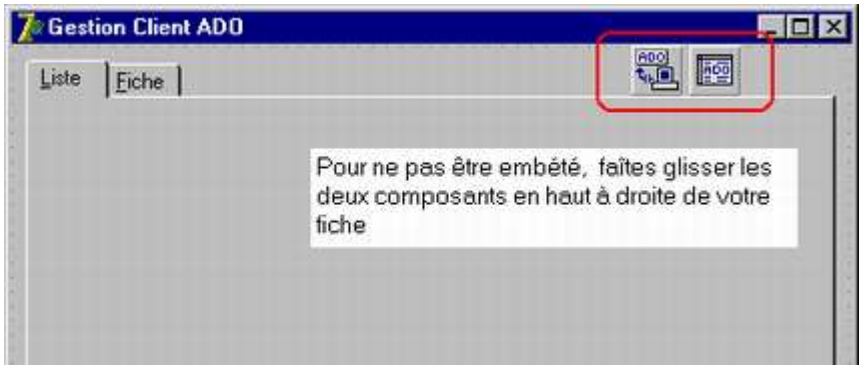
Changez les propriétés **Caption** de ces deux onglets pour "**&Liste**" et "**&Fiche**"

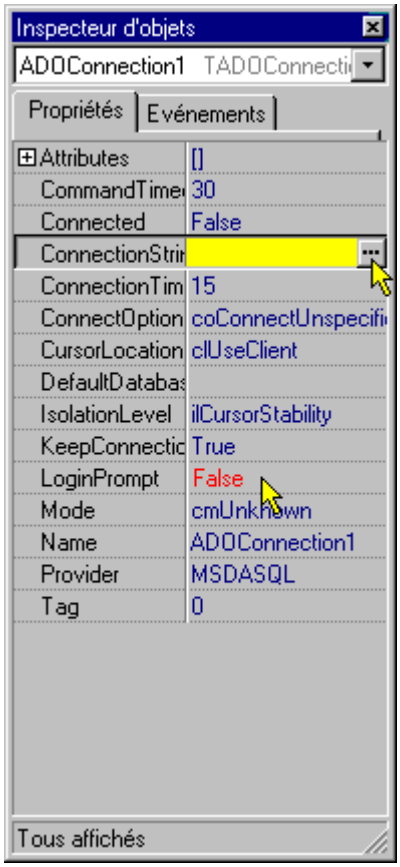
Votre fiche doit maintenant ressembler à l'image ci-dessous :



5.3. La couche ADO

Nous allons maintenant utiliser les composants ADO. Ce sont eux qui vont assurer la connexion à la base de données MS-Access. Depuis la palette, placez un composant **ADOConnection** et un composant **ADOTable**.





Nous allons ensuite procéder au paramétrage de la chaîne de connexion afin de déterminer le pilote de données à utiliser et le chemin d'accès à la base.

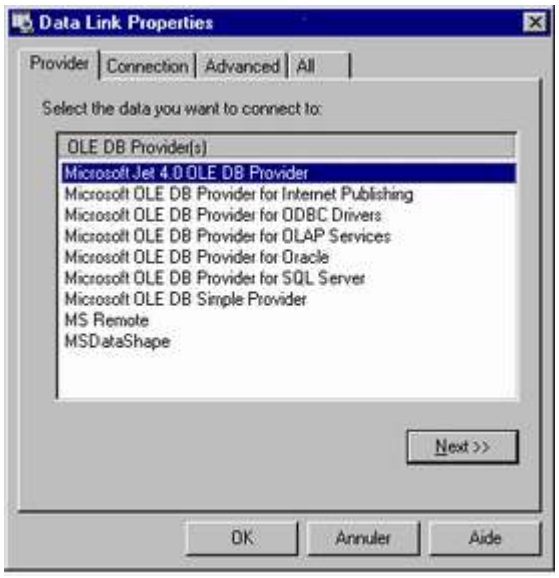
Depuis l'**Inspecteur d'objets**, sélectionnez dans la liste votre composant *ADOConnection1*.

Double cliquez sur la propriété **LoginPrompt** pour la mettre à **False** afin de ne pas avoir de demande d'identification utilisateur lors de la connexion à la base.

Cliquez sur le bouton de configuration de la propriété **ConnectionString** (comme illustré à gauche).

Delphi tout comme Windev est aussi largement pourvu d'Assistant pour vous aider et vous guider dans certaines tâches.

L'Assistant de connexion apparaît. Cliquez sur le bouton **Construire**.

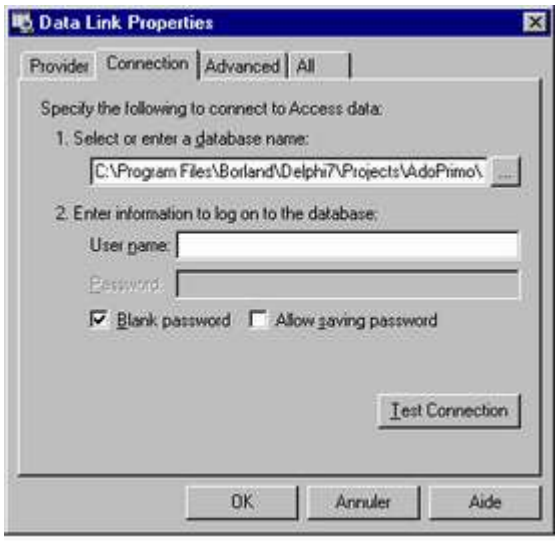


Sélectionnez le pilote **Microsoft Jet 4.0 OLE DB Provider**.

Cliquez sur le bouton **Next >>**

Rappel :

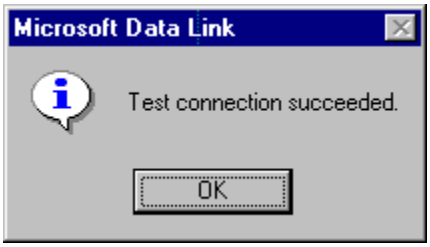
Si le pilote Fournisseur **Microsoft Jet 4.0 OLE DB** n'est pas dans cette liste, c'est que votre version de MDAC est certainement trop récente.



A l'étape 1, sélectionnez votre base de données **Client.mdb**. En principe, vous êtes déjà dans le bon répertoire.

A l'étape 2, inutile de supprimer le nom d'utilisateur (*User name*) : **Admin** si vous laissez cocher la case « Blank password » (mot de passe vide).

Enfin, cliquez sur le bouton **Test Connection** pour vérifier que la connexion est correcte. Et en principe, vous devez voir s'afficher :

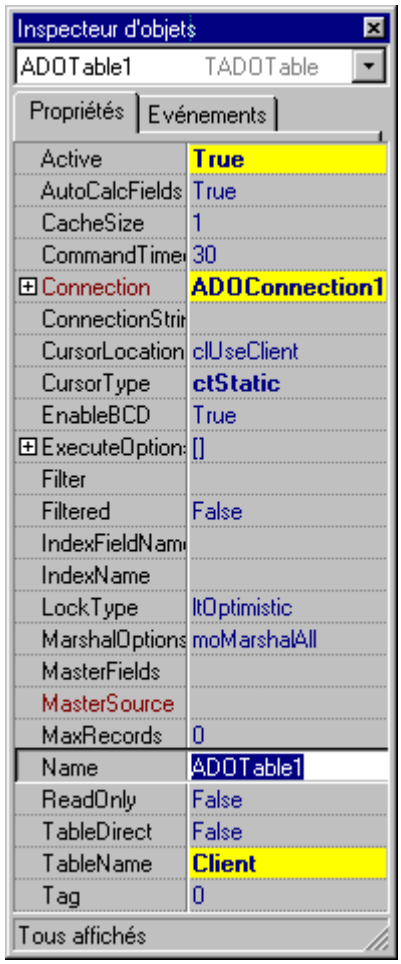


Passons à notre composant *ADOTable1*.

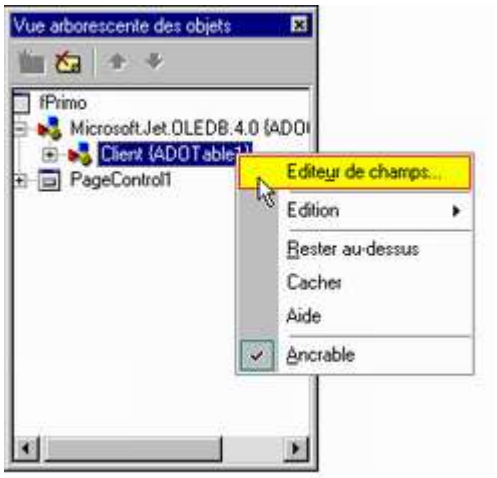
Toujours depuis l'**Inspecteur d'objets**, modifiez les propriétés suivantes :

1. Sélectionnez le composant *ADOConnection1* dans la liste de la propriété **Connection**.
2. Sélectionnez la table **Client** dans la liste de la propriété **TableName**.
3. Passez à la valeur **True** dans la propriété **Active**.

Vous devez maintenant en être au même point que dans l'illustration de droite.



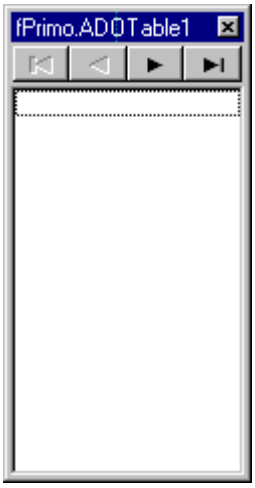
Voilà, nous venons de connecter notre composant de connexion à la base avec notre composant Table. Nous allons maintenant paramétrer ce dernier avec les colonnes de la base Access qui nous intéresse.



Passez à nouveau dans la fenêtre **Vue arborescente des objets**.

Faîtes un clic droit sur le composant *ADOTable1* et sélectionnez le menu **Editeur de champs**

Une petite fenêtre s'ouvre comme ci-dessous.



Cette petite fenêtre va nous permettre d'insérer les colonnes de la table **Client** qui vont nous servir ensuite pour l'affichage des données.

Appuyez sur les touches **Ctrl** + **A** pour afficher la liste des champs disponibles.

Par défaut, tous les champs de la table sont sélectionnés.

Enlevez le champ **Id** qui n'a pas besoin d'être affiché puisqu'il s'agit de la **clé primaire** de la table.

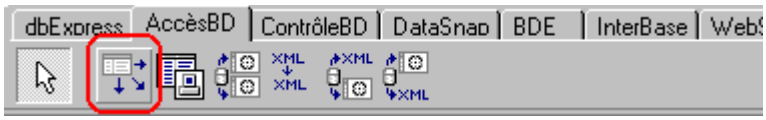
Cliquez sur le bouton **OK** pour valider

Voilà notre couche ADO est paramétrée.

5.4. Accès aux données

Nous allons utiliser un composant **DataSource** pour nous permettre de faire le lien entre les composant ADO d'accès aux données et les contrôles d'affichage de données de notre fiche.

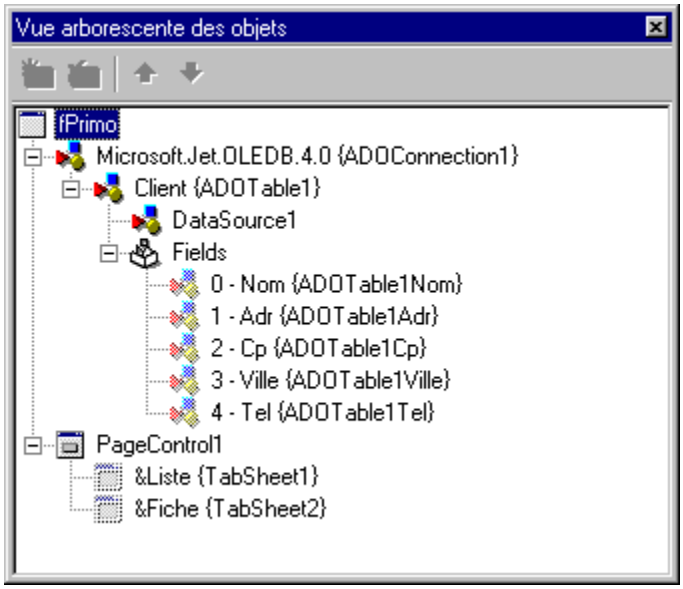
Ce composant se trouve sur l'onglet **Accès BD**



Comme pour les deux composants ADO, placez le **DataSource** en haut à droite pour ne pas être gêné.

Dans l'**Explorateur d'objets**, placez-vous sur *DataSource1*. Mettez-vous sur la propriété **DataSet** et sélectionnez dans la liste *ADOTable1*.

Votre **Vue arborescente des objets** doit maintenant afficher la hiérarchie des composants comme dans l'illustration ci-dessous.

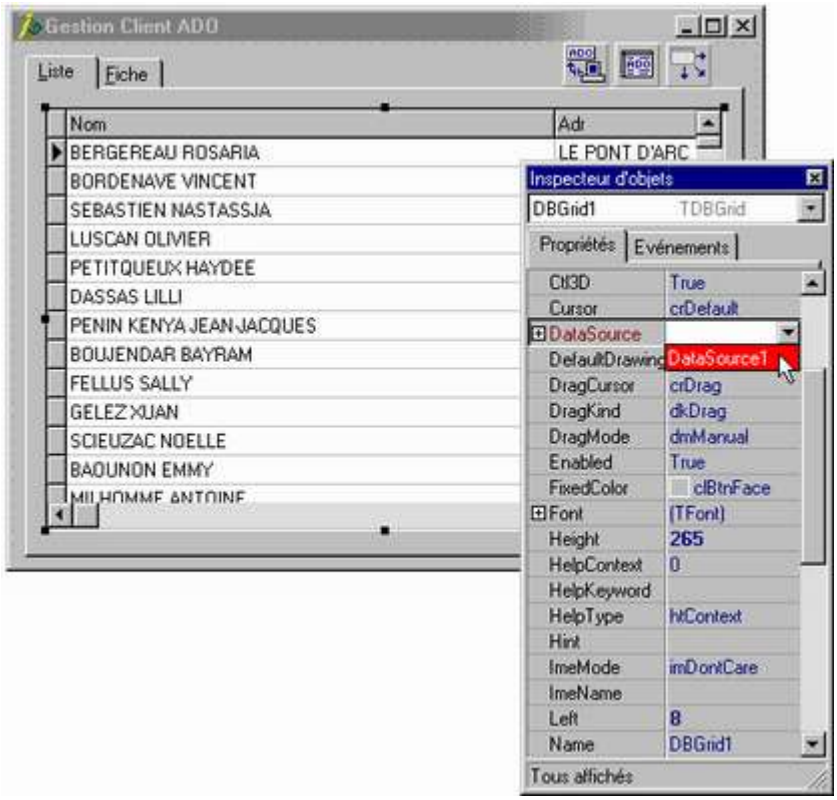


5.5. Affichage des données

Placez maintenant sur votre fiche et sur l'onglet **Liste**, un composant *DBGrid*.

Agrandissez votre composant *DBGrid1* aux dimensions maxima de l'onglet.

Placez-vous dans l'**Explorateur d'objets** et sélectionnez dans la propriété **DataSource**, le composant *DataSource1*.



L'affichage de vos données apparaît alors dans la grille.

Note :

Pensez à régler la largeur des colonnes de la grille en agissant sur la propriété **Width** depuis la **Vue arborescente des objets**.

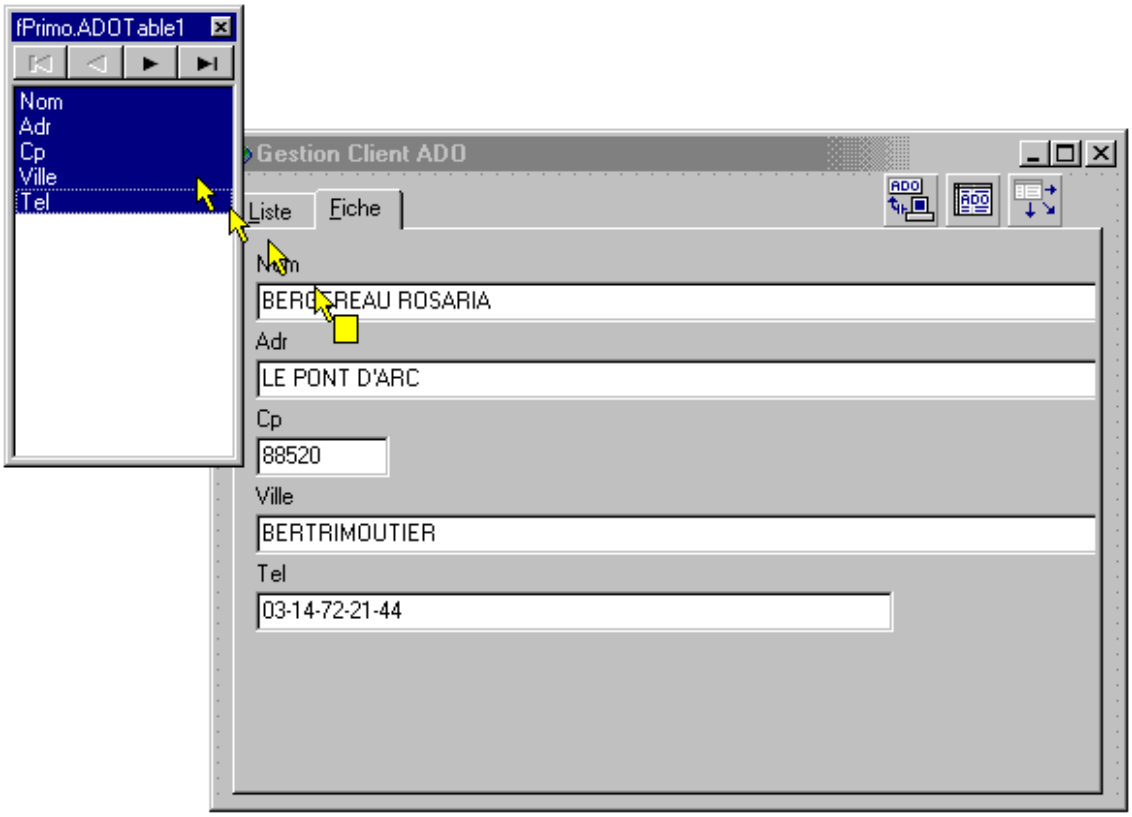
5.6. Modification des données

Le composant **DBGrid** est généralement utilisé pour avoir une vision globale des enregistrements. Il est en revanche peu pratique pour la modification d'une ligne d'informations.

Nous allons donc passer sur l'onglet **Fiche** de notre fenêtre pour créer des champs spécifiques à chaque colonne et rendre plus aisé la mise à jour.

Vous allez voir qu'avec Delphi, la création de ces champs va être d'une extrême simplicité.

Dans la **Vue arborescente des objets**, double cliquez sur le composant *ADOTable1* pour faire apparaître la fenêtre des champs. Faîtes alors glisser l'ensemble des champs sur la fiche comme dans l'illustration ci-dessous.



Note :

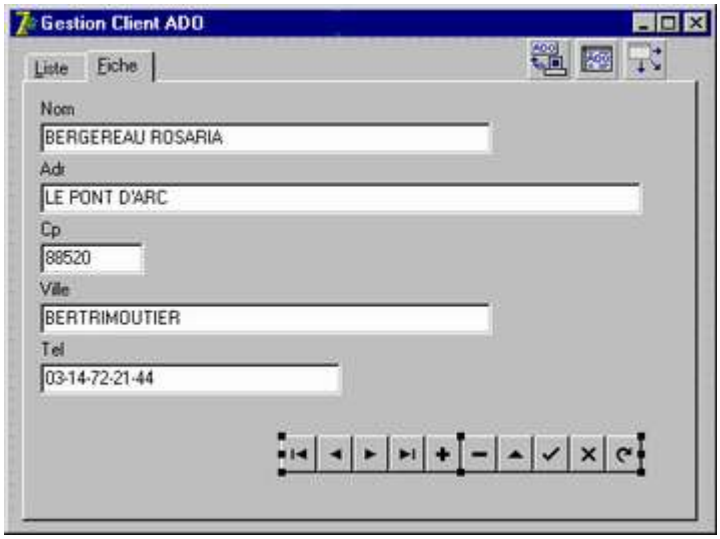
Pensez à régler la largeur des champs en agissant sur la propriété **Width** depuis l'**Inspecteur d'objet**. De la même manière, personnalisez les étiquettes des champs avec la propriété **Caption**.

L'onglet Fiche est quand même peu pratique si l'on veut naviguer d'un enregistrement à un autre. De plus, il nous manque une fonction de création et de suppression.

Nous allons régler ce problème avec le composant **DBNavigator** disponible sur l'onglet **Contrôle BD**.

Placez le comme sur l'illustration.

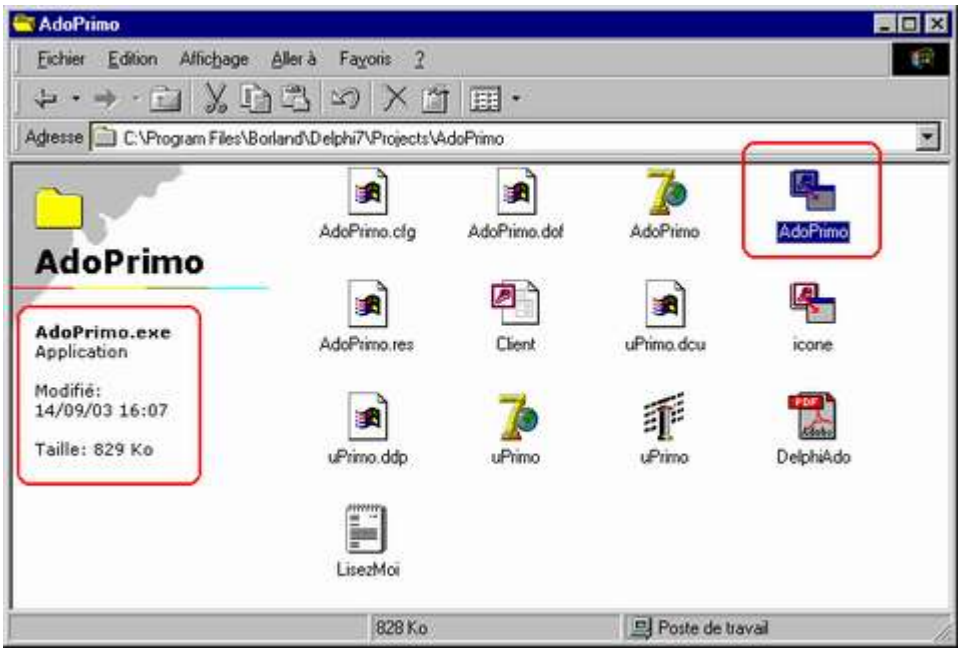
N'oubliez pas de sélectionner dans la propriété **DataSource**, votre composant *DataSource1* sinon les boutons seront inactifs.



Note :

Vous pouvez également paramétrer les boutons du composant **DBNavigator** qui seront ou non visibles dans votre fiche.

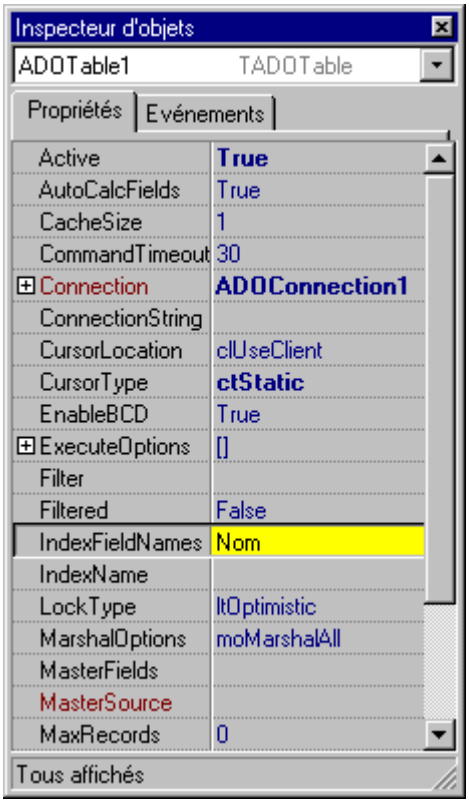
Voilà, l'application est prête. Il nous reste à créer l'exécutable directement en testant par la touche **F9**. Vous devriez avoir dans votre répertoire un programme (complètement autonome) nommé **AdoPrimo.exe**.



6. Amélioration

Nous allons maintenant procéder à quelques modifications de notre application pour qu'elle soit plus facile d'utilisation.

L'affichage des données pour la navigation n'est pas très aisée puisqu'il s'agit du classement par le champ **Id**.



Nous allons donc paramétrer le composant *ADOTable1* pour qu'il extraie les données dans l'ordre alphabétique des noms de client.

Dans l'**Explorateur d'objet**, placez-vous sur la propriété **IndexFieldNames**.

Saisir dans le champ, la colonne sur laquelle on doit trier les enregistrements (pour notre exemple : **Nom**).

Nous allons enfin extraire la chaîne de connexion du composant *ADOConnection1* pour l'inclure directement dans le code d'ouverture de la fiche.

Faîtes un **couper** / **coller** du contenu de la propriété **ConnectionString** et placez le code suivant dans la procédure de création de la fiche.

```
procedure TfPrimo.FormCreate(Sender: TObject);
var
  cheminBD, chaineCnx : string;
begin

cheminBD := 'C:\Program Files\Borland\Delphi7\Projects\' +
            'AdoPrimo\Client.mdb';

chaineCnx:= 'Provider=Microsoft.Jet.OLEDB.4.0;' +
            'User ID=Admin;' +
            'Data Source=' + cheminBD + ';' +
            'Mode=Share Deny None;Extended Properties="";'

{
      Toutes les lignes qui suivent sont inutiles
      la connexion. Vous pouvez les enlever.

            Jet OLEDB:System database="";
            Jet OLEDB:Registry Path="";
            Jet OLEDB:Database Password="";
            Jet OLEDB:Engine Type=5;
            Jet OLEDB:Database Locking Mode=1;
            Jet OLEDB:Global Partial Bulk Ops=2;
            Jet OLEDB:Global Bulk Transactions=1;
            Jet OLEDB:New Database Password="";
            Jet OLEDB:Create System Database=False;
            Jet OLEDB:Encrypt Database=False;
            Jet OLEDB:Don't Copy Locale on Compact=False;
            Jet OLEDB:Compact Without Replica Repair=False;
            Jet OLEDB:SFP=False
}

ADOConnection1.ConnectionString := chaineCnx ;

ADOTable1.Active := True;

end;
```

Note :

Pour éviter d'avoir à recompiler le programme à chaque déplacement physique de la base Access, il serait plus pratique de mettre la variable **cheminBD** dans un fichier **INI** ou dans la base de registre.

Bien sûr comme la chaîne de connexion est vide, les composants n'affichent plus les données. Rassurez-vous à l'exécution du programme, tout est normal.

7. Conclusion

Voilà votre application est prête à être déployée. Vous avez à votre disposition un programme exécutable autonome qui tient sur une simple disquette 1,44 Mo car faisant moins de **900 ko**. La même application sous Windev aurait nécessitée l'ajout des **6 Mo** de bibliothèques maison (les fameuses **WD*.DLL**)

Impressionnant également : le temps de développement puisqu'il m'a fallu à peine 30 minutes pour la réaliser. 

Ceci n'est qu'une introduction à la compréhension et la programmation sous Delphi d'une petite application ADO sans avoir besoin de recourir au langage Pascal Objet. Je vous invite à poursuivre avec un autre support intitulée « La connexion ADO Première application ».

Ce second manuel vous permettra d'approfondir votre apprentissage avec une application beaucoup plus complète qui traite de nombreux cas de figure. Pour plus d'information, reportez-vous à l'adresse suivante : www.beaussier.com/?pg=doc