



Tutoriel : Création d'une application de base de données CLX



Borland®
Delphi™ 7
pour Windows™

COPYRIGHT © 2001–2002 Borland Software Corporation. Tous droits réservés. Tous les produits Borland sont des marques commerciales ou des marques déposées de Borland Software Corporation aux États-Unis ou dans les autres pays. Toutes les autres marques sont la propriété de leurs fabricants respectifs.

D7-DB-0802

Table des matières

Création d'une application de base de données CLX

Présentation de l'architecture des bases de données	1
Création d'un nouveau projet	2
Installation des composants d'accès aux données	3
Installation de la connexion à la base de données	3
Installation de l'ensemble de données unidirectionnel	5
Installation du fournisseur, de l'ensemble de données client et de la source de données	5
Conception de l'interface utilisateur	6
Création de la grille et de la barre de navigation	7

Ajout du support pour un menu	9
Ajout d'un menu	10
Ajout d'un bouton	12
Affichage d'un titre et d'une image	13
Ecriture d'un gestionnaire d'événement	14
Ecriture d'un gestionnaire d'événement pour la commande Mise à jour immédiate !	14
Ecriture d'un gestionnaire d'événement pour la commande Quitter	15
Ecriture d'un gestionnaire d'événement pour fermer la fiche	16

Index

Création d'une application de base de données CLX

Ce tutoriel vous guide dans la création d'une application multiplate-forme vous permettant de voir et de mettre à jour la base de données exemple employée. Les applications multiplates-formes utilisent la CLX, la bibliothèque de composants pour le développement multiplates-formes de Borland. Conçues pour être compilées et exécutées sur différentes plates-formes, les applications CLX nécessitent un minimum de modifications entre les ports Windows et Linux.

Ce tutoriel suppose que vous êtes familier de Windows et que vous avez lu l'introduction à la programmation Delphi et à l'environnement de développement dans la *Prise en main*.

Remarque Ce tutoriel est écrit pour les éditions Professionnel et Entreprise de Delphi, qui comprennent les composants pour bases de données. Vous devez aussi avoir installé InterBase pour effectuer ce tutoriel avec succès.

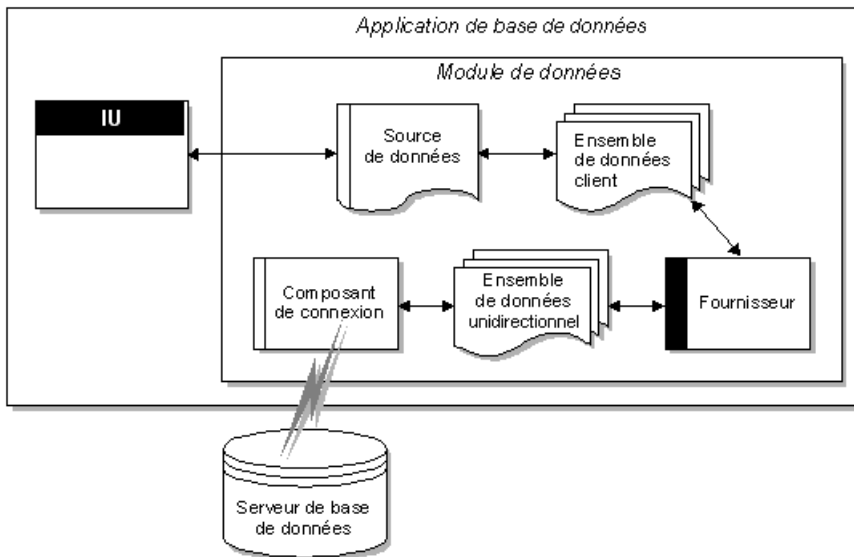
Présentation de l'architecture des bases de données

De prime abord, l'architecture d'une application de base de données peut paraître compliquée, mais, l'utilisation de plusieurs composants simplifie le développement et la maintenance des applications de bases de données réelles.

Les applications de bases de données se composent de trois parties principales : l'interface utilisateur, un ensemble de composants d'accès aux données et la base de données elle-même. Dans ce tutoriel, vous allez créer une application de base de données *dbExpress*. Les applications relatives aux autres bases de données présentent une architecture similaire.

L'interface utilisateur contient des contrôles orientés données, comme une grille, permettant aux utilisateurs de modifier et de valider les données de la base. Les composants d'accès aux données comprennent la source de données, l'ensemble de données client, le fournisseur de données, un ensemble de données

unidirectionnel et un composant connexion. La source de données relie l'interface utilisateur à un ensemble de données client. L'ensemble de données client est le cœur de l'application car il contient un ensemble d'enregistrements issus de la base de données sous-jacente et placés dans un tampon en mémoire. Le fournisseur transfère les données entre l'ensemble de données client et l'ensemble de données unidirectionnel, qui lit directement les données depuis la base de données. Enfin, le composant connexion établit la connexion à la base de données. Chaque type d'ensemble de données unidirectionnel utilise un type de composant connexion différent.



Pour plus d'informations sur le développement des bases de données, voir "Conception d'applications de bases de données" dans le *Guide du développeur* ou dans l'aide en ligne.

Création d'un nouveau projet

Avant de commencer le tutoriel, créez le dossier qui contiendra les fichiers source. Puis, créez et enregistrez un nouveau projet.

- 1 Créez un dossier appelé Tutoriel pour contenir les fichiers du projet que vous allez créer pendant votre travail au cours de ce tutoriel.
- 2 Commencez un nouveau projet CLX. Pour créer un nouveau projet multiplate-forme, choisissez Fichier | Nouveau | Application CLX. (Quand vous démarrez Delphi, un nouveau projet est créé pour une application VCL ; veillez à suivre ces étapes pour créer une application CLX.)
- 3 Choisissez Fichier | Tout enregistrer pour enregistrer sur disque vos fichiers. Lorsque le dialogue d'enregistrement apparaît, naviguez jusqu'à votre dossier Tutoriel et enregistrez chaque fichier en utilisant le nom par défaut.

Par la suite, vous pourrez enregistrer votre travail à tout moment en choisissant Fichier | Tout enregistrer. Si vous décidez de ne pas terminer le tutoriel en une fois, vous pourrez ouvrir la version enregistrée en choisissant Fichier | Réouvrir et en sélectionnant le tutoriel dans la liste.

Installation des composants d'accès aux données

Les composants d'accès aux données représentent à la fois les données (ensembles de données) et les composants qui connectent les ensembles de données aux autres parties de votre application. Chacun de ces composants d'accès aux données pointe sur le composant immédiatement inférieur. Par exemple, la source de données pointe sur l'ensemble de données client, l'ensemble de données client pointe sur le fournisseur, et ainsi de suite. Quand vous installerez vos composants d'accès aux données, vous ajouterez d'abord le composant inférieur.

Dans les sections suivantes, vous allez ajouter les composants de bases de données pour créer la connexion à la base, l'ensemble de données unidirectionnel, le fournisseur, l'ensemble de données client et la source de données. Ensuite, vous créerez l'interface utilisateur de l'application. Ces composants se trouvent sur les pages dbExpress, AccèsBD et ContrôleBD de la palette de composants.

Astuce Il est conseillé d'isoler l'interface utilisateur sur une fiche indépendante et de placer les composants d'accès aux données dans un module de données. Mais, pour simplifier ce tutoriel, vous placerez l'interface utilisateur et tous les composants sur la même fiche.

Installation de la connexion à la base de données

La page dbExpress contient un ensemble de composants qui permettent d'accéder rapidement aux serveurs de bases de données SQL.

Vous devez ajouter un composant connexion qui vous permettra de vous connecter à une base de données. Le type du composant connexion dépendra du type de composant ensemble de données utilisé. Dans ce tutoriel, vous utiliserez les composants *TSQLConnection* et *TSQLDataSet*.

Pour ajouter un composant de connexion dbExpress :



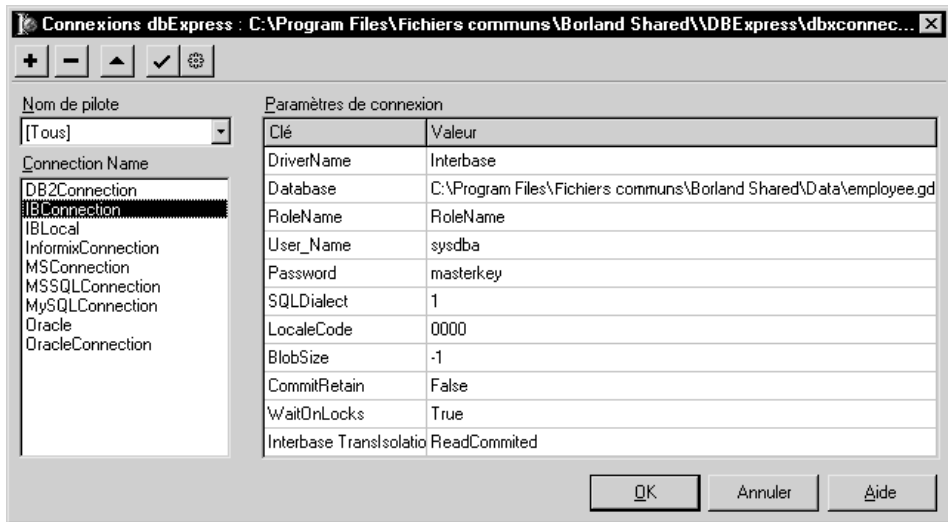
- 1 Cliquez sur la page dbExpress de la palette des composants et double-cliquez sur le composant *TSQLConnection* pour le placer sur la fiche. Pour trouver le composant *TSQLConnection*, pointez sur chaque icône de la palette pendant un moment ; un conseil d'aide vous montrera le nom de chaque composant. Le composant est nommé *SQLConnection1* par défaut.

Le composant *TSQLConnection* n'étant pas visuel, l'endroit où vous le placez n'a pas d'importance. Mais pour ce tutoriel, le plus simple est d'aligner les composants non visuels en haut de la fiche.

Astuce

Pour afficher les libellés des composants non visuels sur une fiche, choisissez Outils | Options d'environnement | Concepteur et cliquez sur Afficher les libellés des composants.

- 2 Dans l'inspecteur d'objets, définissez sa propriété *ConnectionString* par *IBConnection* (ce terme apparaît dans la liste déroulante).
- 3 Initialisez la propriété *LoginPrompt* à *False*. (Définir cette propriété par *False* évite une demande de connexion à chacun de vos accès à la base.)
- 4 Double-cliquez sur le composant *TSQLConnection* pour afficher l'éditeur de connexion.



Vous utiliserez l'éditeur de connexion afin de sélectionner une configuration de connexion pour le composant *TSQLConnection* ou pour modifier les connexions stockées dans le fichier *dbxconnections.ini*. Toutes les modifications effectuées dans le dialogue sont écrites dans ce fichier lorsque vous cliquez sur OK. De plus, lorsque vous cliquez sur OK, la connexion sélectionnée est attribuée comme valeur à la propriété *ConnectionString* du composant connexion SQL.

- 5 Dans l'éditeur de connexion, spécifiez le chemin d'accès sur votre système au fichier base de données *employee.gdb*. Dans ce tutoriel, vous vous connecterez à une base de données exemple *InterBase*, *employee.gdb*, fournie avec Delphi. Par défaut, l'installation d'*InterBase* place *employee.gdb* dans *C:\Program Files\Fichiers communs\Borland Shared\Data*.
- 6 Vérifiez que les champs *Nom d'utilisateur* et *Mot de passe* contiennent des valeurs acceptables. Si vous n'avez pas modifié les valeurs par défaut, vous n'avez pas besoin de modifier ces champs. Si l'accès à la base de données est sous le contrôle d'un autre administrateur, vous risquez d'avoir besoin d'un nom d'utilisateur et d'un mot de passe pour accéder à la base de données.

- 7 Lorsque vous avez fini de vérifier et de modifier les champs, cliquez sur OK pour fermer l'éditeur de connexion et enregistrer vos modifications.

Ces modifications sont écrites dans le fichier `dbxconnections.ini` et la connexion sélectionnée est attribuée comme valeur à la propriété `ConnectionString` du composant `SQLConnection`.

Astuce Si vous avez besoin d'une assistance particulière lorsque vous utilisez l'éditeur de connexion, cliquez sur le bouton Aide.

- 8 Choisissez Fichier | Tout enregistrer pour enregistrer votre projet.

Installation de l'ensemble de données unidirectionnel

Une application de base de données type utilise un ensemble de données pour accéder aux informations de la base. Dans les applications dbExpress, on utilise un ensemble de données unidirectionnel. Un ensemble de données unidirectionnel lit les données de la base mais ne les met pas à jour.

Pour ajouter l'ensemble de données unidirectionnel :



- 1 A partir de la page dbExpress, placez un `TSQLDataSet` en haut de la fiche.
- 2 Dans l'inspecteur d'objets, définissez la propriété `SQLConnection` par `SQLConnection1` (la connexion de base de données créée précédemment).
- 3 Définissez `CommandText` par `Select * from SALES` pour spécifier la commande exécutée par l'ensemble de données. Vous pouvez soit taper l'instruction `Select` dans l'inspecteur d'objets, soit cliquer sur les points de suspension à droite de `CommandText` pour afficher l'éditeur de `CommandText`, dans lequel vous pouvez construire votre propre instruction de requête.

Astuce Si vous avez besoin d'une assistance particulière lorsque vous utilisez l'éditeur de `CommandText`, cliquez sur le bouton Aide.

- 4 Définissez `Active` par `True` pour ouvrir l'ensemble de données.
- 5 Choisissez Fichier | Tout enregistrer pour enregistrer le projet.

Installation du fournisseur, de l'ensemble de données client et de la source de données

La page AccèsBD contient des composants pouvant être utilisés avec n'importe quel mécanisme d'accès aux données, et pas seulement avec dbExpress.

Les composants fournisseur sont pour les ensembles de données client le moyen d'obtenir leurs données en provenance des autres ensembles de données.

Le fournisseur reçoit les demandes de données issues de l'ensemble de données client, lit les données, crée des paquets de données et les renvoie à l'ensemble de données client. Si vous utilisez dbExpress, le fournisseur reçoit les mises à jour d'un ensemble de données client et les applique au serveur de la base de données.

Pour ajouter le fournisseur :



- 1 A partir de la page AccèsBD, déposez un composant *TDataSetProvider* en haut de la fiche.
- 2 Dans l'inspecteur d'objets, définissez la propriété *DataSet* du fournisseur par *SQLDataSet1*.

L'ensemble de données client met ses données en mémoire tampon. Il met également en mémoire cache les mises à jour à envoyer à la base de données. Vous pouvez utiliser les ensembles de données client pour fournir des données aux contrôles orientés données de l'interface utilisateur à l'aide du composant source de données.

Pour ajouter l'ensemble de données client :



- 1 A partir de la page AccèsBD, déposez un composant *TClientDataSet* à droite du composant *TDataSetProvider*.
- 2 Définissez la propriété *ProviderName* par *DataSetProvider1*.
- 3 Définissez la propriété *Active* par *True* pour permettre la transmission des données à votre application.

Une source de données se connecte à l'ensemble de données client par des contrôles orientés données. Chaque contrôle orienté données doit être associé à un composant source de données afin de pouvoir afficher et manipuler des données. De même, tous les ensembles de données doivent être associés à un composant source de données afin que leurs données puissent être affichées et manipulées dans les contrôles orientés données d'une fiche.

Pour ajouter la source de données :



- 1 A partir de la page AccèsBD, déposez un composant *TDataSource* à droite du composant *TClientDataSet*.
- 2 Définissez la propriété *DataSet* de la source de données par *ClientDataSet1*.
- 3 Choisissez *Fichier | Tout enregistrer* pour enregistrer le projet.

Jusqu'ici vous avez ajouté à votre application son infrastructure de base de données non visuelle. Maintenant, vous devez concevoir son interface utilisateur.

Conception de l'interface utilisateur

Vous devez maintenant ajouter les contrôles visuels de l'application afin que les utilisateurs puissent voir, modifier et enregistrer les données. La page *ContrôleBD* fournit des contrôles orientés données qui fonctionnent avec les données d'une base et construisent une interface utilisateur. Vous allez afficher la base de données dans une grille et ajouter quelques commandes et une barre de navigation.

Création de la grille et de la barre de navigation

Pour créer l'interface de l'application :



- 1 Vous pouvez commencer par ajouter une grille à la fiche. A partir de la page ContrôleBD, déposez un composant *TDBGrid* sur la fiche.
- 2 Définissez les propriétés de *DBGrid* afin d'ancrer la grille. Cliquez sur le signe **+** proche de *Anchors* dans l'inspecteur d'objets afin d'afficher *akLeft*, *akTop*, *akRight* et *akBottom* ; définissez-les tous par *True*. La façon la plus rapide est de double-cliquer sur *False* en face de *chaque propriété* dans l'inspecteur d'objets.
- 3 Alignez la grille et le bas de la fiche en définissant la propriété *Align* par *alBottom*. Vous pouvez aussi augmenter la taille de la grille en la faisant glisser ou en définissant sa propriété *Height* par 400.
- 4 Définissez la propriété *DataSource* de la grille par *DataSource1*. Lorsque vous faites cela, la grille se remplit de données issues de la base de données *employee*. Si la grille n'affiche pas de données, vérifiez que vous avez correctement défini les propriétés de tous les objets présents sur la fiche, comme expliqué par les instructions précédentes.

A ce stade, votre application doit ressembler à ceci :

PO_NUMBER	CUST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE
V91E0210	1004	11	shipped	3/4/1991
V92E0340	1004	11	shipped	10/15/1992
V92F3004	1012	11	shipped	10/15/1992
V92J1003	1010	61	shipped	7/26/1992
V9320630	1001	127	open	12/12/1993
V9324200	1001	72	shipped	8/9/1993
V9324320	1001	127	shipped	8/16/1993
V9333005	1002	11	shipped	2/3/1993
V9333006	1002	11	shipped	4/27/1993
V9336100	1002	11	waiting	12/27/1993
V9345139	1003	127	shipped	9/9/1993
V9345200	1003	11	shipped	11/11/1993
V9346200	1003	11	waiting	12/31/1993
V93B1002	1014	134	shipped	9/20/1993

Le contrôle *DBGrid* affiche les données en mode conception, pendant que vous travaillez dans l'EDI. Cela vous permet de vérifier que vous êtes bien connecté à la base de données. Vous ne pouvez pas, cependant, modifier les données pendant les phases de conception ; pour modifier les données de la table, vous devez exécuter l'application.



- 5 A partir de la page ContrôleBD, placez un contrôle *TDBNavigator* sur la fiche. Dans une base de données, le navigateur est l'outil qui permet de se déplacer dans les données d'un ensemble de données (en utilisant les flèches suivante et précédente, par exemple) et d'effectuer des opérations sur les données.
- 6 Définissez la propriété *DataSource* de la barre de navigation par `DataSource1` de façon à ce que le navigateur recherche les données dans l'ensemble de données client.
- 7 Définissez la propriété *ShowHint* de la barre de navigation par `True`. (Si *ShowHint* est à `True`, les conseils apparaissent lorsque le curseur est positionné, en mode exécution, sur l'un des éléments de la barre de navigation.)
- 8 Choisissez `Fichier | Tout enregistrer` pour enregistrer le projet.
- 9 Appuyez sur `F9` pour compiler et exécuter le projet. Vous pouvez également exécuter le projet en cliquant sur le bouton `Exécuter` dans la barre d'outils du débogueur, ou en choisissant `Exécuter` dans le menu `Exécuter`.



PO_NUMBER	CUST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE
V91E0210	1004	11	shipped	3/4/1991
V92E0340	1004	11	shipped	10/15/1992
V92F3004	1012	11	shipped	10/15/1992
V92J1003	1010	61	shipped	7/26/1992
V9320630	1001	127	open	12/12/1993
V9324200	1001	72	shipped	8/9/1993
V9324320	1001	127	shipped	8/16/1993
V9333005	1002	11	shipped	2/3/1993
V9333006	1002	11	shipped	4/27/1993
V9336100	1002	11	waiting	12/27/1993
V9345139	1003	127	shipped	9/9/1993
V9345200	1003	11	shipped	11/11/1993
V9346200	1003	11	waiting	12/31/1993
V93B1002	1014	134	shipped	9/20/1993

Lorsque vous exécutez votre projet, le programme s'ouvre dans une fenêtre semblable à celle que vous avez dessinée sur la fiche. Vous pouvez tester la barre de navigation sur la base de données `employee`. Par exemple, vous pouvez aller d'enregistrement enregistrement en utilisant les commandes fléchées, ajouter des enregistrements en utilisant la commande `+`, supprimer des enregistrements en utilisant la commande `-`.



Astuce

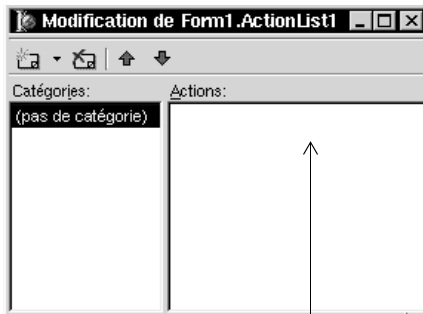
Si une erreur se produit lorsque vous testez une version initiale de votre application, choisissez `Exécuter | Réinitialiser le programme` pour revenir à la vue conception.

Ajout du support pour un menu

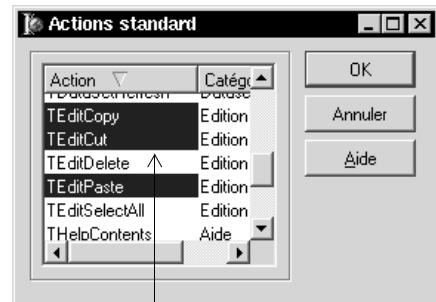
Bien que votre programme possède un certain nombre de fonctionnalités, il lui manque encore la majorité des fonctions habituellement disponibles dans les applications à interface utilisateur graphique. Par exemple, la plupart des applications implémentent des menus et des boutons pour faciliter leur utilisation.

Dans cette section, vous allez ajouter une *liste d'actions*. Bien qu'il soit tout à fait possible de créer menus, barres d'outils et boutons sans liste d'actions, celles-ci simplifient le développement et la maintenance en centralisant les réponses aux commandes de l'utilisateur. (Notez que dans le cadre du développement pour Windows uniquement, vous pouvez aussi utiliser les bandes d'actions pour faciliter la création des barres d'outils et des menus.)

- 1 Si l'application est toujours en mode exécution, cliquez sur le **X** dans le coin supérieur droit, pour la fermer et revenir à la vue conception de la fiche.
- 2  A partir de la page Contrôles communs de la palette de composants, déposez un composant *ImageList* sur la fiche. Aligned-le à côté des composants non visuels. *ImageList* contiendra des icônes représentant des actions standard comme couper et coller.
- 3  A partir de la page Standard de la palette de composants, déposez un composant *ActionList* sur la fiche. Définissez la propriété *Images* de la liste d'actions par *ImageList1*.
- 4 Double-cliquez sur la liste d'actions pour afficher l'éditeur de liste d'actions.



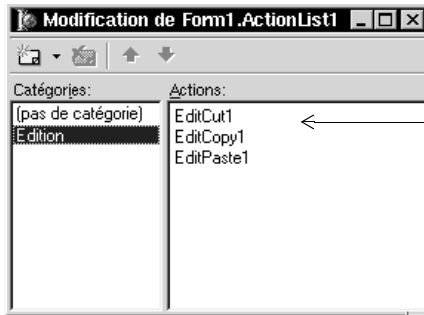
Cliquez avec le bouton droit de la souris dans l'éditeur et choisissez Nouvelle action standard pour afficher la boîte liste Actions standard.



Sélectionnez les actions que vous voulez et cliquez sur OK. Appuyez sur *Ctrl* pour sélectionner plusieurs actions.

- 5 Cliquez avec le bouton droit de la souris dans l'éditeur de liste d'actions et choisissez Nouvelle action standard. La boîte liste Actions standard apparaît.

- 6 Sélectionnez les actions suivantes : *TEditCopy*, *TEditCut* et *TEditPaste*. (Utilisez la touche *Ctrl* pour sélectionner plusieurs éléments.) Cliquez ensuite sur OK.



Vous avez ajouté des actions standard livrées avec le produit.

Vous les utiliserez sur un menu.

- 7 Cliquez avec le bouton droit de la souris dans l'éditeur de liste d'actions et choisissez Nouvelle action pour ajouter une action qui n'est pas fournie par défaut. Action1 est ajoutée par défaut. Dans l'inspecteur d'objets, définissez sa propriété *Caption* par Mise à jour immédiate !

La même action sera utilisée par un menu et un bouton. Par la suite, nous ajouterons un gestionnaire d'événement qui mettra à jour la base de données.

- 8 Cliquez sur (pas de catégorie), cliquez avec le bouton droit de la souris et choisissez Nouvelle action pour ajouter une autre action. Action2 est ajoutée. Définissez sa propriété *Caption* par &Quitter.

- 9 Cliquez sur le **X** (dans le coin supérieur droit) pour fermer l'éditeur de liste d'actions.

Vous avez ajouté trois actions standard plus deux autres actions que nous connecterons plus tard à des gestionnaires d'événements.

- 10 Choisissez Fichier | Tout enregistrer pour enregistrer le projet.

Ajout d'un menu

Dans cette section, vous allez ajouter une barre de menu principal contenant deux menus déroulants — Fichier et Edition — et vous ajouterez des éléments à chacun d'eux en utilisant les actions de la liste d'actions.

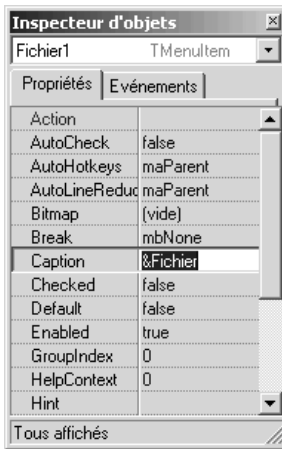


- 1 A partir de la page Standard de la palette de composants, déposez un composant *TMainMenu* sur la fiche. Faites-le glisser à côté des autres composants non visuels.
- 2 Définissez la propriété *Images* du menu principal par *ImageList1* afin d'associer la liste d'images aux éléments des menus.

- 3 Double-cliquez sur le composant menu pour afficher le concepteur de menus.



- 4 Tapez &Fichier pour définir la propriété *Caption* du premier élément de niveau supérieur et appuyez sur *Entrée*.

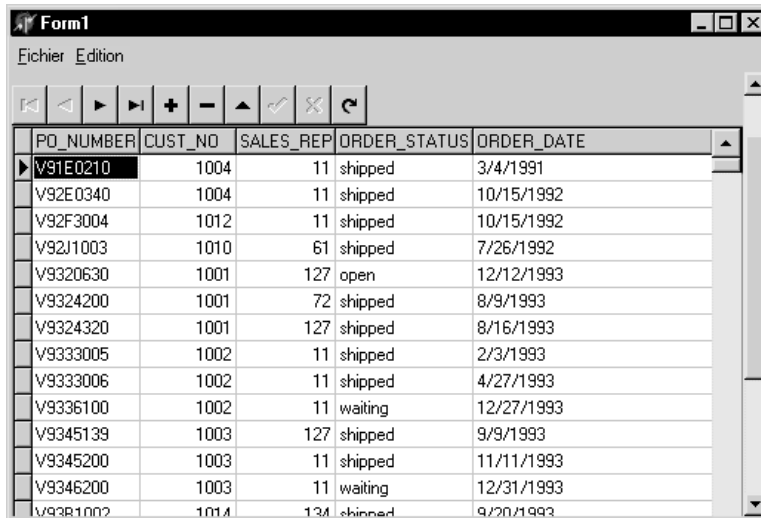


Quand vous tapez &Fichier et appuyez sur *Entrée*, la commande Fichier de niveau supérieur apparaît.
Le et commercial avant un caractère active une touche accélératrice.



- 5 Cliquez sur le menu Fichier et sélectionnez l'élément de menu vierge qui apparaît en-dessous. Définissez la propriété *Action* de l'élément de menu vierge par *Action2*. L'élément de menu Quitter apparaît sous le menu Fichier.
- 6 Cliquez sur le deuxième élément de menu de niveau supérieur (à droite de Fichier). Définissez sa propriété *Caption* par &Edition et appuyez sur *Entrée*. Cliquez sur le menu Edition et sélectionnez l'élément de menu vierge qui apparaît en-dessous.
- 7 Dans l'inspecteur d'objets, définissez sa propriété *Action* par *EditCut1* et appuyez sur *Entrée*. Le libellé de l'élément est défini automatiquement par *Couper* et le bitmap *couper* par défaut apparaît sur le menu.
- 8 Sélectionnez l'élément de menu vierge (sous *Couper*) et définissez sa propriété *Action* par *EditCopy1* (le bitmap *copier* par défaut apparaît sur le menu).
- 9 Sélectionnez l'élément de menu vierge (sous *Copier*) et définissez sa propriété *Action* par *EditPaste1* (le bitmap *coller* par défaut apparaît sur le menu).
- 10 Sélectionnez l'élément de menu vierge (sous *Coller*) et définissez sa propriété *Caption* par un tiret (-) afin de créer une ligne de séparation dans le menu. Appuyez sur *Entrée*.

- 11 Sélectionnez l'élément de menu vierge (sous la ligne de séparation) et définissez sa propriété *Action* par `Action1`. L'élément de menu affiche Mise à jour immédiate !
- 12 Cliquez sur le **X** pour fermer le concepteur de menus.
- 13 Choisissez *Fichier* | *Tout enregistrer* pour enregistrer le projet.
- 14 Appuyez sur *F9* ou sur *Exécuter* dans la barre d'outils pour exécuter votre programme et voir à quoi il ressemble.



PO_NUMBER	CUJST_NO	SALES_REP	ORDER_STATUS	ORDER_DATE
V91E0210	1004	11	shipped	3/4/1991
V92E0340	1004	11	shipped	10/15/1992
V92F3004	1012	11	shipped	10/15/1992
V92J1003	1010	61	shipped	7/26/1992
V9320630	1001	127	open	12/12/1993
V9324200	1001	72	shipped	8/9/1993
V9324320	1001	127	shipped	8/16/1993
V9333005	1002	11	shipped	2/3/1993
V9333006	1002	11	shipped	4/27/1993
V9336100	1002	11	waiting	12/27/1993
V9345139	1003	127	shipped	9/9/1993
V9345200	1003	11	shipped	11/11/1993
V9346200	1003	11	waiting	12/31/1993
V9381002	1011	131	shipped	9/20/1993

- 15 Fermez l'application lorsque vous êtes prêt à poursuivre.

Ajout d'un bouton

Cette section décrit comment ajouter à l'application un bouton Mise à jour immédiate. Ce bouton servira à appliquer toutes les modifications apportées par un utilisateur à la base de données, comme la modification, l'ajout ou la suppression d'enregistrements.

Pour ajouter un bouton :



- 1 A partir de la page Standard de la palette de composants, déposez un composant *TButton* sur la fiche. (Sélectionnez le composant et cliquez sur la fiche à proximité de la barre de navigation.)
- 2 Définissez la propriété *Action* du bouton par `Action1`.

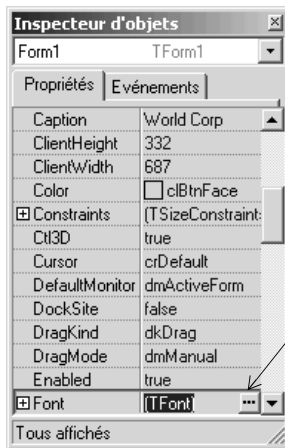
L'intitulé du bouton est modifié en Mise à jour immédiate ! Lorsque vous exécuterez l'application, il restera estompé tant qu'un gestionnaire d'événement ne sera pas ajouté pour le faire fonctionner.

Affichage d'un titre et d'une image

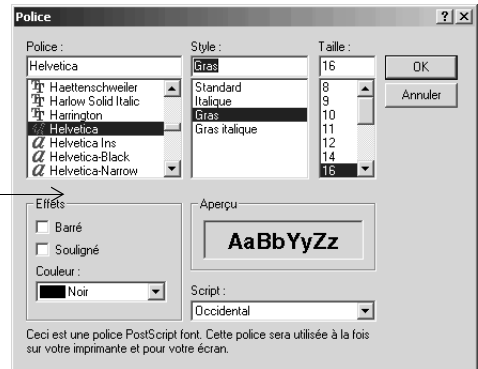
Vous pouvez ajouter le nom de votre société et une image pour donner à votre application un aspect plus professionnel :



- 1 A partir de la page Standard de la palette de composants, déposez un composant *TLabel* sur la fiche (nommé *Label1* par défaut).
- 2 Dans l'inspecteur d'objets, modifiez la propriété *Caption* du libellé par World Corp ou un autre nom de société.
- 3 Changez la police du nom de la société en cliquant sur la propriété *Font*. Cliquez sur les points de suspension qui apparaissent à droite et, dans la boîte de dialogue des fontes, définissez la police par Helvetica Gras, 16 points. (Si l'Helvetica n'est pas disponible, choisissez une autre police.) Cliquez sur OK.



Vous pouvez changer la police du libellé en utilisant la propriété *Font* dans l'inspecteur d'objets.

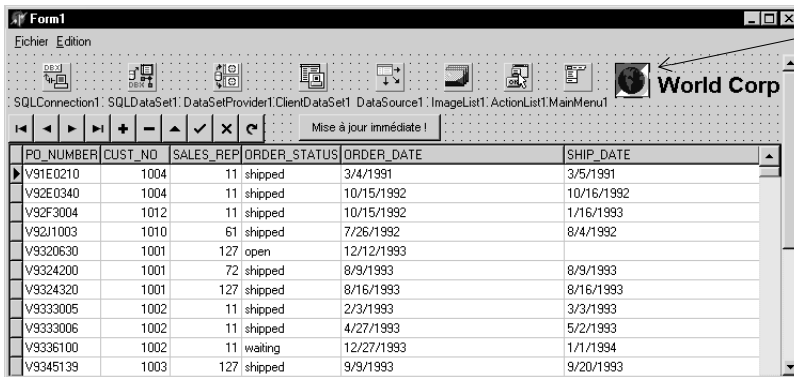


Cliquez sur les points de suspension pour afficher le dialogue standard des fontes.

- 4 Positionnez le libellé dans le coin supérieur droit.
- 5 A partir de la page Supplément de la palette de composants, déposez un composant *TImage* à proximité du libellé (nommé *Image1* par défaut).
- 6 Pour ajouter une image au composant *Image1*, cliquez sur la propriété *Picture*. Cliquez sur les points de suspension pour afficher l'éditeur d'images.
- 7 Dans l'éditeur d'image, choisissez Charger et naviguez jusqu'à Program Files\Fichiers communs\Borland Shared\images\icons.
- 8 Double-cliquez sur earth.ico. Cliquez sur OK pour charger l'image et fermer l'éditeur d'images.



- 9 Donnez à la zone *image* par défaut la taille de l'image. Placez l'image près du libellé.



- 10 Choisissez Fichier | Tout enregistrer pour enregistrer le projet.
 11 Appuyez sur F9 pour compiler et exécuter votre application.
 Fermez l'application lorsque vous êtes prêt à poursuivre.

Ecriture d'un gestionnaire d'événement

La majorité des composants de la palette possèdent des événements, et beaucoup ont un événement par défaut. *OnClick* est un événement par défaut commun, qui est appelé chaque fois que l'on clique sur un composant, par exemple sur un *TButton*. Si vous sélectionnez un composant dans une fiche et cliquez sur l'onglet Événements de l'inspecteur d'objets, vous verrez la liste des événements de ce composant.

Pour plus d'informations sur les événements et sur les gestionnaires d'événements, voir "Développement de l'interface utilisateur d'une application" dans le *Guide du développeur* ou dans l'aide en ligne.

Ecriture d'un gestionnaire d'événement pour la commande Mise à jour immédiate !

D'abord, vous allez écrire le gestionnaire d'événement pour la commande et le bouton Mise à jour immédiate ! :

- 1 Double-cliquez sur le composant *ActionList* pour afficher l'éditeur de liste d'actions.
- 2 Sélectionnez (Toutes les actions) pour afficher la liste des actions.

- 3 Double-cliquez sur Action1. Dans l'éditeur de code, le squelette du gestionnaire d'événement apparaît :

```
procédure TForm1.Action1Execute(Sender: TObject);
begin

end;
```

Exactement à la position du curseur (entre **begin** et **end**), tapez :

```
if ClientDataSet1.State in [dsEdit, dsInsert] then ClientDataSet1.Post;
ClientDataSet1.ApplyUpdates(-1);
```

Ce gestionnaire d'événement teste d'abord l'état dans lequel se trouve la base de données. Lorsque vous quittez un enregistrement modifié, celui-ci est automatiquement validé. Mais, si vous ne quittez pas l'enregistrement modifié, la base reste en mode édition ou en mode insertion. L'instruction **if** valide toute donnée éventuellement modifiée mais non transmise à l'ensemble de données client. L'instruction suivante applique à la base les mises à jour contenues dans l'ensemble de données client.

Remarque Les modifications apportées aux données ne sont pas automatiquement validées dans la base lorsque vous utilisez dbExpress. Il vous faut appeler la méthode *ApplyUpdates* pour écrire tous les enregistrements modifiés, insérés ou supprimés, depuis l'ensemble de données client vers la base de données.

Ecriture d'un gestionnaire d'événement pour la commande Quitter

Ensuite, nous allons écrire le gestionnaire d'événement pour la commande Quitter :

- 1 Double-cliquez sur le composant *ActionList* pour afficher l'éditeur de liste d'actions si ce n'est déjà fait.
- 2 Sélectionnez (Toutes les actions) pour afficher la liste des actions.
- 3 Double-cliquez sur Action2. L'éditeur de code affiche le squelette du gestionnaire d'événement :

```
procédure TForm1.Action2Execute(Sender: TObject);
begin

end;
```

Exactement à la position du curseur (entre **begin** et **end**), tapez :

```
Close;
```

Ce gestionnaire d'événement fermera l'application lorsque la commande Fichier | Quitter sera utilisée.

- 4 Fermez l'éditeur de liste d'actions.
- 5 Choisissez Fichier | Tout enregistrer pour enregistrer le projet.

Ecriture d'un gestionnaire d'événement pour fermer la fiche

Enfin, vous allez écrire un autre gestionnaire d'événement qui sera invoqué à la fermeture de l'application. L'application peut être fermée soit en utilisant Fichier|Quitter, soit en cliquant sur le **X** du coin supérieur droit. De toute façon, le programme vérifie s'il y a des mises à jour en attente par rapport à la base de données et affiche éventuellement une fenêtre de message demandant à l'utilisateur ce qu'il souhaite en faire.

Vous pourriez placer ce code dans le gestionnaire de l'événement Quitter, mais les éventuelles modifications de la base seraient perdues dans le cas où l'utilisateur aurait choisi de fermer l'application avec le **X**.

- 1 Cliquez sur la zone grille de la fiche principale pour sélectionner la fiche.
- 2 Sélectionnez l'onglet Evénements de l'inspecteur d'objets pour voir les événements de la fiche.
- 3 A côté de *OnClose* tapez FormClose puis double-cliquez sur FormClose. Le squelette d'un gestionnaire de l'événement FormClose est écrit et affiché dans l'éditeur de code à la suite des autres gestionnaires d'événements.

```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin

end;
```

Exactement à la position du curseur (entre **begin** et **end**), tapez :

```
Action := caFree;
if ClientDataSet1.State in [dsEdit, dsInsert] then
  ClientDataSet1.Post;
if ClientDataSet1.ChangeCount > 0 then
begin
  Option := Application.MessageBox(
    'Il y a des mises à jour en attente. Voulez-vous les écrire dans la base?',
    'Mises à jour en attente', [smbYes, smbNo, smbCancel], smsWarning, smbYes);
  case Option of
    smbYes: ClientDataSet1.ApplyUpdates(-1);
    smbCancel: Action := caNone;
  end;
end;
```

Ce gestionnaire d'événement teste l'état de la base de données. Si des mises à jour sont en attente, elles sont validées dans l'ensemble de données client où le compteur de modifications est incrémenté. Puis, avant la fermeture de l'application, une boîte de message est affichée pour demander comment gérer ces modifications. Les réponses possibles sont Oui, Non ou Annuler. Répondre Oui applique les mises à jour à la base ; Non ferme l'application sans modifier la base ; Annuler annule la sortie sans annuler les modifications de la base de données et conserve l'application en exécution.

- 4 Vous devez déclarer la variable utilisée dans la procédure. Sur une ligne entre **procedure** et **begin**, tapez :

```
var
  Option: TMessageButton;
```

- 5 Vérifiez que l'ensemble de la procédure ressemble à ce qui suit :

```
procédure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
var
  Option: TMessageButton;
begin
  Action := caFree;
  if ClientDataSet1.State in [dsEdit, dsInsert] then
    ClientDataSet1.Post;
  if ClientDataSet1.ChangeCount > 0 then
    begin
      Option := Application.MessageBox(
        'Il y a des mises à jour en attente. Voulez-vous les écrire dans la base?',
        'Mises à jour en attente',[smbYes, smbNo, smbCancel], smsWarning, smbYes);
      case Option of
        smbYes: ClientDataSet1.ApplyUpdates(-1);
        smbCancel: Action := caNone;
      end;
    end;
  end;
```

- 6 Pour finir, choisissez Fichier | Tout enregistrer pour enregistrer le projet. Puis, appuyez sur *F9* pour exécuter l'application.

Astuce Corrigez les erreurs en double-cliquant sur le message correspondant pour aller à la ligne de code concernée ou en appuyant sur *F1* pour obtenir de l'aide sur le message.

Et voilà ! Vous pouvez essayer l'application et observer son fonctionnement. Pour quitter le programme, vous pouvez utiliser la commande Fichier | Quitter désormais totalement fonctionnelle.

Index

A

ActionList, composant 9
affichage des images 13
ajout

- connexion de base de données à une application 3
- grille à une application 7
- image à une application 13
- menu principal à une application 10
- titre à une application 13

application des modifications à une base de données 12
applications de bases de données

- accès 3–5
- présentation 1

architecture, base de données 1

B

base de données exemple 1–17
base de données, nom d'utilisateur et mot de passe 4
bouton, ajout de Mise à jour immédiate 12
Button, composant 12

C

ClientDataSet, composant 6
compilation des programmes 8
composants

- ActionList 9
- Bouton 12
- ClientDataSet 6
- DataSetProvider 6
- DataSource 6
- DBGrid 7
- DBNavigator 8
- Image 13
- ImageList 9
- Libellé 13
- MainMenu 10
- non visuels 3
- SQLConnection 3
- SQLDataSet 5

composants non visuels 3
création d'un projet 2

D

DataSetProvider, composant 6
DataSource, composant 6
DBGrid, composant 7
DBNavigator, composant 8

E

écriture de gestionnaires d'événements 14–17
employee.gdb, base de données exemple 4
ensembles de données

- client 6
- unidirectionnels 5

Exécuter, bouton 8
exécution des applications 8
exemple, base de données 4

G

gestionnaires d'événements, écriture 14–17
graphiques, affichage 13
grille, ajout à une application 7

I

icônes, ajout à une application 13
Image, composant 13
ImageList, composant 9

L

Label, composant 13

M

MainMenu, composant 10
menu, ajout à une application 10
mise à jour d'une base de données 12
Mise à jour immédiate, bouton 12

N

navigation parmi les données d'un ensemble de données 8

P

Panel, composant 13
programme exemple 1–17
projet

- création 2
- exécution 8

S

SQLConnection, composant 3
SQLDataSet, composant 5

T

tutoriel 1–17

U

unidirectionnel, ensemble de données 5