

Création d'une base de donnée avec DELPHI (pascal de BORLAND)

Création du répertoire de TRAVAIL ;

Commencer par créer un répertoire dans le répertoire DELPHI. Lancer l'Explorateur, Activer le répertoire DELPHI puis dans le menu **Fichiers** de l'explorateur demander **Nouveau Dossier**. L'explorateur crée immédiatement un sous dossier de DELPHI que l'on nomme **FIC2ND** (il suffit en fait de remplacer **Nouveau dossier** par **FIC2ND**). Ce répertoire contiendra le fichier exécutable que nous allons créer ainsi que la base de données.

1- Création d'une Fiche au Format DBASE.

DBase est un format officiel de base de données. Il est donc facile d'exporter ou d'importer un fichier DBase. Nous allons développer le logiciel que nous avons utilisé en seconde : un fichier de classe : une fiche contiendra les champs suivants :

- Nom (20 lettres) - Prénom (20) - Classe (2)
- Commune (30) - Sexe (1)

Lancer **Delphi32.exe**.

Dérouler le menu **Outils** et demander **Module Base de données**.

Faire **Fichiers-Nouveau-Table** et demander comme type de Table : **Dbase pour Windows**.

Remplir alors les champs, leur Type (Caractères) et les tailles. La colonne **nombre de décimales** n'est utilisée que pour un format numérique et ne nous concerne donc pas.

	Nom de champ	Type	Taille	Dec
1	NOM	C	20	
2	PRENOM	C	20	
3	CLASSE	C	2	
4	COMMUNE	C	30	
5	SEXE	C	1	

Cliquer sur **définir les index** (les index permettent de trier une base de donnée). Nous allons définir un index pour chacun des champs de façon à pouvoir trier sur n'importe lequel des critères.

Cliquer sur le champ **NOM** puis sur **expression d'index** puis sur **NOM**, appuyer sur **ENTREE** et taper **NOM1**. La module contiendra un champ caché **NOM1** permettant d'enregistrer la base sous un ordre défini (croissant par défaut).

Recommencer avec le champ **PRENOM** (et taper **PRENOM1**), puis ainsi de suite avec tous les champs.

Cliquer sur **Enregistrer sous** et aller dans le répertoire **\DELPHI\FIC2ND**, taper le nom de la base (ex : **JEAN01**). Votre base de données au format DBASE est créée et enregistrée. Elle peut être utilisée pour programmer une interface utilisateur à l'aide de Delphi.

2- Interface utilisateur

Refermer le module **Base de données**.

Description rapide de l'interface Delphi. On se trouve en permanence en présence de trois modules :

- Le module principal **Delphi 3** (en haut) contenant le **menu déroulant** et toutes les **barrettes d'outils**. C'est lui qui permet de décider des composants à placer sur la fiche et de **compiler** ou **exécuter** l'application.
- La **fiche de travail** (fiche vierge au départ nommée **Form1**) qui est double :
 - Partie visuelle telle qu'elle apparaît au yeux de l'utilisateur
 - Partie texte dans lequel Delphi écrit lui-même le code pascal et nous laisse la possibilité de travailler sur ce code.
- **L'inspecteur d'objet**, propre a chaque objet créé sur la fiche (bouton par exemple) et contient deux onglets :
 - Les **propriétés** de l'objet, appelées aussi méthodes, nous permettent de modifier sa taille, se couleur, ses actions etc.
 - Les **événements** qui permettent d'écrire les procédures qui doivent s'exécuter lors de tel ou tel événement (clic, touche, fermeture de fenêtre etc.)

Nous allons apprendre à travailler avec ces trois modules et voir avec quelle facilité il devient possible de développer une véritable application windows. La programmation avec Delphi est **orientée objets** c'est à dire que le programme n'a pas de début (ni de fin) mais est constitué d'un ensemble d'objets qui réagissent aux événements (clic de souris, appui sur une touche, déplacement de souris etc.)

a- Nom de la fiche de travail

Pour créer une interface utilisateur nous partons d'une fiche vierge nommée **Form1**. Nous allons commencer par personnaliser le nom de cette fiche :

- Dans les **propriétés de l'inspecteur d'objets** remplacer dans le champ **caption** **Form1** par **Base seconde**. La fiche vierge se nomme désormais Base seconde.

Utiliser le menu **Fichiers-Enregistrer le projet sous** et enregistrer le projet dans le répertoire **\DELPHI\FIC2ND** sous le nom de **FIC2ND1** par exemple.

Le fait d'enregistrer le projet immédiatement permet de définir le répertoire **\DELPHI\FIC2ND** comme répertoire par défaut.

b- Cadre de travail

Dans la **barrette Standard** cliquer sur le dernier composant **Panel**. Cliquer alors sur la fiche vierge pour placer un fond de travail sur la fiche.

Dans les **propriétés de l'inspecteur d'objets** régler le champ **Align** sur **AlClient**. Le cadre de travail occupe alors toute la fiche (la zone client de la fiche).

Dans les **propriétés de l'inspecteur d'objets** effacer le champ **Caption** (qui doit contenir **Panel1**) sinon le mot **Panel1** reste affiché au centre de la fiche.

Il nous reste maintenant à placer sur notre fiche vierge des objets liés à la base de données que nous venons de construire.

c- Accès à la base de données

Il faut maintenant ouvrir notre base de donnée pour pouvoir en utiliser les champs sur notre fiche.

- Dans la **barrette AccèsBD** cliquer sur le composant **Table** et déposer une table sur la fiche de travail. Nous pouvons la poser n'importe où, elle ne sera pas visible. Elle sert uniquement à ouvrir les champs de la base que nous venons de construire.

Dans les **propriétés de l'inspecteur d'objets** de la table (nommée **Table1**) régler le champ **TableName** sur le Nom de votre base de Données : **JEAN01.DBF** (par exemple).

Ouvrir la base de données en réglant le champ **Active** à **TRUE**.

- Dans la **barrette AccèsBD** cliquer sur le composant **DataSource** et déposer le sur la fiche (lui non plus n'apparaît pas à l'exécution).

Dans les **propriétés de l'inspecteur d'objets** du composant **DataSource** (nommée **DataSource1**) régler le champ **DataSet** sur **Table1**.

Notre fiche est liée à la base de donnée que nous venons de construire et nous allons pouvoir en manipuler les champs à notre guise.

d- Contrôles de la base de données

- Dans la **barrette ControlBD** cliquer sur le composant **DBGrid** et déposer l'objet tableau sur la fiche de travail. Attention ce tableau sera visible ! Vous pouvez le placer à gauche comme sur la copie d'écran de la dernière page. Ce tableau est vide pour l'instant et il faut le lier à la table que nous venons de définir.

Dans les **propriétés de l'inspecteur d'objets** du composant **DBGrid** (nommé **DBgrid1**) régler le champ **DataSource** sur **DataSource1** (qui est lié à l'objet **Table1**, lui-même lié à notre base **DB**).

Le tableau ne contient que les noms de champs. Nous pourrions le remplir et le compléter à loisir lors de l'exécution du programme.

Dans les **propriétés de l'inspecteur d'objets** du composant **DBGrid** (nommé **DBgrid1**) régler le champ **Columns** en cliquant sur les Ceci ouvre une fenêtre de dialogue permettant de n'afficher que les colonnes désirées. Cliquer alors sur **Tous les Champs** et supprimer les champs **CLASSE**, **COMMUNE** et **SEXE**. Refermer la fenêtre de dialogue. Seuls les champs **NOM** et **PRENOM** sont affichés. En fait le tableau va nous servir à naviguer dans la base. Régler les dimensions du tableau comme sur la copie d'écran jointe.

Dans les **propriétés de l'inspecteur d'objets** du composant **DBGrid** dérouler le menu **+Options** en double-cliquant. Régler alors le **dgEditing** à **FALSE**. De cette façon le tableau ne pourra plus accepter de saisie ni de modifications.

Nous allons définir une grille d'édition pour nous permettre d'ajouter ou de modifier les fiches.

e- Grille d'édition

Nous allons définir cinq fenêtre permettant de saisir les cinq champs de nos fiches.

- Dans la **barrette ControlBD** cliquer sur le composant **DBEdit** et déposer le cadre d'édition sur la fiche de travail (voir la copie d'écran). Ce premier cadre servira à saisir le **NOM**.

Dans les **propriétés de l'inspecteur d'objets** du composant **DBEdit** (nommé **DBEdit1**) régler le champ **DataSource** sur **DataSource1** et le champ **DataField** sur **NOM**.

- Dans la **barrette Standard** utiliser le bouton **Label** et placer un objet **Label** devant la fenêtre d'édition du NOM. Taper **NOM** dans la propriété **caption** de ce Label.

Alignement : sélectionner les deux objets (**NOM** et **EDIT1**) en cliquant tout en maintenant la touche **SHIFT enfoncée**. Dérouler alors le **menu contextuel** (bouton droit de la souris) et demander **Aligner** puis choisir un alignement vertical des centres.

Recommencer les mêmes opérations pour les autres champs de la fiche.

La dernière propriété **Width** de chacune des fenêtres **EDIT** peut être réglée en fonction de la longueur des chaînes de caractères demandée.

Penser à **enregistrer** votre projet votre projet.

f- Ajouter et modifier des fiches

Le tableau **Dbgrid1** permet de naviguer facilement dans la base données. Il nous manque seulement deux **boutons** : un pour **ajouter** une fiche et un pour **détruire** une fiche. Nous allons créer ces boutons et cette fois il va falloir écrire un peu de code Pascal !!

Dans la **barrette Supplément** utiliser un bouton **BitBtn** et le placer sous les cadres de saisies. Renommer la commande **Caption** en **&Ajouter** et le Champ **Name** en **Ajouter** (à la place de **BitBtn1**). Cliquer sur les ... de la commande **Glyph** pour charger l'image **\DELPHI\IMAGES\BUTTONS\INSERT.BMP**.

Il faut maintenant écrire la procédure qui va permettre d'entrer dans la fenêtre d'édition et d'ajouter une fiche. Double-cliquer sur le bouton **&Ajouter**. Delphi fabrique aussitôt le squelette de la procédure Pascal **AjouterClick1** qui s'exécutera lorsque l'on cliquera sur ce bouton. Il nous faut alors écrire le code pascal permettant l'ajout d'une fiche entre le **Begin** et le **End** ; de la procédure. Nous allons en fait écrire seulement deux instructions :

- **Table1.Insert** ; cette instruction insère un enregistrement dans notre base **Table1**.
- **DBEdit1.SetFocus** ; pour obliger le curseur à se placer dans la fenêtre d'édition du NOM : Edit1.

La procédure pascal sera donc la suivante (vous n'avez à écrire que les deux lignes en italique) :

```
procedure TForm1.AjouterClick(Sender: TObject);  
begin  
    Table1.Insert ;  
    DBEdit1.SetFocus ;  
end;  
  
end. {dernier end et fin (.) du programme }
```

De la même façon fabriquons un bouton **Supprimer** auquel nous allons coller l'image **DELETE.BMP**.

Dans la procédure pascal **SupprimerClick(Sender : TObject)** il nous suffira de rajouter la seule ligne **Table1.Delete** entre le **Begin** et le **End** ;

Régler alors la taille et l'emplacement des boutons.

g- Premier essai.

Nous allons maintenant enregistrer nos premières fiches. **Exécuter** le programme (menu exécuter). Ajouter quelques fiches pour commencer à remplir la base et pouvoir faire des tris. Une fois les premières fiches saisies vous pouvez refermer la fenêtre pour revenir au développement sous Delphi.

h- Un fenêtre de TRI

Nous allons définir deux boutons permettant de **trier** suivant le **NOM** ou suivant la **CLASSE**. Ces deux boutons seront placés sur une **fenêtre de groupe** intitulée **TRI**.

A l'aide de la **barrette Standard** définir une fenêtre **GroupBox** et la nommer **TRI SUIVANT ...**

Sur cette fenêtre placer deux boutons nommés **NOM** et **CLASSE**. Associer à ces boutons l'image \DELPHI\IMAGES\BUTTONS\Sort.BMP.

La commande Pascal associée au Bouton **NOM** sera :

Table1.IndexFieldNames := 'NOM';

La commande Pascal associée au Bouton **CLASSE** sera :

Table1.IndexFieldNames := 'CLASSE';

Relancer le programme (**F9**) et essayer.

Enregistrer votre travail

i- Des sélections

Pour qu'une base de données soit utilisable il faut pouvoir sélectionner une partie de la base : un classe ou toutes les filles etc.

A l'aide de la **barrette Standard** définir une fenêtre **GroupBox** et la nommer **SELECTIONS ...**

Nous allons d'abord définir une fenêtre de sélection permettant à l'utilisateur de rentrer une chaîne de caractères représentant la sélection qu'il veut effectuer. Par exemple « CLASSE = '2B' » ou « COMMUNE = 'Beaupréau' » ou encore « CLASSE < '2E' ».

Dans la **barrette Standard** utiliser une fenêtre **Edit** et la placer sur la boîte de groupe **SELECTIONS...** Vider le Champ **Text** (il est **Edit1** par défaut).

Il faut pouvoir récupérer la phrase tapée dans une **Variable Globale** de Type **STRING**. Nous allons appeler cette variable **SELECT**.

Allons dans la fiche Pascal **Fic2nd**. En parcourant le code on trouve dans la partie **VAR** une seule variable qui est **Form1** (c'est la variable associée à la fiche principale). Ajoutons alors notre variable globale en tapant une nouvelle ligne **SELECT : STRING ;** La partie **VAR** de l'application doit donc être maintenant :

VAR

Form1 : TForm1 ;

Select : **STRING** ;

Revenons maintenant à la fiche et par un double clic tapons la procédure associée à la fenêtre **Edit1** :

procedure TForm1.Edit1Change(Sender: TObject);

begin

Select:=Edit1.Text;

end;

Ajouter le mot **Filtre** devant la fenêtre **Edit1**.

Ajouter alors un bouton **GO** qui va nous permettre de lancer la sélection utilisant le filtre tapé dans la fenêtre Edit1.

Associons à ce bouton la procédure :

```

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
  Table1.Filtered:=TRUE;  {prépare la base à l'opération de filtrage}
  Table1.Filter:=Select;   {lance le filtrage avec la chaîne Select}
end;

```

Relançons le programme et essayons une sélection : tapons dans la case d'édition réservée au filtre « **CLASSE = '2A'** » et cliquons sur le bouton **GO** : seuls les élèves de la classe 2A s'affichent dans le tableau. Attention au filtre (Risque de plantage !) le premier mot CLASSE est un nom de champ alors que 2A est un texte et doit être **entre guillemets**. Une instruction très puissante sous Delphi permet de **gérer les exceptions** (plantages) . Il s'agit de l'instruction

```

TRY
  {Bloc d'instructions 1}
EXCEPT
  {Bloc d'instructions 2}
END ;

```

Si aucune erreur ne se produit lors de l'exécution du Bloc 1 le Bloc 2 est ignoré , par contre si une erreur se produit lors de l'exécution du Bloc 1 alors le programme n'effectue pas cette instruction et passe directement la main au Bloc 2 qui doit gérer l'erreur. Le code devient donc le suivant :

```

procedure TForm1.BitBtn3Click(Sender: TObject);
VAR St : ARRAY[0..100] OF CHAR;  {Variable de type Pointeur Chaîne}
begin
  TRY
    Table1.Filter:=Select;
    Table1.Filtered:=TRUE;
  EXCEPT
    StrCopy(St,'Erreur de syntaxe. '+#10+#13);  {#10 et #13 pour un retour à la ligne}
    StrCat(St,'Utiliser la syntaxe SEXE = "M" pour sélectionner les hommes. ');
    Application.MessageBox(St,'ATTENTION !',mb_OK);
    {Affiche une boîte de message dont le texte est St, le titre 'ATTENTION ! ' et avec le
    seul bouton OK}
    Table1.Filter:="";  {vide le filtre}
    Table1.Filtered:=FALSE; {annule le filtrage}
  END;
end;

```

*L'instruction **TRY** ne fonctionne pas lorsque le sous trouve sous Delphi en mode **compilation**, elle fonctionnera lorsque l'on lancera le programme de façon indépendante.*

Il faut maintenant un bouton permettant **d'annuler la sélection** pour revenir à la liste complète.

Créons ce bouton (Titre **Annuler** et Glyph **Undo**), et associons lui la procédure

```

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
  Table1.Filtered:=FALSE;  {annule les opérations de filtrage}

```

end;

j- Un compteur

Il est important de connaître le **nombre de fiches** (total ou avec sélection).

Créons une fenêtre d'édition **MaskEdit1** en utilisant dans la **barrette Supplément MaskEdit**. Régler la **propriété ReadOnly** sur **TRUE** pour qu'il soit uniquement en lecture et ne puisse être modifié par l'utilisateur.

Il faut maintenant insérer dans ce masque d'édition le nombre d'enregistrements de la table (**Table1.RecordCount**) et ceci à chaque fois que quelque chose a bougé dans la table. Cliquer sur **Table1** pour en ouvrir l'**inspecteur d'objets**. Dans l'**événement AfterScroll** taper le nom de procédure **Mise_a_jour**, double-cliquer et remplir la procédure de la façon suivante :

```
procedure TForm1.Mise_a_Jour(DataSet: TDataSet);  
VAR St : STRING;  
begin  
  Str(Table1.RecordCount,St); {transforme le nombre d'enregistrements en chaîne dans St}  
  MaskEdit1.EditText:=St; {place la chaîne St dans le masque d'édition}  
end;
```

Cette procédure sera appelée chaque fois que quelque chose aura bougé dans le tableau (**AfterScroll**) et le compteur sera donc modifié en permanence.

Ajouter le texte '**Nombre de fiches**' devant la fenêtre MaskEdit1.

k- Imprimer les fiches

Nous allons fabriquer une nouvelle fiche qui n'apparaîtra pas à l'écran mais qui va nous permettre de préparer le format d'affichage pour l'impression de l'ensemble de la base.

Demander **Fichiers-Nouvelle Fiche**. Nommer **impression** la nouvelle fiche créée par Delphi. Sur cette fiche nous allons poser tous les composants que nous voulons imprimer.

Dans la **barrette Qrep** choisir un composant **QuickRep** et le placer sur la nouvelle fiche. A l'aide du **bouton droit** de la souris et en cliquant sur l'objet **QuickRep** faire un **Zoom Arrière** pour visualiser l'ensemble du composant. Régler la propriété **DataSet** sur **Form1.Table1** pour associer ce rapport à notre base de données.

Dans la **barrette Qrep** Choisir un composant **Qrband** et le placer sur l'objet **QuickRep**. Cet objet est nommé **Qrband1**. Régler la propriété **BandType** sur **Rbdetail** pour afficher tous les champs les uns à la suite des autres.

Poser sur ce composant **QuickRep1** un composant **QrDBtext**. Régler la propriété **DataSet** sur **Form1.Table1** pour lier ce composant à notre base. Régler le composant **DataField** sur le champ **NOM** (il suffit de faire dérouler et choisir).

Recommencer et poser un autre composant **QrDBtext** dont on règle **DataSet** sur **Form1.table1** et **DataField** sur **PRENOM**. Mettre sa propriété **Left** à **90**.

Recommencer pour placer le champ **COMMUNE** à la suite (**Left = 170**).

Recommencer pour placer le champ **CLASSE** à la suite (**Left = 310**).

Recommencer pour placer le champ **SEXE** à la suite (**Left = 370**).

Dérouler le **Menu contextuel** en cliquant sur **QuickRep1** avec le **bouton droit** et demander une **prévisualisation**. La fenêtre de dialogue **aperçu avant impression** s'ouvre et vous montre la page telle qu'elle sera imprimée.

Refermer cette page.

Il faut maintenant ajouter une barre de titre à la page comportant les mots NOM, PRENOM, COMMUNE, CLASSE, SEXE.

Ajouter un autre composant **Qrband** (**Qrband2**). Régler la propriété **bandType** sur **RbTitle**. Régler la propriété **Height** à **26** et enfin dérouler la propriété **Frame** et mettre **DrawBottom** à **TRUE** pour tracer une ligne **SOUS** la bande **Title**.

Sur cette bande **Title** placer cinq objet **Qrlabel** et nommer les champs **Caption** respectivement **NOM**, **PRENOM** (**Left = 90**), **COMMUNE** (**Left = 170**), **CLASSE** (**Left = 310**) et **SEXE** (**Left = 370**).

Il faut maintenant ajouter un Bouton sur notre Fiche « Base seconde » pour avoir accès à la fenêtre de prévisualisation et pouvoir imprimer notre travail.

Pour ce faire il suffit de créer un bouton **Imprimer** et de lui lier le script :

```
procedure TForm1.BitBtn5Click(Sender: TObject);  
begin  
    Form2.QuickRep1.preview;  
end;
```

Lors de la première exécution du programme Delphi nous demande si nous voulons ajouter l'unité liée à Form2 dans les clauses Uses de l'application : nous répondons **oui** et nous pouvons tester le programme sous sa version presque définitive.

Ajouter enfin un bouton **Quitter** auquel on associe le script :

```
procedure TForm1.BitBtn6Click(Sender: TObject);  
begin  
    Application.Terminate;  
end;
```

I- Ajout de menu déroulant

Pour donner un peu d'allure à notre application nous allons lui ajouter un menu comme dans toute bonne application Windows.

Poser un objet **MainMenu** (Barrette Standard) sur la fiche (n'importe où). Double cliquer sur cet objet pour ouvrir la fenêtre de conception du menu. Taper le titre **Fichiers** puis le sous titre **Imprimer**. Régler l'événement **OnClick** de Imprimer sur **BitBtn5Click**. Ajouter (toujours dans le menu Fichiers) le sous titre **Quitter** (Régler le sur **BitBtn6Click**). Ajouter alors après le menu Fichiers un menu **Editer** contenant les rubriques **Ajouter** (lié à la procédure **AjouterClick**) et **Supprimer** (lié à la procédure **SupprimerClick**). Ajouter enfin (après le menu Editer) un menu **TRI** dans lequel on propose les deux tris : **suivant NOM** et **suivant PRENOM**.

Attention, l'ajout d'un menu décale les objets vers le bas. Si nécessaires nous les remontons.

Il ne reste plus qu'à tester notre gestionnaire de fiches grandeur nature.

m- Les fichiers nécessaires à l'exécution du programme

A l'aide de l'**Explorateur** de windows aller dans le répertoire **\DELPHI\FIC2ND**. Dans ce répertoire on y trouve les fichiers suivants :

- *.PAS fichiers pascal dans lesquels sont toutes les procédures
- *.DCU Unités compilées utiles au projet
- *.DPR fichier de projet. C'est le fichier principal, il contient toutes les informations relatives au projet.
- *.RES fichier de ressources (contient le menu déroulant par exemple...)
- *.EXE fichier exécutable. Ce fichiers est indépendant et peut être exécuter sans l'aide des autres fichiers. Seuls les fichiers propre à la base de donnée lui sont nécessaires.
- *.DBF fichier contenant le nom des champs et index de notre base de donnée.
- *.MDX fichier contenant toute les fiches créées.

Seuls les **trois derniers fichiers** sont nécessaires à l'exécution du programme. Le fichier *.EXE permet de lancer notre application. Les fichiers *.DBF et *.MDX contiennent la base au format dBase et peuvent être utilisés par toutes les applications connaissant ce format. Vous pouvez ouvrir le fichier *.DBF avec **excel** par exemple.

Essayons de lancer notre application en cliquant sur le fichier *.EXE. Cela fonctionne parfaitement.

Pour installer l'application sur une autre machine il suffit de copier les trois fichiers *.EXE, *.DBF et *.MDX. Même si la machine ne contient pas Delphi elle pourra exécuter l'application.

n- Taille du fichier exécutable

Regarder dans l'explorateur la taille du fichier *.EXE. Ce fichier a une taille importante car il contient toutes les informations nécessaires au débogage de l'application. Il contient en fait les adresses de chacune des procédures pour pouvoir indiquer au développeur d'où vient l'erreur en cas de plantage. Une fois la mise au point du programme faite il n'est plus nécessaire de garder ces informations qui surchargent inutilement le code.

Retourner sous Delphi. Dans le menu **Projet-Options** cliquer sur l'onglet **Lieur** et désactiver **Inclure info de débogage TD32**. Après avoir refermé la fenêtre de dialogue des options demander **Tout construire** dans le menu **Projet**. Revenir alors dans le répertoire **DELPHI\FIC2ND** avec l'**explorateur windows** et regarder la nouvelle taille du fichier exécutable : elle a été divisée par trois !

Copie d'écran de l'application BASE SECONDE

