Delphi

Dephi est une marque déposée
Borland Software Corporation

Sommaire

I. Créer votre premier projet

A. Programme console

```
program Projet4;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var
  s:string;

begin
  WriteLn('Hello the World :');
  ReadLn;
end.
```

B. Projet CLX ?

La norme CLX permet, en fait, de concevoir des programmes à la fois pour Linux et Windows. Borland propose un compilateur de projets CLX sous Linux : Kylix. En mode CLX, plutôt que de faire appel aux API Windows, spécifiques à cette plate-forme, les programmes font appel aux librairies Qt définies et conçues par TrollTech, librairies qui sont à l'origine du développement de Kde. Avec l'abandon de Kylix par Borland, la norme CLX n'est plus maintenue par Borland. Vous pouvez vous réfugier vers FreeCLX et Lazarus, un environnement de développement intégré sous Windows et Linux.

C. Sauvegarde

Les éléments du projet sont le projet lui-même ou programme principal, dont l'extension est dpr. Les unités ont l'extension pas. Attention, dans un projet, l'unité ne peut pas avoir le même nom que le projet.
Au fur et à mesure de l'écriture de votre code, pensez à sauvegarder tous les éléments par le raccourci CTRL+SHIFT+S.
Les fichiers dof et cfg contiennent respectivement les propriétés du projet et les options de compilation.

D. Compilation

Lors de la compilation, les unités sont compilées sous forme de fichiers Dcu alors que le projet est compilé et linké au format exe.

E. Bascule entre les fenêtres de code et unité

La touche F12 vous permet de basculer de l'une à l'autre.

F. L'inspecteur d'objets

L'inspecteur d'objets référence les propriétés attachées à l'objet sélectionné. F11 est une touche bascule tout aussi pratique et majeure que F12. Les propriétés des objets que vous avez modifiées sont stockées dans un fichier Dfm.

G. La palette des composants

Borland avec Delphi a été l'un des tous premiers éditeurs à introduire la notion de palette de composants à onglets. Ce système est beaucoup plus pratique que les IDE (Integrated Development Environment) concurrents tels que Visual Studio.

H. Interaction code et composants

A partir de la palette standard, dessinez une zone de texte (TLabel). Dans l'inspecteur d'objet, nommez-la à l'aide de la propriété Name.
Double-cliquez ensuite sur le formulaire. Vous basculez alors dans la fenêtre de code :

```
procedure TFormMain.FormCreate(Sender: TObject);
begin
  Self.LabelHello.Caption:='Hello the World !';
end;
```

I. Utilisation des unités

1. Mode Console

a) L'unité

```
unit biblio;

interface
  procedure hello();

implementation
  procedure hello();
  begin
    WriteLn('Hello the World !');
    ReadLn;
  end;

end.
```

b) Le programme

```
program Project6;

{$APPTYPE CONSOLE}

uses
  SysUtils,
  biblio;

begin
  Hello;
end.
```

2. Mode graphique

a) Le programme

```
program projet7;

uses
  Forms,Dialogs,
  formulaire in 'formulaire.pas' {FormHello};

{$R *.res}
var
  S:string;
begin
  ShowMessage('Hello the World');
  S:=InputBox('Boîte de saisie','Invite','Hello the World');
  Application.MessageBox(PChar(S),'Look');
  Application.Initialize;
  Application.CreateForm(TFormHello, FormHello);
  Application.Run;
end.
```

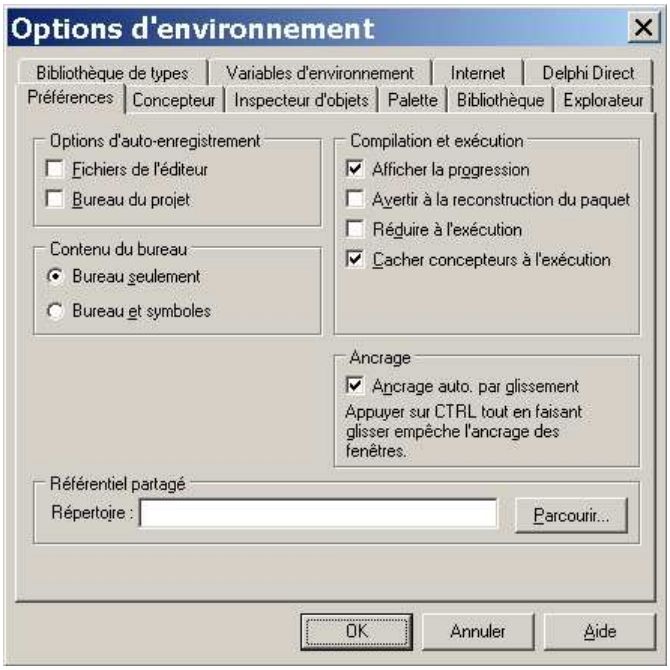
b) L'unité

```
unit formulaire;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;

type
  TFormHello = class(TForm)
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;
```

```
var
  FormHello: TFormHello;

implementation

{$R *.dfm}

end.
```

J. Options

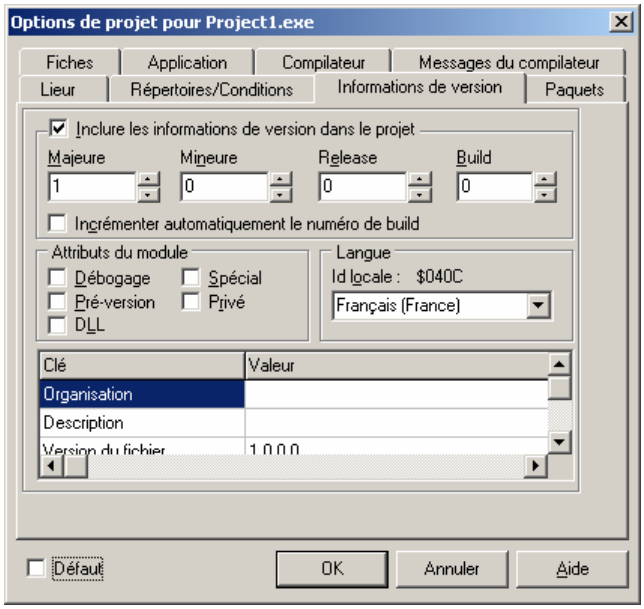
1. Configurer le comportement du compilateur



Pour disposer d'informations lors de la compilation, allez dans Outils | Options d'environnement. Cochez alors dans l'onglet Préférences dans le bloc Compilation et exécution Afficher la progression.

Lors de la compilation, vous pouvez voir d'un coup d'oeil le nombre d'erreurs encore présentes dans votre code.

2. Options du projet : le fichier Dof

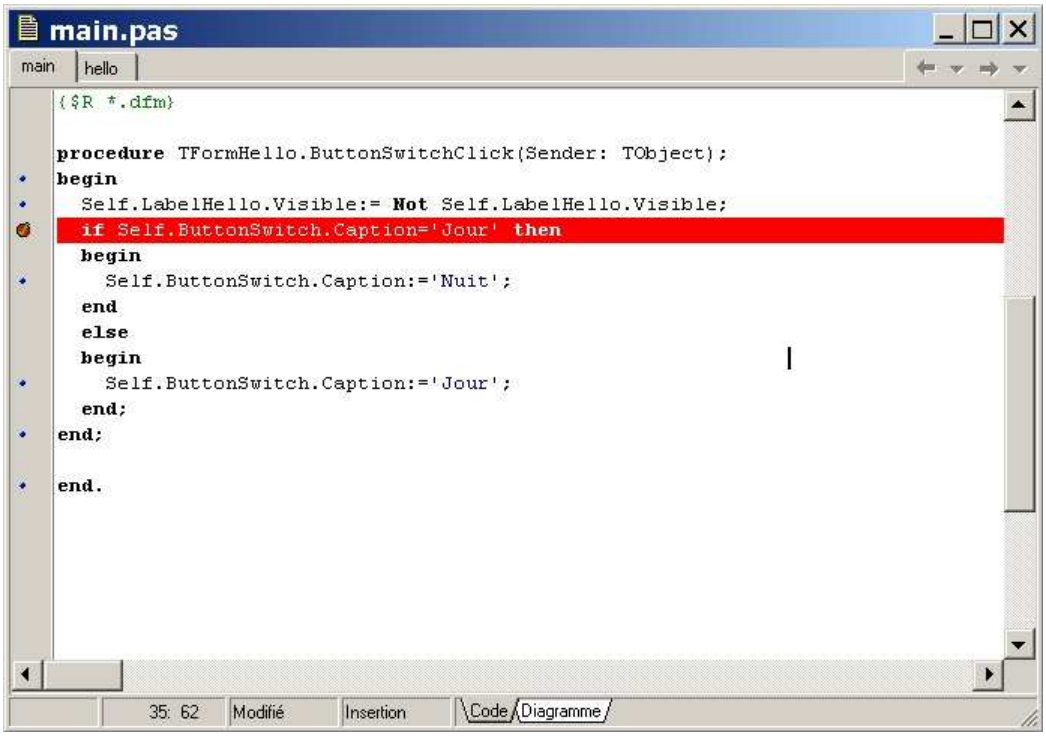


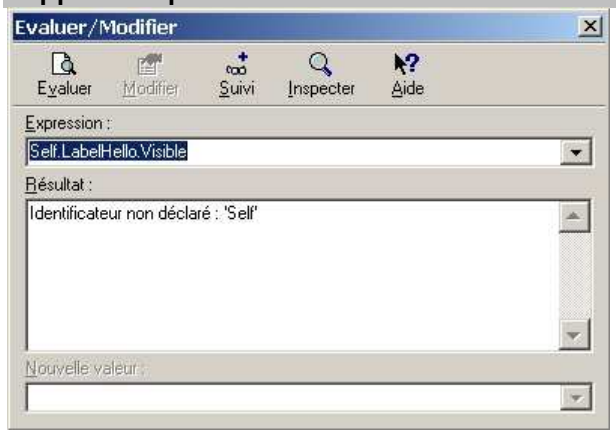
Ce fichier se situe dans le répertoire du projet. Il contient les options du projet. Vous pouvez renseigner ces informations à partir de l'interface graphique : Projet|Options du projet.

```
[Version Info Keys]
CompanyName=Denis Szalkowski
FileDescription=Hello
FileVersion=0.1.0.0
InternalName=Hello
LegalCopyright=Denis Szalkowski
LegalTrademarks=Denis Szalkowski
OriginalFilename=hello.exe
ProductName=Hello
ProductVersion=0.1.0.0
Comments=Un programme inutile
```

K. Débogage

Pour insérer, un point d'arrêt dans votre programme, cliquez en marge de la ligne où vous souhaitez arrêter le code lors de l'exécution. Pour le retirer, cliquez au même endroit.





Pour évaluer une expression modifiée lors de l'exécution du code, sélectionnez l'expression et allez dans Exécuter | Évaluer/Modifier.

L. Eléments de POO Programmation Orientée Objet

1. L'unité/classe texte

```
unit texte;

interface
  type TTexte=class
    private
    protected
    public
      procedure Afficher(Message:string);
      constructor Create();overload;
      destructor Destroy;override;
    published
  end;

implementation
  constructor TTexte.Create();
  begin
    WriteLn('Je me construis !');
  end;
  destructor TTexte.Destroy;
  begin
    WriteLn('Je me détruis !');
  end;
  procedure TTexte.Afficher(Message:string);
  begin
    WriteLn(Message);
    ReadLn;
  end;
end.
```

2. Le programme

```
program poo;

{$APPTYPE CONSOLE}

uses
  SysUtils,
  texte in 'texte.pas';
var
  oTexte:Ttexte;
begin
  oTexte.Create;
  oTexte.Afficher('Hello the World');
  oTexte.Free;
  oTexte:=nil;
end.
```

M. Directives de compilation conditionnelles

Delphi intègre trois symboles pour isoler du code selon les plates-formes sur lequel il s'exécute : MSWINDOWS, LINUX et CLR.

```
unit texte;
```

```
interface
  type TTexte=class
    private
    protected
    public
      procedure Afficher(Message:string);
    published
  end;

implementation
  procedure TTexte.Afficher(Message:string);
  begin
    WriteLn(Message);
    {$IFDEF LINUX}
    WriteLn('Vous êtes sous Linux');
    {$ENDIF}
    {$IFDEF MSWINDOWS}
    WriteLn('Vous êtes sous Windows');
    {$ENDIF}
    ReadLn;
  end;

end.
```

II. Éléments du langage

A. Mots réservés

and	exports	mod	shr
array	file	nil	string
as	finalization	not	then
asm	finally	object	threadvar
begin	for	of	to
case	function	or	try
class	goto	out	type
const	if	packed	unit
constructor	implementation	procedure	until
destructor	in	program	uses
dispinterface	inherited	property	var
div	initialization	raise	while
do	inline	record	with
downto	interface	repeat	xor
else	is	resourcestring	
end	label	set	
except	library	shl	

B. Commentaires

```
{
	bloc de commentaires sur plusieurs lignes
}
//Préfixe un commentaire sur une ligne
(*
	bloc de commentaires sur plusieurs lignes
*)
```

C. Opérateurs (extrait de l'aide en ligne)

1. Opérateurs arithmétiques binaires

Opérateur	Opération	Types d'opérande	Type du résultat	Exemple
+	addition	entier, réel	entier, réel	X + Y
-	soustraction	entier, réel	entier, réel	Resultat - 1
*	multiplication	entier, réel	entier, réel	P * TauxInteret
/	division réelle	entier, réel	réel	X / 2
div	division entière	entier	entier	Total div TailleUnit
mod	reste	entier	entier	Y mod 6

2. Opérateurs arithmétiques unaires

Opérateur	Opération	Types des opérandes	Type du résultat	Exemple
+ (unaire)	identité signe	entier, réel	entier, réel	+7
- (unaire)	négation signe	entier, réel	entier, réel	-X

D. Les constantes

1. Etendues

Etendue de la constante (hexadécimal)	Etendue de la constante (décimal)	Type
-$8000000000000000..-$80000001	-2^63..-2147483649	Int64
-$80000000..-$8001	-2147483648..-32769	Integer
-$8000..-$81	-32768..-129	Smallint

-$80..-1	-128..-1	Shortint
0..$7F	0..127	0..127
$80..$FF	128..255	Byte
$0100..$7FFF	256..32767	0..32767
$8000..$FFFF	32768..65535	Word
$10000..$7FFFFFFF	65536..2147483647	0..2147483647
$80000000..$FFFFFFFF	2147483648..4294967295	Cardinal
$100000000..$7FFFFFFFFFFFFFFF	4294967296..2^63-1	Int64

2. Exemples

```
const
  Min = 0;
  Max = 100;
  Centre = (Max - Min) div 2;
  Beta = Chr(225);
  NbCars = Ord('Z') - Ord('A') + 1;
  Message = 'Mémoire insuffisante';
  ErrStr = ' Erreur : ' + Message + '. ';
  ErrPos = 80 - Length(ErrStr) div 2;
  Ln10 = 2.3025850929940456840;
  Ln10R = 1 / Ln10;
  Numerique = ['0'..'9'];
  Alpha = ['A'..'Z', 'a'..'z'];
  AlphaNum = Alpha + Numerique;
```

E. Types de données

1. Les types simples

a) Les scalaires

Fonction	Paramètre	Valeur renvoyée	Remarque
Ord	expression scalaire	rang de la valeur de l'expression	Ne prend pas d'arguments Int64.
Pred	expression scalaire	prédécesseur de la valeur de l'expression	
Succ	expression scalaire	successeur de la valeur de l'expression	
High	identificateur de type scalaire ou variable de type scalaire	plus grande valeur du type	Opère également sur les types de chaîne courte et les tableaux.

Low identificateur de type scalaire ou variable de type scalaire plus petite valeur du type Opère également sur les types de chaîne courte et les tableaux.

b) Les entiers

Type	Etendue	Format
Integer	-2147483648..2147483647	32 bits signé
Cardinal	0..4294967295	32 bits non signé
Shortint	-128..127	8 bits signé
Smallint	-32768..32767	16 bits signé
Longint	-2147483648..2147483647	32 bits signé
Int64	-2^63..2^63-1	64 bits signé
Byte	0..255	8 bits non signé
Word	0..65535	16 bits non signé
Longword	0..4294967295	32 bits non signé

c) Le type Char

d) Les booléens

Booléen	ByteBool, WordBool, LongBool, boolean
False < True	False <> True
Ord(False) = 0	Ord(False) = 0
Ord(True) = 1	Ord(True) <> 0
Succ(False) = True	Succ(False) = True
Pred(True) = False	Pred(False) = True

e) Les énumérations

```
type nomType = (val1, ..., valn)
type nomType = (val1=valeur1, ..., valn=valeurn)
```

f) Les intervalles

```
type TCouleurs = (Rouge, Bleu, Vert, Jaune, Orange, Violet, Blanc, Noir);
type TMesCouleurs = Vert..Blanc;
type
 DesNombres = -128..127;
 Majs = 'A'..'Z';
```

g) Les réels

Type	Etendue	Chiffres significatifs	Taille en octets
Real48	2.9 x 10^-39 .. 1.7 x 10^38	11-12	6
Single	1.5 x 10^-45 .. 3.4 x 10^38	7-8	4
Double	5.0 x 10^-324 .. 1.7 x 10^308	15-16	8
Extended	3.6 x 10^-4951 .. 1.1 x 10^4932	19-20	10
Comp	-2^63+1 .. 2^63 -1	19-20	8
Currency	-922337203685477.5808.. 922337203685477.5807	19-20	8
Real	5.0 x 10^-324 .. 1.7 x 10^308	15-16	8

2. Les chaînes ou String

Type	Longueur maximum	Mémoire nécessaire	Utilisation
ShortString	255 caractères	de 2 à 256 octets	Compatibilité ascendante
AnsiString	~2^31 caractères	de 4 octets à 2Go	Caractère sur 8 bits (ANSI), DBCS ANSI, MBCS ANSI, etc.
WideString	~2^30 caractères	de 4 octets à 2Go	Caractères Unicode ; serveurs muli-utilisateurs et applications multilingues

3. Les types structurés

a) Les ensembles

```
type
  TCertainsEntiers = 1..250;
  TEnsembleEntiers = set of TCertainsEntiers;
type TEnsembleEntiers = set of 1..250;
var Ens1, Ens2: TEnsembleEntiers;
 ...
Ens1 := [1, 3, 5, 7, 9];
Ens2 := [2, 4, 6, 8, 10]
var MonEnsemble: set of 'a'..'z';
 ...
MonEnsemble := ['a','b','c'];
set of Byte
set of (Trefle, Carreau, Coeur, Pique)
set of Char;
if 'a' in MonEnsemble then ...;
```

b) Les tableaux statiques

```
var MonTableau: array[1..100] of Char;
type TMatrice = array[1..10] of array[1..50] of Real;
type TMatrice = array[1..10, 1..50] of Real;
```

c) Les tableaux dynamiques

```
var
  A, B: array of Integer;
begin
  SetLength(A, 1);
  SetLength(B, 1);
  A[0] := 2;
  B[0] := 2;
end;
```

d) Les enregistrements

```
type
  TDateRec = record
    Annee: Integer;
    Mois: (Jan, Fev, Mar, Avr, Mai, Jun,
           Jul, Aou, Sep, Oct, Nov, Dec);
    Jour: 1..31;
  end;
var Record1, Record2: TDateRec;
Record1.Annee := 1922;
Record1.Mois := Nov;
Record1.Jour := 26;
with Record1 do
begin
  Annee := 1922;
  Mois := Nov;
  Jour := 26;
end;
```

e) Les fichiers

```
type
  EntreeRepertoire = record
    Prenom, Nom: string[20];
    Telephone: string[15];
    Liste: Boolean;
  end;
  Annuaire = file of EntreeRepertoire;
var List1: file of EntreeRepertoire;
var FichierDonnees: file;
```

f) Les variants

CibleSource	entier	réel	chaîne	booléen
entier	convertit au format entier	convertit en réel	convertit en représentation chaîne	renvoie False si la valeur est 0 et True sinon
réel	arrondit à l'entier le plus proche	convertit au format réel	convertit en chaîne en utilisant les paramètres régionaux	renvoie False si la valeur est 0 et True sinon
chaîne	convertit en entier en tronquant si nécessaire. Déclenche une exception si la chaîne n'est pas numérique.	convertit en réel en utilisant les paramètres régionaux. Déclenche une exception si la chaîne n'est pas numérique.	convertit au format chaîne/caractère	renvoie False si la chaîne est "false" (pas de différence majuscule/minuscule) ou une chaîne numérique qui s'évalue à 0, True si la chaîne est "true" ou une chaîne numérique non nulle ; sinon déclenche une exception

CibleSource	entier	réel	chaîne	booléen
caractère	pareil que pour une chaîne (ci-dessus)	pareil que pour une chaîne (ci-dessus)	pareil que pour une chaîne (ci-dessus)	pareil que pour une chaîne (ci-dessus)
booléen	False = 0, □True = -1 □(255 si Byte)	False = 0, True = -1	False = "0", True = "-1"	False = False, True = True
Unassigned	renvoie 0	renvoie 0	renvoie une chaîne vide	renvoie False
Null	déclenche une exception	déclenche une exception	déclenche une exception	déclenche une exception

g) Les pointeurs

(1) Types de pointeur

Type de pointeur	Pointe sur des variables de type
PAnsiString, PString	AnsiString
PByteArray	TByteArray (déclaré dans SysUtils). Utilisé pour transtyper dynamiquement de la mémoire allouée pour les tableaux.
PCurrency, PDouble, PExtended, PSingle	Currency, Double, Extended, Single
PInteger	Integer
POleVariant	OleVariant
PShortString	ShortString. Utilisé pour adapter du code ancien utilisant le type PString.
PTextBuf	TTextBuf (déclaré dans SysUtils). TTextBuf est le type interne de tampon d'un enregistrement fichier TTextRec.
PVarRec	TVarRec (déclaré dans System)
PVariant	Variant
PWideString	WideString
PWordArray	TWordArray (déclaré dans SysUtils). Utilisé pour transtyper dynamiquement de la mémoire allouée pour des tableaux de valeurs sur deux octets.

h) Exemple

```
program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var
X, Y: Integer;   // X et Y sont des variables Integer
//P: ^Integer;     // P pointe sur un Integer
P:PInteger;
begin
X := 17;         // affecte une valeur à X
P := @X;         // affecte l'adresse de X à P
Y := P^;         // déréférence P; affecte le résultat à Y
Writeln('X : ',X);
Writeln('Y : ',Y);
ReadLn;
end.
```

F. Procédures et fonctions

G. Structures de contrôle

1. Les conditionnelles

a) If

```
if condition1 then
begin
…
end
Else if condition2
Begin
…
End
…
Else
Begin
end
end;
```

b) Case

```
case expression of
  cas1: …
  cas2: …
else
…
end;
```

2. Les répétitives

a) While

```
while Condition  do
begin
…
end;
```

b) Repeat

```
Repeat
…
Until Condition
```

c) For

```
For expression:=debut to/downto fin do
Begin
…
End;
```

d) Break

```
program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;
var
  I:integer;
begin
  { TODO -oUser -cConsole Main : placez le code ici }
  for I:=1 to 100 do
  begin
    WriteLn(I);
    if I=5 then
    begin
      break;
    end;
  end;
  ReadLn;
end.
```

3. Gestion d'erreur : try catch except finally

a) Classes d'exception

Classe d'exception	Description
EAbort	Interrompt une séquence d'événements sans afficher de boîte de dialogue de message d'erreur.
EAccessViolation	Vérifie la présence d'erreurs d'accès mémoire.
EBitsError	Empêche les tentatives incorrectes d'accès à un tableau booléen.
EComponentError	Signale une tentative incorrecte de recensement ou de modification du nom d'un composant.
EConvertError	Indique des erreurs de conversion d'objet ou de chaîne.
EDatabaseError	Spécifie une erreur d'accès à une base de données.
EDBEditError	Intercepte des données incompatibles avec un masque spécifié.
EDivByZero	Intercepte des erreurs de division par zéro des entiers.
EExternalException	Indique un code d'exception non reconnu.
EInOutError	Représente une erreur d'E/S de fichier.
EIntOverflow	Spécifie des calculs d'entiers dont les résultats sont trop élevés pour le registre alloué.
EInvalidCast	Vérifie s'il existe des transtypages incorrects.
EInvalidGraphic	Indique une tentative d'utilisation d'un format de fichier graphique incorrect.
EInvalidOperation	Se produit lorsque des opérations incorrectes sont tentées sur un composant.
EInvalidPointer	Se produit suite à des opérations de pointeur incorrect.
EMenuError	Implique un problème dû à un élément de menu.
EOleCtrlError	Détecte des problèmes de liaison avec les contrôles ActiveX.
EOleError	Spécifie des erreurs d'automation OLE.
EPrinterError	Signale une erreur d'impression.
EPropertyError	Se produit suite à des tentatives infructueuses de définition de la valeur d'une propriété.
ERangeError	Indique que la valeur d'un entier est trop élevée pour le type déclaré auquel il est affecté.
ERegistryException	Spécifie des erreurs de registre.
EZeroDivide	Intercepte des erreurs de division par zéro de virgule flottante.

b) Exemple

```
program Project1;

{$APPTYPE CONSOLE}

uses
  SysUtils;
var
  X,Y,Z: single;
begin
  X:=1;Y:=0;Z:=0;
  try
    Z:=X/Y;
  {
  except
    on Exception do
    begin
      WriteLn('Erreur');
    end;
  }
  finally
    WriteLn('Finallement');
  end;
  ReadLn;
end.
```

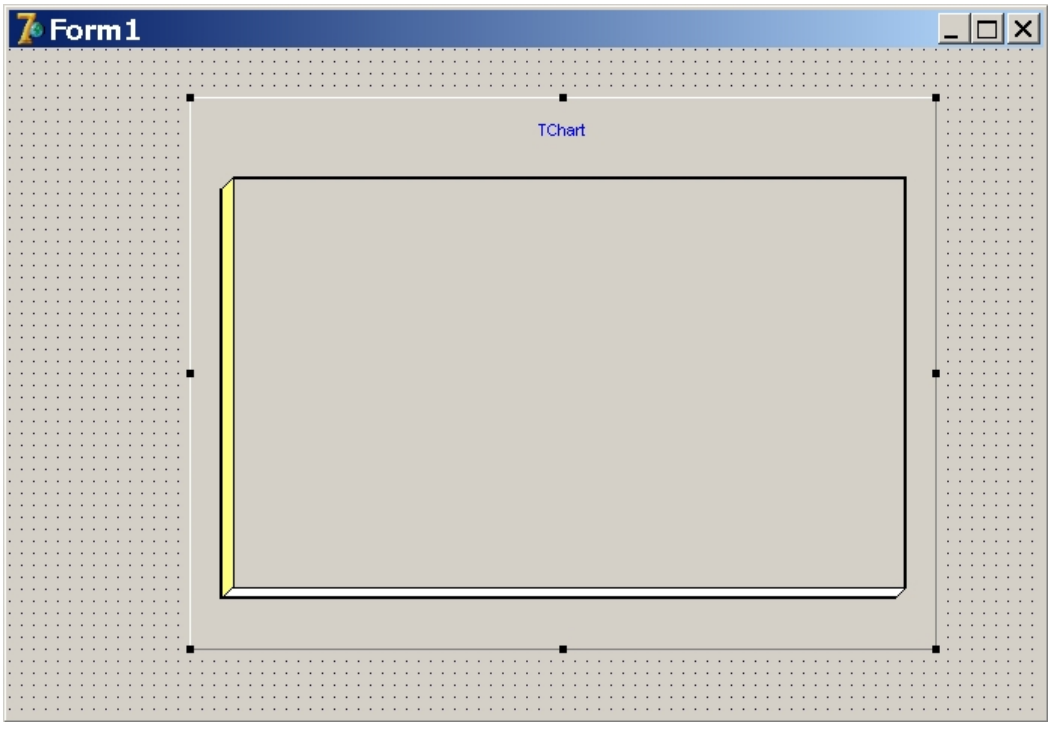
III. Eléments de programmation avancée

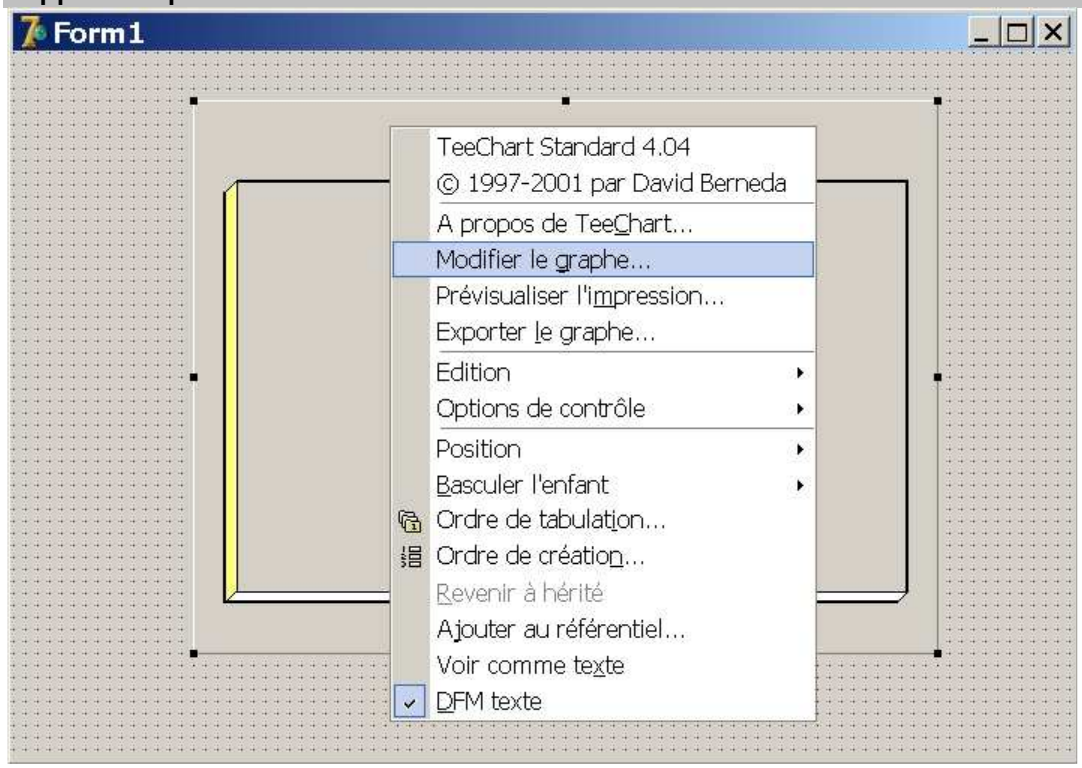
A. La gestion de fichiers

```
program Project5;

{$APPTYPE CONSOLE}

uses
  SysUtils;

var
  F1,F2:TextFile;
  S:String;
begin
  AssignFile(F1,'g:\boot.ini');
  Reset(F1);
  AssignFile(F2,'g:\boot.ini.ori');
  Rewrite(F2);
  while not Eof(F1)do
  begin
    ReadLn(F1,S);
    WriteLn(S);
    WriteLn(F2,S);
  end;
  CloseFile(F2);
  CloseFile(F1);
  ReadLn;
end.
```

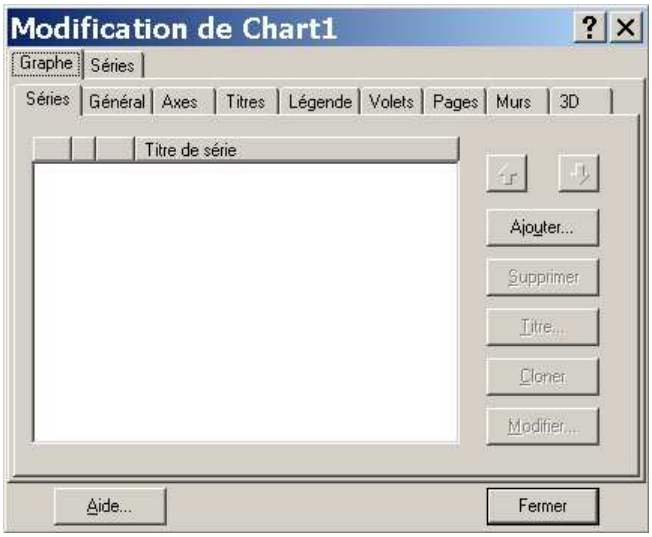
B. Création de graphiques : le composant TChart

Dessinez le composant à partir de la barre de composants Supplément.

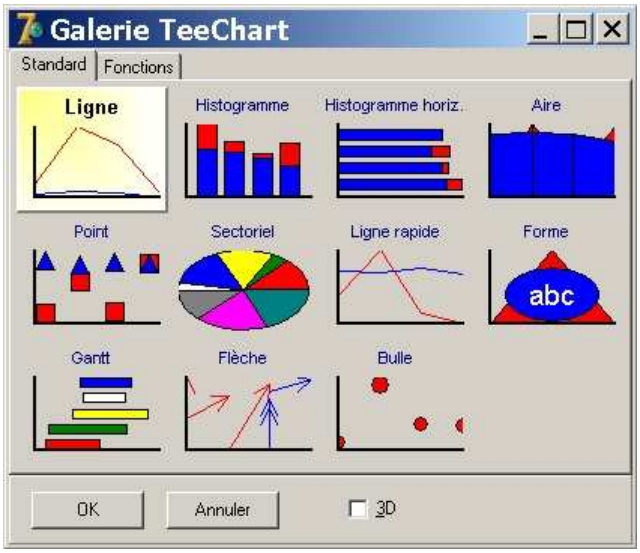




Par un clic droit, éditez le composant.

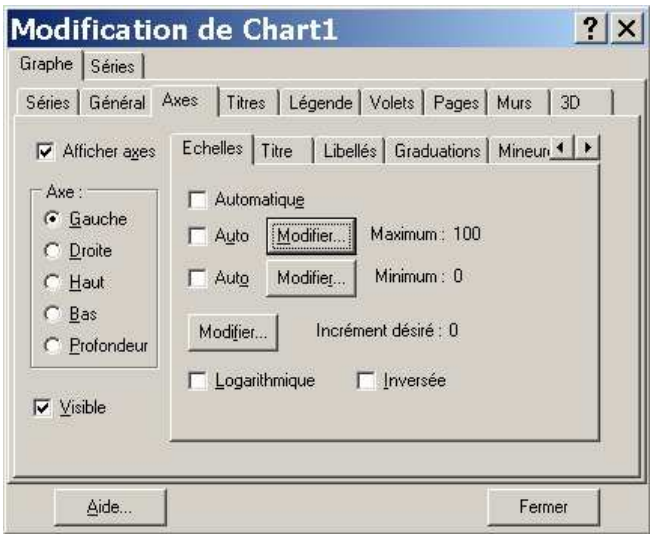


Ajoutez une Série à partir de l'onglet Graphe.



Spécifiez le type de graphique.

Entrez le titre de la série (visuel)



Modifiez l'échelle en fonction des valeurs affichées.

Entrez la valeur maximale de l'échelle.

Pour alimenter la série par le code, vous trouverez un exemple dans l'exemple Biorythme ci-dessous :

```
Self.Series1.Add(BioPhy,DateToStr(DateObservation),clGreen);
Self.Series2.Add(BioEmo,DateToStr(DateObservation),clRed);
Self.Series3.Add(BioInt,DateToStr(DateObservation),clBlue);
```

IV. Applications Sdi - Mdi

A. Présentation

Les applications SDI (Single Device Interface) sont des applications dans lesquelles il n'y a, en général, aucun menu, ni barre d'outils. Ces applications, simples, son souvent réservés à de petits projets.

B. Appel d'une fiche

A partir d'une fiche, pour appeler une autre fiche, il existe deux méthodes.

1. La méthode CreateForm

Dans le programme principal de l'application, vous trouverez l'utilisation de la méthode suivante :

```
Application.Initialize;
Application.CreateForm(TForm1, Form1);
Application.Run;
```

Pour détruire la fiche, vous disposez alors de la méthode *Destroy* que vous pouvez associer à l'événement *Close* lié à la fiche : *Form1.Destroy*
Pour l'application, préférez *Application.Terminate*.

2. Les méthodes Show et ShowModal

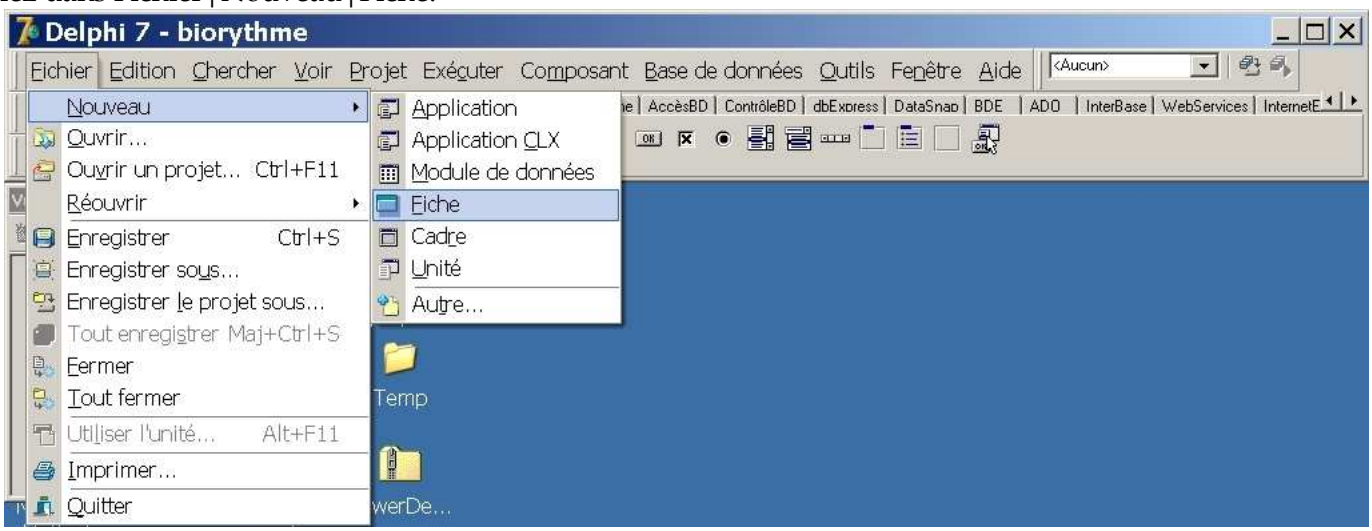
Pour visualiser un formulaire, vous pouvez utiliser aussi la méthode *Show* ou *ShowModal* (impossible de changer de fenêtre dans l'application autre que la Fenêtre appelée). Cela suppose que vous ayez préalablement chargé toutes les fenêtres avec *Application.CreateForm* au préalable. Pour fermer la fiche, utilisez la méthode *Close*. Vous pouvez aussi utiliser la méthode *Hide* pour la masquer.

C. Création d'application Mdi

Ces applications comportent une fenêtre principale (parente) et d'autres fenêtres enfants.

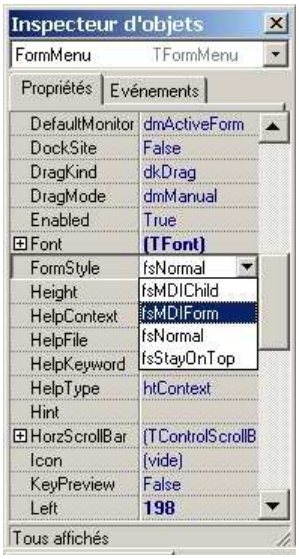
1. Ajouter une fiche à un projet existant

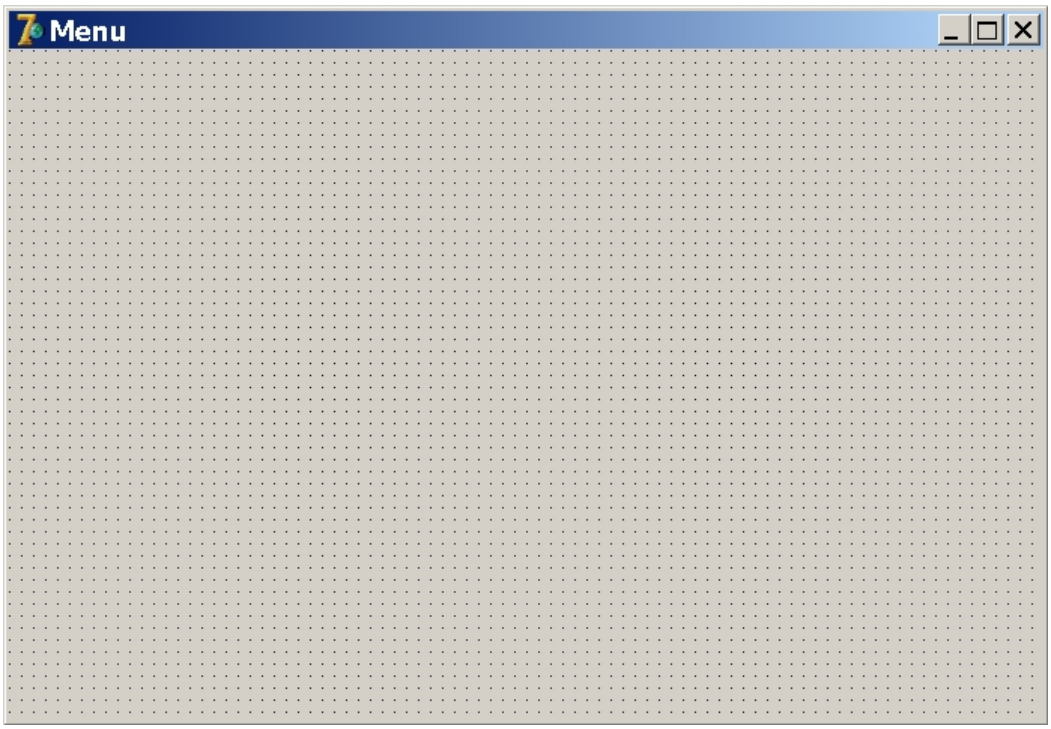
Allez dans Fichier | Nouveau | Fiche.



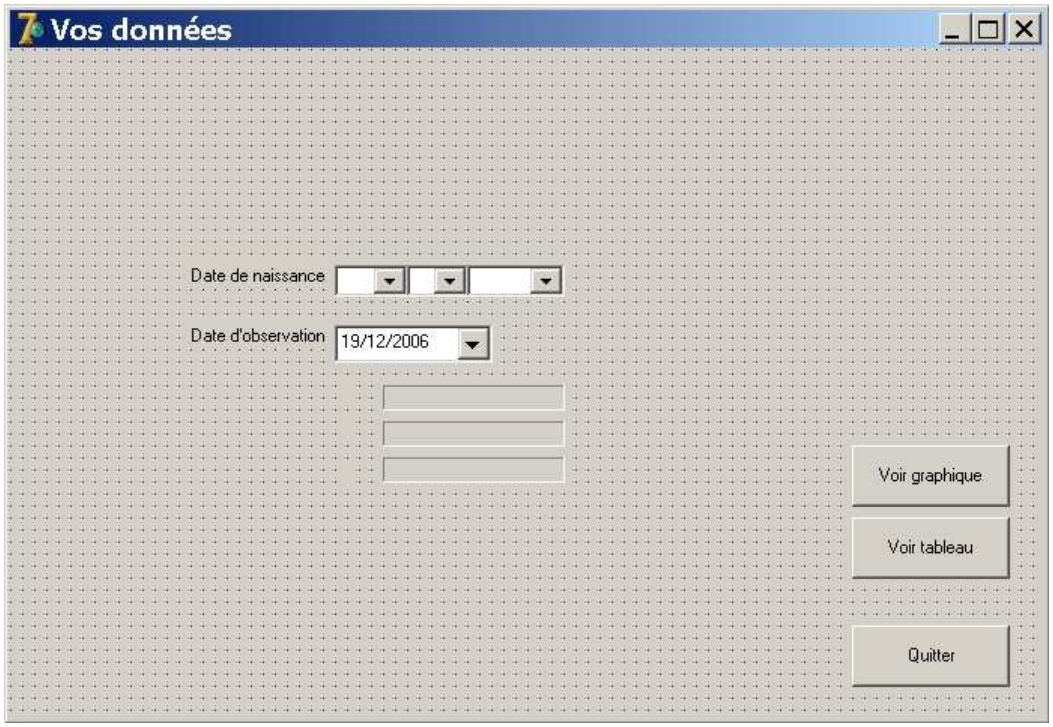
2. Propriétés de la fenêtre principale

A partir de la fiche principale à laquelle vous allez associer le menu. Passez la propriété FormStyle à la valeur fsMDIForm.

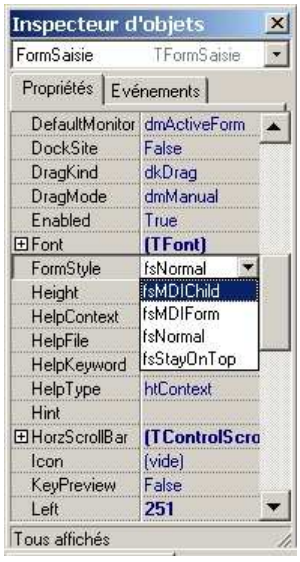




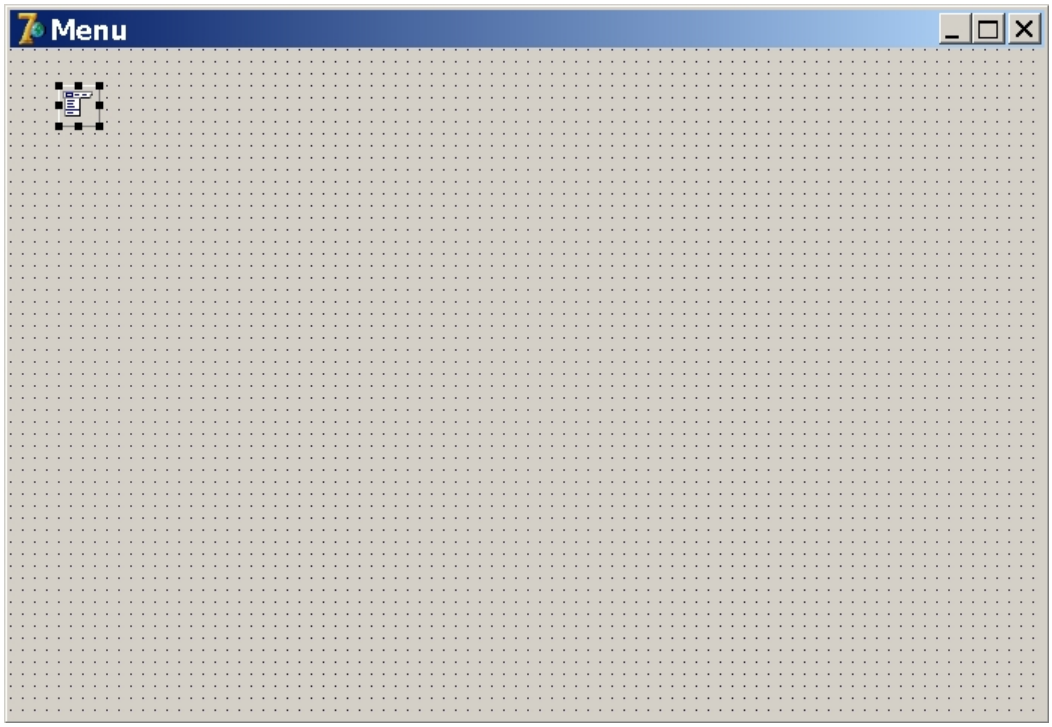
3. Propriétés de la fenêtre enfant



Pour les fenêtres enfants, cette propriété doit être passée à fsMDIChild.



4. Création d'un menu



A partir de la palette des composants Standard, ajoutez le composant Menu. Donnez un nom à ce menu.

Double cliquez alors sur le composant pour éditer le menu. Vous arrivez dans L'éditeur de menu.

Dans les propriétés de l'élément de menu, auquel vous devez donner un nom. La propriété Enabled permet de griser l'élément de menu.

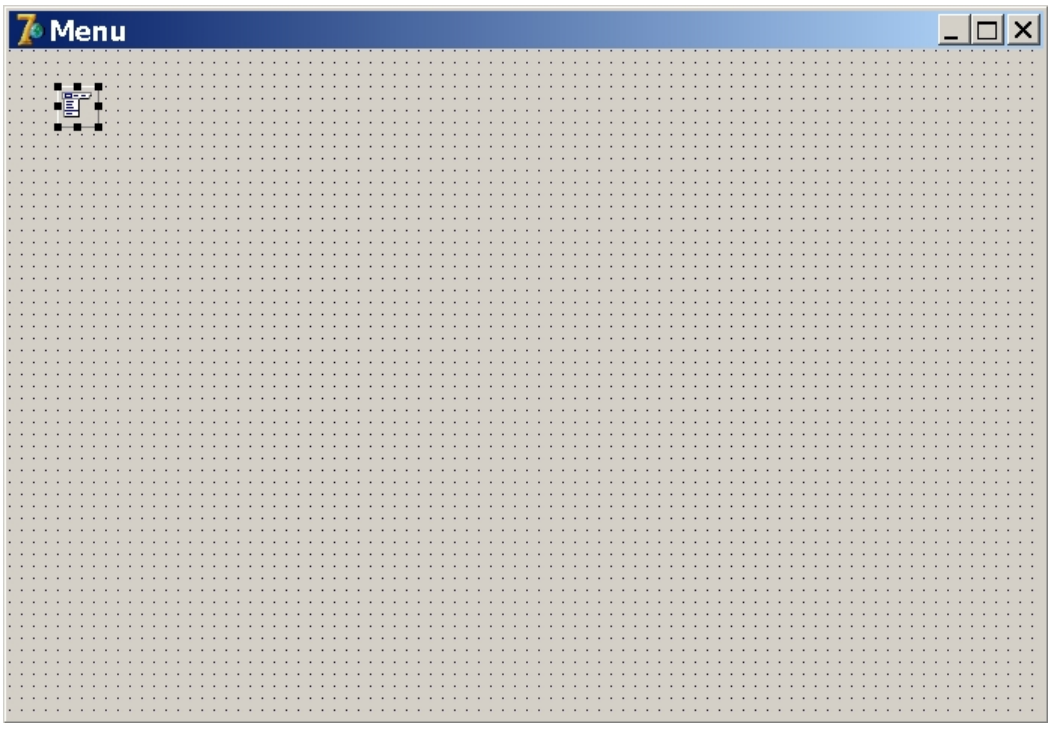
D. *Exemple d'application : le biorythme*

1. Le programme principal : biorythme.dpr

```
program biorythme;

uses
  Forms,
  main in 'main.pas' {FormSaisie},
  trousse in 'trousse.pas',
  tableau in 'tableau.pas' {FormTableau},
  graphe in 'graphe.pas' {FormGraphe},
  menu in 'menu.pas' {FormMenu};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TFormMenu, FormMenu);
//  Application.CreateForm(TFormSaisie, FormSaisie);
// Application.CreateForm(TFormTableau, FormTableau);
//  Application.CreateForm(TFormGraphe, FormGraphe);
  Application.Run;
end.
```

2. Le menu : menu.pas

a) La Fiche



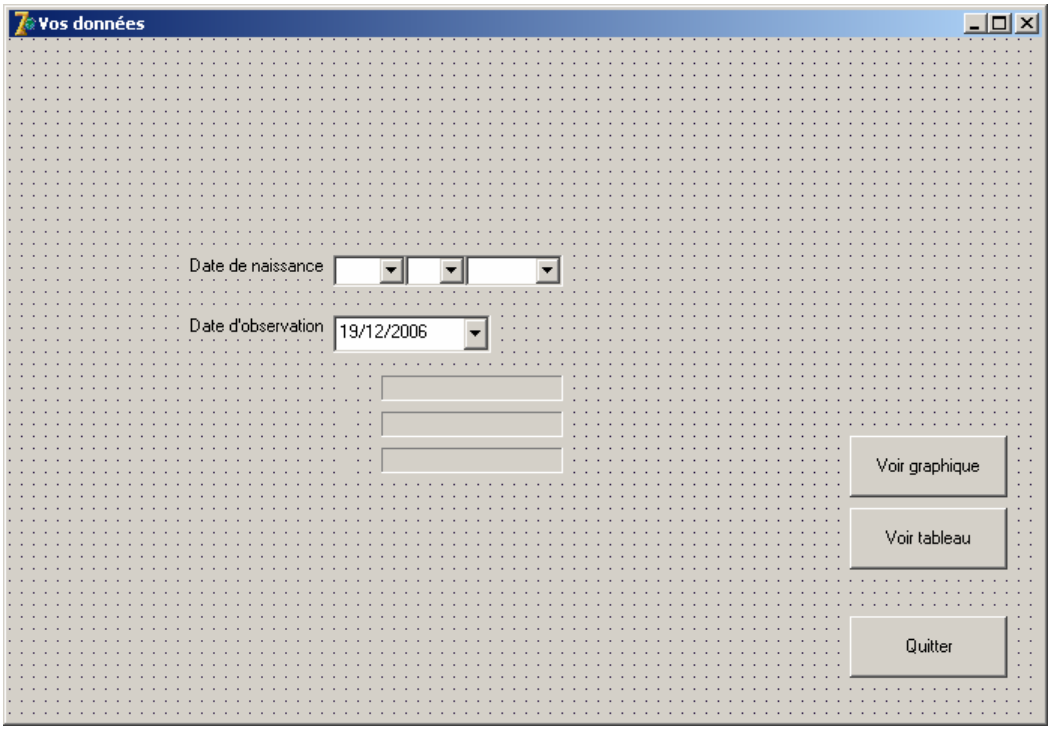
b) Les éléments visuels

```
object FormMenu: TFormMenu
  Left = 198
  Top = 132
  Width = 696
  Height = 480
  Caption = 'Menu'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  FormStyle = fsMDIForm
  Menu = MainMenuBiorythme
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
  object MainMenuBiorythme: TMainMenu
    Left = 32
    Top = 24
    object Fichier: TMenuItem
      Caption = '&Fichier'
      object Donnees: TMenuItem
        Caption = '&Donn'#233'es'
        OnClick = DonneesClick
      end
      object Tableau: TMenuItem
        Caption = '&Tableau'
        Enabled = False
        OnClick = TableauClick
```

```
      end
      object Graphique: TMenuItem
        Caption = '&Graphique'
        Enabled = False
        OnClick = GraphiqueClick
      end
      object Fermer: TMenuItem
        Caption = '&Fermer'
        OnClick = FermerClick
      end
      object Quitter: TMenuItem
        Caption = '&Quitter'
        OnClick = QuitterClick
      end
    end
      object Fenetre: TMenuItem
        Caption = 'F&en'#234'tre'
        object Vertical: TMenuItem
          Caption = 'R'#233'organiser
&Vertical'
          OnClick = VerticalClick
        end
        object Horizontal: TMenuItem
          Caption = 'R'#233'organiser
&Horizontal'
          OnClick = HorizontalClick
        end
      end
    end
end
```

3. La fiche de saisie : main.pas

a) La fiche



b) Les propriétés de la fiche : main.dfm

```
object FormSaisie: TFormSaisie
  Left = 251
  Top = 145
  Width = 696
  Height = 480
  Caption = 'Vos donn'#233'es'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  FormStyle = fsMDIChild
  OldCreateOrder = False
  Position = poDefault
  Visible = True
  OnClose = FormClose
  OnCreate = FormCreate
  PixelsPerInch = 96
  TextHeight = 13
  object LabelNaissance: TLabel
    Left = 120
    Top = 144
    Width = 89
    Height = 13
    Caption = 'Date de naissance'
  end
  object LabelObservation: TLabel
    Left = 120
    Top = 184
    Width = 89
    Height = 13
    Caption = 'Date d'#39'observation'
  end
  object LabelResPhy: TLabel
    Left = 224
    Top = 224
    Width = 3
    Height = 13
  end
  object LabelResEmo: TLabel
    Left = 224
    Top = 248
    Width = 3
    Height = 13
  end
  object LabelResInt: TLabel
    Left = 224
    Top = 280
    Width = 3
```

```
    Height = 13
  end
  object ComboBoxJour: TComboBox
    Left = 216
    Top = 144
    Width = 49
    Height = 21
    ItemHeight = 13
    TabOrder = 0
  end
  object ComboBoxMois: TComboBox
    Left = 264
    Top = 144
    Width = 41
    Height = 21
    ItemHeight = 13
    TabOrder = 1
  end
  object ComboBoxAnnee: TComboBox
    Left = 304
    Top = 144
    Width = 65
    Height = 21
    ItemHeight = 13
    TabOrder = 2
  end
  object DateTimePickerJour:
TDateTimePicker
    Left = 216
    Top = 184
    Width = 105
    Height = 25
    Date = 39070.413248333330000000
    Time = 39070.413248333330000000
    TabOrder = 3
    OnChange = DateTimePickerJourChange
  end
  object ButtonQuitter: TButton
    Left = 560
    Top = 384
    Width = 105
    Height = 41
    Caption = 'Quitter'
    TabOrder = 4
    OnClick = ButtonQuitterClick
  end
  object ButtonTableau: TButton
    Left = 560
    Top = 312
    Width = 105
    Height = 41
    Caption = 'Voir tableau'
    TabOrder = 5
    OnClick = ButtonTableauClick
  end
  object ButtonGraphique: TButton
    Left = 560
    Top = 264
    Width = 105
    Height = 41
    Caption = 'Voir graphique'
    TabOrder = 6
    OnClick = ButtonGraphiqueClick
  end
  object ProgressBarPhy: TProgressBar
    Left = 248
    Top = 224
    Width = 121
    Height = 17
    TabOrder = 7
  end
  object ProgressBarEmo: TProgressBar
    Left = 248
    Top = 248
    Width = 121
    Height = 17
    TabOrder = 8
  end
  object ProgressBarInt: TProgressBar
    Left = 248
    Top = 272
    Width = 121
    Height = 17
    TabOrder = 9
  end
end
```

c) Le code

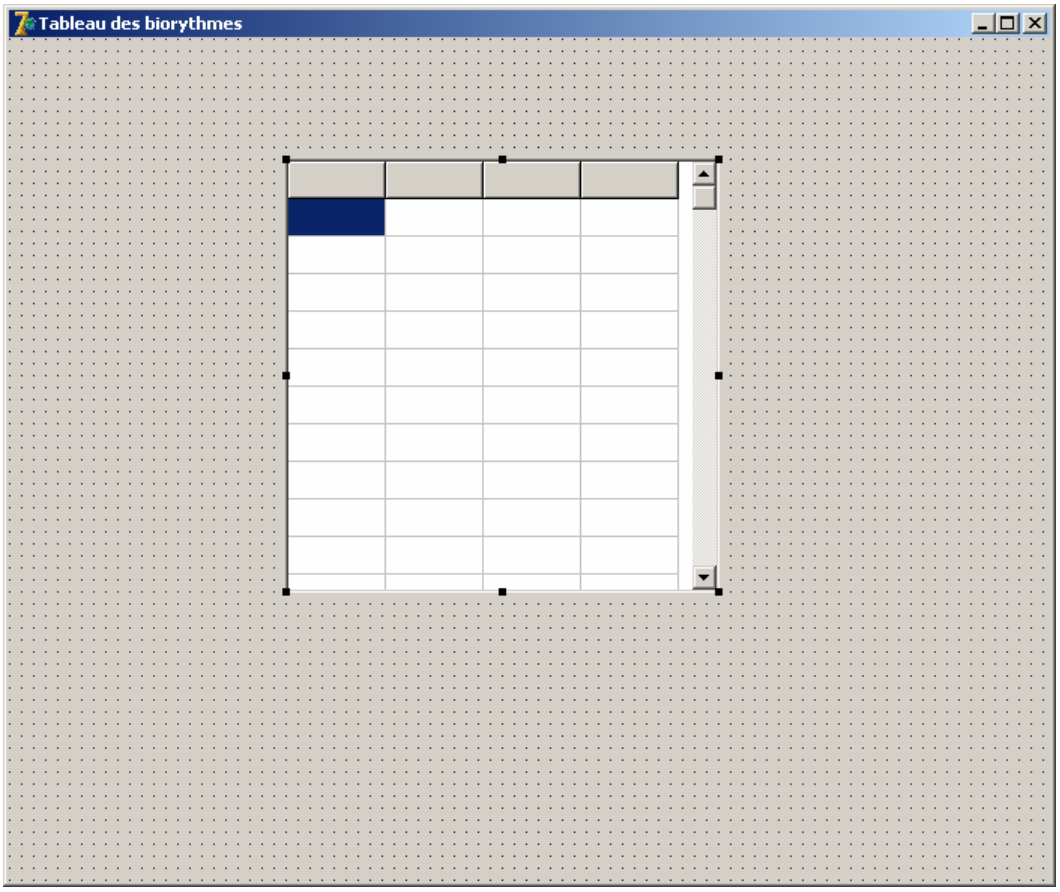
```
unit main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls,DateUtils, ComCtrls,Trousse,Tableau,Graphe;

type
  TFormSaisie = class(TForm)
    ComboBoxJour: TComboBox;
    ComboBoxMois: TComboBox;
    ComboBoxAnnee: TComboBox;
    DateTimePickerJour: TDateTimePicker;
    LabelNaissance: TLabel;
    LabelObservation: TLabel;
    LabelResPhy: TLabel;
    LabelResEmo: TLabel;
    LabelResInt: TLabel;
    ButtonQuitter: TButton;
    ButtonTableau: TButton;
    ButtonGraphique: TButton;
    ProgressBarPhy: TProgressBar;
    ProgressBarEmo: TProgressBar;
    ProgressBarInt: TProgressBar;
    procedure FormCreate(Sender: TObject);
    procedure DateTimePickerJourChange(Sender: TObject);
    procedure ButtonQuitterClick(Sender: TObject);
    procedure ButtonTableauClick(Sender: TObject);
```

```
    procedure ButtonGraphiqueClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
  public
  end;


var
  FormSaisie: TFormSaisie;

implementation

uses menu;



{$R *.dfm}



procedure TFormSaisie.FormCreate(Sender: TObject);
var
  Annee,Mois,Jour,I:integer;
  Naissance:TDateTime;
  strMois,strJour:string;
begin
  Naissance:=Date;
  Annee:=YearOf(Naissance);
  Mois:=MonthOf(Naissance);
  Jour:=DayOf(Naissance);
  for I:= Annee+5 downto Annee - 85 do
  begin
    Self.ComboBoxAnnee.AddItem(IntToStr(I),Self);
  end;
  (*
  for I:=1 to 12 do
  begin
    strMois:=IntToStr(I);
    if I<10 then
    begin
      strMois:='0'+strMois;
    end;
    Self.ComboBoxMois.AddItem(strMois,Self);
  end;
  *)
  for I:=1 to 12 do
  begin
    Self.ComboBoxMois.AddItem(ZeroNonSignificatif(I),Self);
  end;
  (*
  for I:=1 to 31 do
  begin
    strJour:=IntToStr(I);
    if I<10 then
    begin
      strJour:='0'+strJour;
    end;
    Self.ComboBoxJour.AddItem(strJour,Self);
  end;
  *)
  for I:=1 to 31 do
  begin
    Self.ComboBoxJour.AddItem(ZeroNonSignificatif(I),Self);
  end;

  Self.ComboBoxAnnee.Text:= IntToStr(1964);
  Self.ComboBoxMois.Text:= IntToStr(1);
  Self.ComboBoxJour.Text:= IntToStr(21);
end;

procedure TFormSaisie.DateTimePickerJourChange(Sender: TObject);
var
  DateObservation,DateNaissance:TDateTime;
  Annee,Mois,Jour:integer;
```

```
  BioPhy,BioEmo,BioInt:Integer;
  I:integer;
begin
  DateObservation:=Self.DateTimePickerJour.Date;
  Annee:=StrToInt(Self.ComboBoxAnnee.Text);
  Mois:=StrToInt(Self.ComboBoxMois.Text);
  Jour:=StrToInt(Self.ComboBoxJour.Text);
  DateNaissance:=EncodeDate(Annee,Mois,Jour);
  BioPhy:=Trunc(BioRythme(DateObservation,DateNaissance,23));
  BioEmo:=Trunc(BioRythme(DateObservation,DateNaissance,28));
  BioInt:=Trunc(BioRythme(DateObservation,DateNaissance,33));
  Self.LabelResPhy.Caption:=IntToStr(BioPhy);
  Self.LabelResEmo.Caption:=IntToStr(BioEmo);
  Self.LabelResInt.Caption:=IntToStr(BioInt);
  Self.ProgressBarPhy.Position:=0;
  //Self.ProgressBarPhy.Step:=1;
  Repeat
    Self.ProgressBarPhy.StepBy(1);
    //Self.ProgressBarPhy.StepIt;
    Self.ProgressBarPhy.Refresh;
  until Self.ProgressBarPhy.Position=BioPhy;
  Self.ProgressBarEmo.Position:=BioEmo;
  Self.ProgressBarInt.Position:=BioInt;
  Self.Refresh;

end;

procedure TFormSaisie.ButtonQuitterClick(Sender: TObject);
begin
  Self.Close;
end;

procedure TFormSaisie.ButtonTableauClick(Sender: TObject);
begin
  //Self.Hide;
  FormTableau.Show;
end;

procedure TFormSaisie.ButtonGraphiqueClick(Sender: TObject);
begin
  FormGraphe.Show;
end;

procedure TFormSaisie.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  FormMenu.Tableau.Enabled:=false;
  formMenu.Graphique.Enabled:=false;
  Self.Destroy;
end;

end.
```

4. La fiche tableau.pas

a) La fiche



b) Les propriétés

```
object FormTableau: TFormTableau
  Left = -4
  Top = -4
  Width = 1032
  Height = 741
  Caption = 'Tableau des biorythmes'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  FormStyle = fsMDIChild
  OldCreateOrder = False
  Position = poDefault
  Visible = True
  OnClose = FormClose
  OnShow = FormShow
  PixelsPerInch = 96
  TextHeight = 13
  object StringGridBio: TStringGrid
    Left = 184
    Top = 80
    Width = 289
    Height = 289
    ColCount = 4
    FixedCols = 0
    RowCount = 31
    TabOrder = 0
  end
end
```

c) Le code

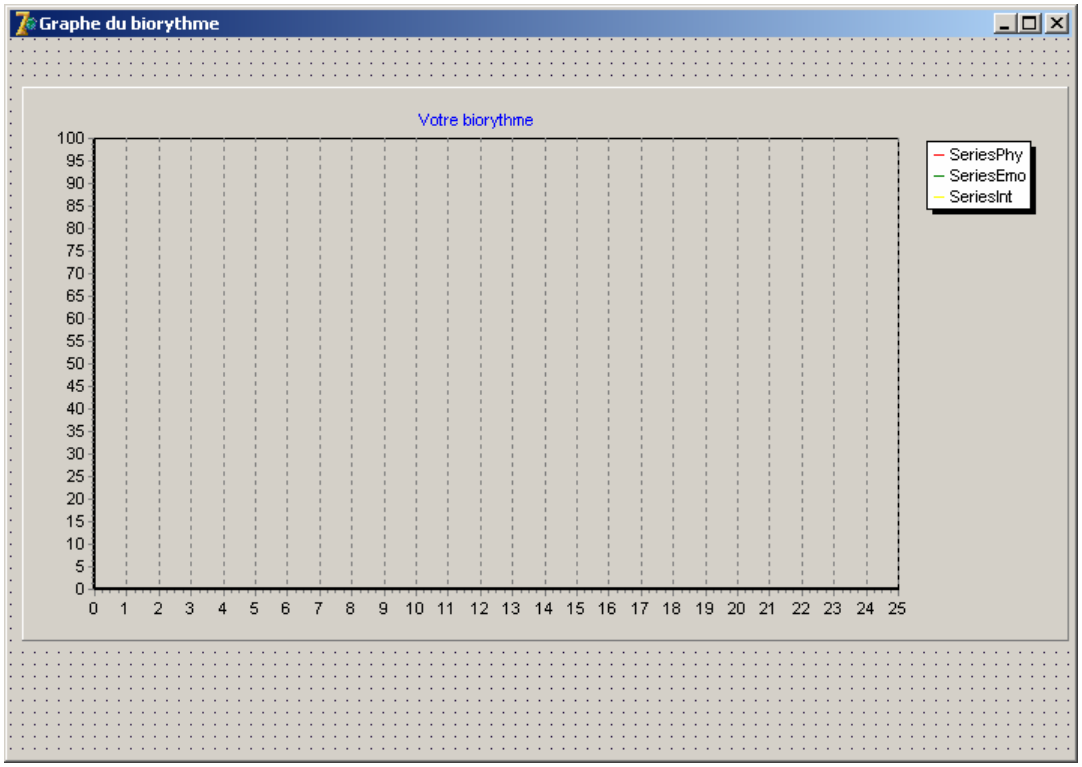
```
unit tableau;
```

```
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids,DateUtils,Trousse;

type
  TFormTableau = class(TForm)
    StringGridBio: TStringGrid;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormShow(Sender: TObject);
    procedure FormCreate(Sender: TObject);

  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;

var
  FormTableau: TFormTableau;

implementation

uses main;

{$R *.dfm}

procedure TFormTableau.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  //FormSaisie.Show;
end;

procedure TFormTableau.FormShow(Sender: TObject);
var
  I,J,Annee,Mois,Jour:integer;
  DateObservation,DateNaissance:TDateTime;
  Indice:Integer;

begin
  DateObservation:=FormSaisie.DateTimePickerJour.Date;
  Annee:=StrToInt(FormSaisie.ComboBoxAnnee.Text);
  Mois:=StrToInt(FormSaisie.ComboBoxMois.Text);
  Jour:=StrToInt(FormSaisie.ComboBoxJour.Text);
  DateNaissance:=EncodeDate(Annee,Mois,Jour);
  Indice:=90;
  Self.StringGridBio.RowCount:=Indice+1;

  for I:=1 to Indice do
  begin
    DateObservation:=IncDay(DateObservation,1);
    Self.StringGridBio.Cells[0,I]:=DateToStr(DateObservation);

Self.StringGridBio.Cells[1,I]:=IntToStr(Trunc(BioRythme(DateObservation,DateNaissance,23)
));

Self.StringGridBio.Cells[2,I]:=IntToStr(Trunc(BioRythme(DateObservation,DateNaissance,28)
));

Self.StringGridBio.Cells[3,I]:=IntToStr(Trunc(BioRythme(DateObservation,DateNaissance,33)
));
  end;
  with Self.StringGridBio do
  begin
    Cells[0,0]:='Jour';
    Cells[1,0]:='Physique';
    Cells[2,0]:='Emotionnel';
    Cells[3,0]:='Intellectuel';
  end;
  for I:=1 to 3 do
  begin
    for J:=1 to Indice do
```

```
    begin
      if StrToInt(Self.StringGridBio.Cells[I,J])<25 Then
      begin
      end
      else if StrToInt(Self.StringGridBio.Cells[I,J])>75 Then
      begin
      end;
    end;
  end;

end;

procedure TFormTableau.FormCreate(Sender: TObject);
begin

end;

end.
```

5. La fiche graphe

a) La fiche



b) Les propriétés

```
object FormGraphe: TFormGraphe
  Left = 214
  Top = 147
  Width = 715
  Height = 505
  Caption = 'Graphe du biorythme'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  FormStyle = fsMDIChild
  OldCreateOrder = False
  Position = poDefault
  Visible = True
  OnShow = FormShow
  PixelsPerInch = 96
  TextHeight = 13
  object ChartBio: TChart
    Left = 8
    Top = 32
    Width = 697
    Height = 369
    BackWall.Brush.Color = clWhite
    BackWall.Brush.Style = bsClear
    Title.Text.Strings = (
      'Votre biorythme')
    LeftAxis.Automatic = False
    LeftAxis.AutomaticMaximum = False
    LeftAxis.AutomaticMinimum = False
    LeftAxis.AxisValuesFormat = '0'
    LeftAxis.ExactDateTime = False
    LeftAxis.Grid.Visible = False
    LeftAxis.Increment =
1.000000000000000000
    LeftAxis.Maximum =
100.000000000000000000
    View3D = False
    TabOrder = 0
```

```
    object Series1: TLineSeries
      Marks.ArrowLength = 8
      Marks.Visible = False
      SeriesColor = clRed
      Title = 'SeriesPhy'
      ValueFormat = 'jj/mm'
      Pointer.InflateMargins = True
      Pointer.Style = psRectangle
      Pointer.Visible = False
      XValues.DateTime = False
      XValues.Name = 'X '
      XValues.Multiplier =
1.000000000000000000
      XValues.Order = loAscending
      YValues.DateTime = False
      YValues.Name = 'Y '
      YValues.Multiplier =
1.000000000000000000
      YValues.Order = loNone
    end
    object Series2: TLineSeries
      Marks.ArrowLength = 8
      Marks.Visible = False
      SeriesColor = clGreen
      Title = 'SeriesEmo'
      Pointer.InflateMargins = True
      Pointer.Style = psRectangle
      Pointer.Visible = False
      XValues.DateTime = False
      XValues.Name = 'X '
      XValues.Multiplier =
1.000000000000000000
      XValues.Order = loAscending
      YValues.DateTime = False
      YValues.Name = 'Y '
      YValues.Multiplier =
1.000000000000000000
      YValues.Order = loNone
    end
    object Series3: TLineSeries
      Marks.ArrowLength = 8
      Marks.Visible = False
      SeriesColor = clYellow
      Title = 'SeriesInt'
      Pointer.InflateMargins = True
      Pointer.Style = psRectangle
      Pointer.Visible = False
      XValues.DateTime = False
      XValues.Name = 'X '
      XValues.Multiplier =
1.000000000000000000
      XValues.Order = loAscending
      YValues.DateTime = False
      YValues.Name = 'Y '
      YValues.Multiplier =
1.000000000000000000
      YValues.Order = loNone
    end
  end
end
```

c) Le code

```
unit graphe;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, TeEngine, Series, ExtCtrls, TeeProcs, Chart,DateUtils,Trousse;

type
  TFormGraphe = class(TForm)
    ChartBio: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    Series3: TLineSeries;
    procedure FormShow(Sender: TObject);
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;

var
  FormGraphe: TFormGraphe;

implementation

uses main;

{$R *.dfm}

procedure TFormGraphe.FormShow(Sender: TObject);
var
  I:integer;
  DateObservation,DateNaissance:TDateTime;
  Annee,Jour,Mois:Integer;
  BioPhy,BioEmo,BioInt:Integer;
begin
  DateObservation:=FormSaisie.DateTimePickerJour.Date;
  Annee:=StrToInt(FormSaisie.ComboBoxAnnee.Text);
  Mois:=StrToInt(FormSaisie.ComboBoxMois.Text);
  Jour:=StrToInt(FormSaisie.ComboBoxJour.Text);
  DateNaissance:=EncodeDate(Annee,Mois,Jour);
```

```
  Self.Series1.Clear;
  Self.Series2.Clear;
  Self.Series3.Clear;
  for I:=1 to 90 do
  begin
    DateObservation:=IncDay(DateObservation,1);
    BioPhy:=Trunc(BioRythme(DateObservation,DateNaissance,23));
    BioEmo:=Trunc(BioRythme(DateObservation,DateNaissance,28));
    BioInt:=Trunc(BioRythme(DateObservation,DateNaissance,33));
    Self.Series1.Add(BioPhy,DateToStr(DateObservation),clGreen);
    Self.Series2.Add(BioEmo,DateToStr(DateObservation),clRed);
    Self.Series3.Add(BioInt,DateToStr(DateObservation),clBlue);
  end;
end;

end.
```

V. Interbase

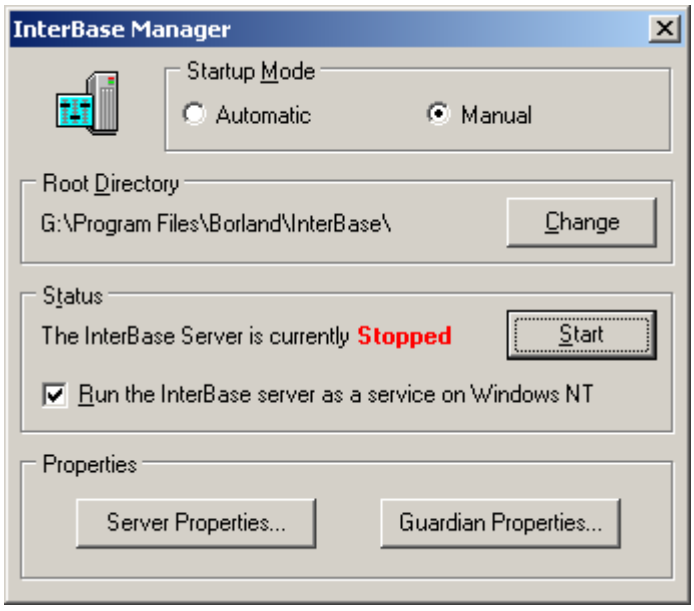
A. Présentation

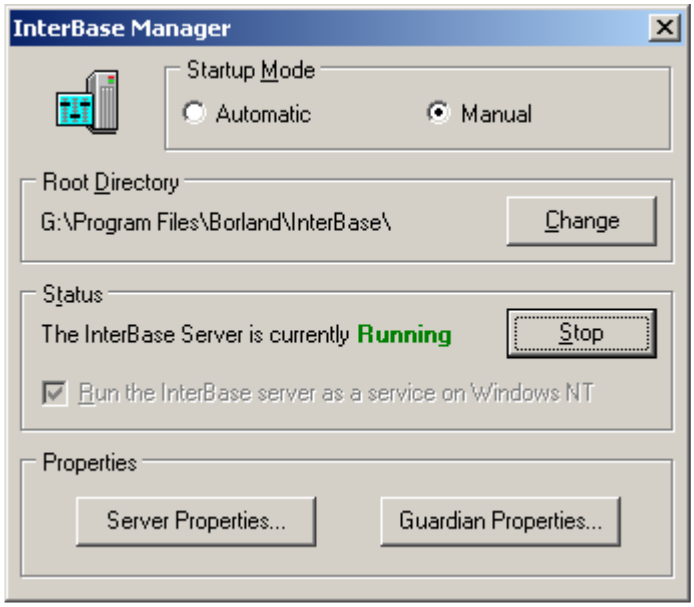
Interbase est un moteur de base de données relationnel de très grande qualité. Il fonctionne en mode client-serveur et se comporte comme un service réseau. Il écoute sur le port Tcp/3050. Le fichier de configuration est, sous Windows, C:\Program Files\Borland\InterBase\ibconfig.

```
#V4_LOCK_MEM_SIZE        98304
#V4_LOCK_SEM_COUNT               32
#V4_LOCK_SIGNAL          16
#V4_EVENT_MEM_SIZE               32768
#DATABASE_CACHE_PAGES            75
#SERVER_PRIORITY_CLASS           1
#SERVER_CLIENT_MAPPING           4096
#SERVER_WORKING_SIZE_MIN                 0
#SERVER_WORKING_SIZE_MAX                 0
#V4_LOCK_GRANT_ORDER             1
#ANY_LOCK_MEM_SIZE               98304
#ANY_LOCK_SEM_COUNT              32
#ANY_LOCK_SIGNAL         16
#ANY_EVENT_MEM_SIZE              32768
#CPU_AFFINITY            1
```

B. Le gestionnaire Interbase Server Manager

Lors de l'installation de Delphi, l'assistant vous propose d'installer Interbase Client. En réalité, il procède à l'installation du moteur de base de Données relationnel Interbase Server. Cliquez sur le Bouton Start pour lancer Interbase.



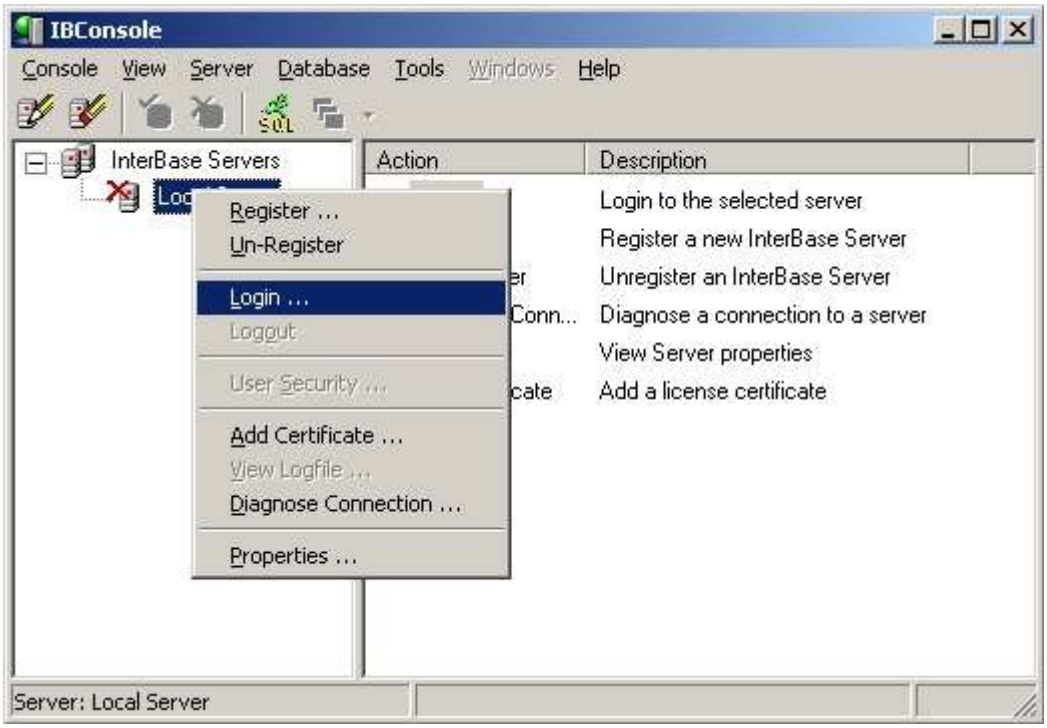


La base est lancée lorsque le mot Running est affiché.

Lorsque vous disposez d'un pare-feu et que Vous souhaitez Accéder à partir D'une machine distante, pensez à ouvrir le port 3050.

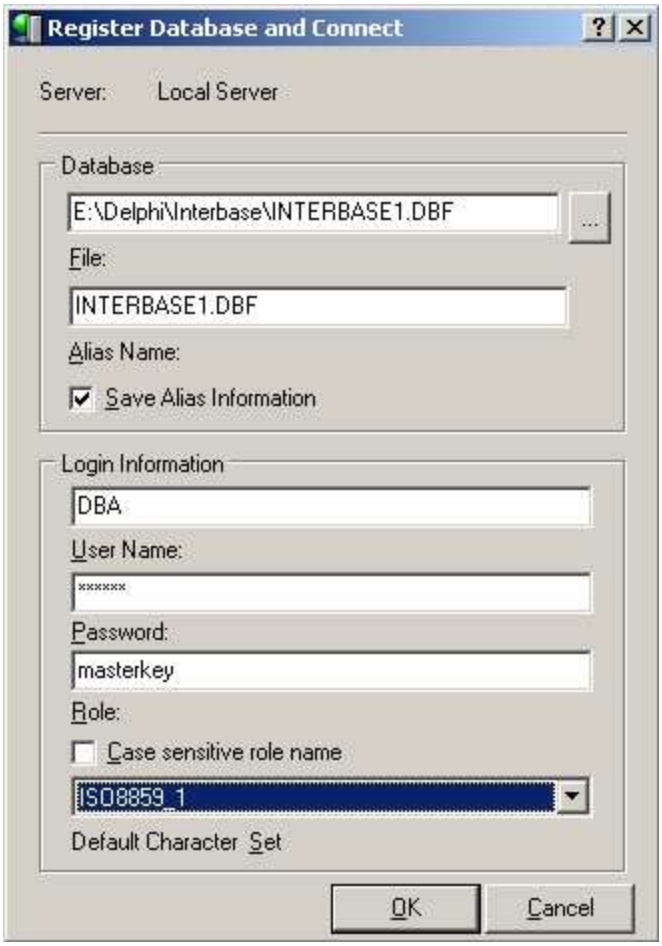
C. Outils d'administration

Borland fournit un outil d'administration IBConsole. Lancez l'outil à partir du menu Démarrer et, par un clic droit, choisissez Login.



Entrez le compte Dba SYSDBA et le mot de passe par défaut masterkey.

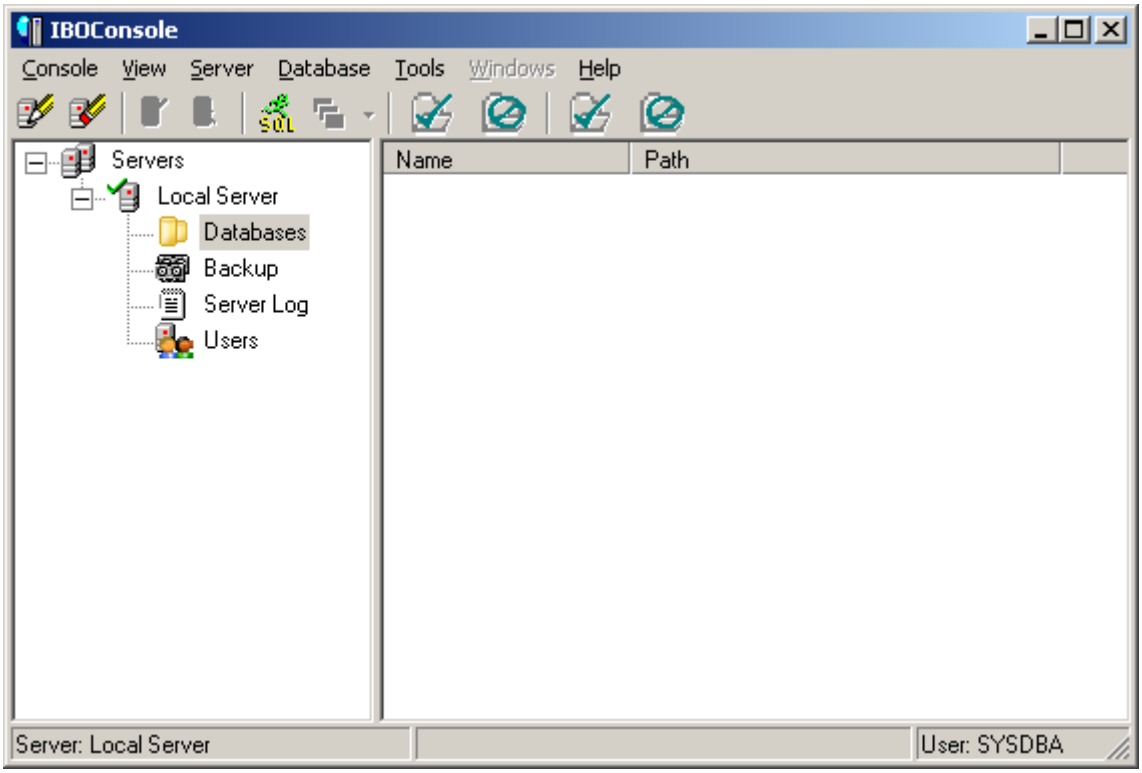
Pour vous connecter à la base de données, vous devez la créer au préalable par un clic droit. Une fois créée, enregistrez la dans votre environnement d'administration fourni par IBConsole.



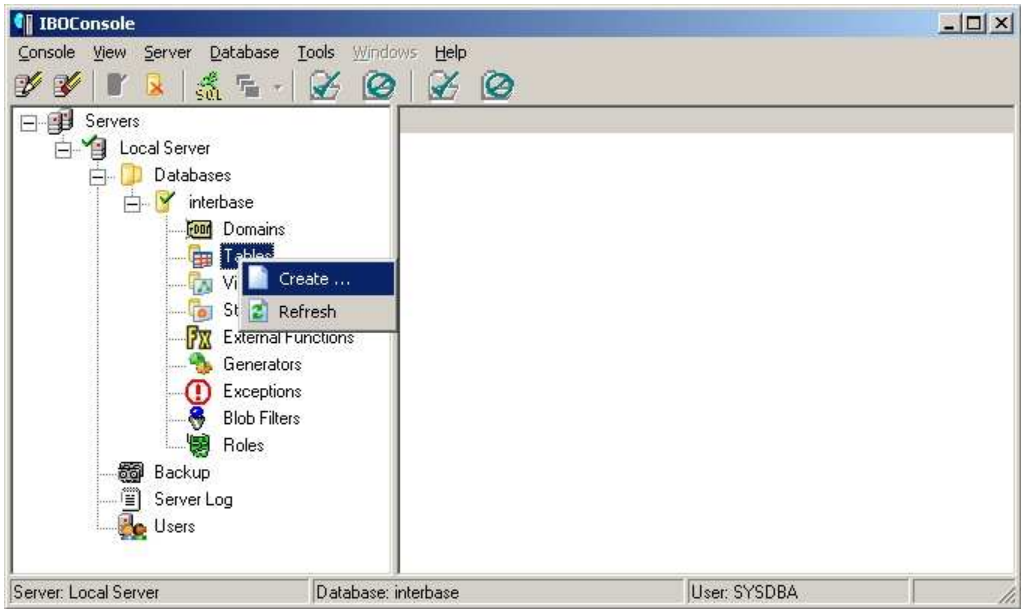
Entrez le fichier qui représente la base de données (extension gdb par défaut). Entrez l'alias, les informations de login et choisissez le jeu de caractères correspondant à celui choisi lors de la création.

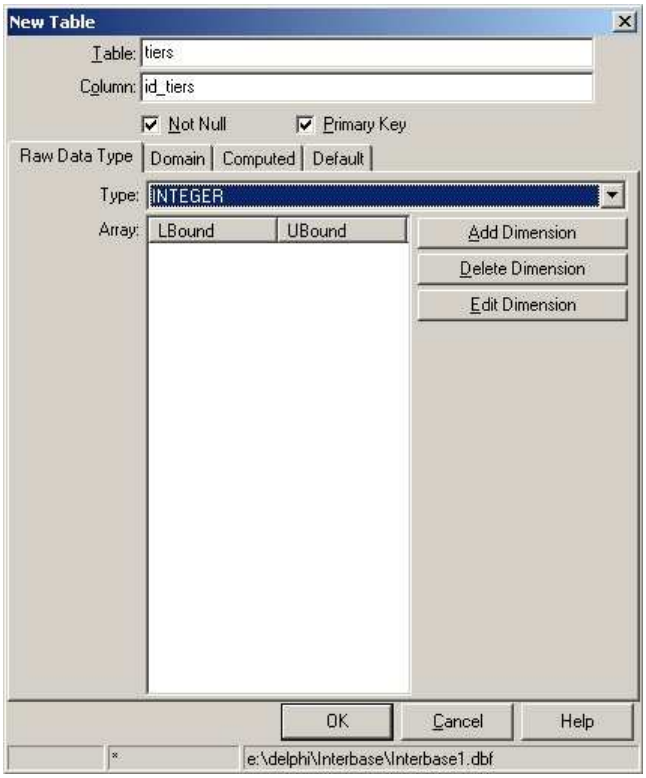
L'outil fourni par Borland ne permet pas de créer les tables et d'autres objets.

Il existe un clone de cette console : IBOConsole sous licence MPL qui vous permet de créer les objets de la base.



D. Création de tables





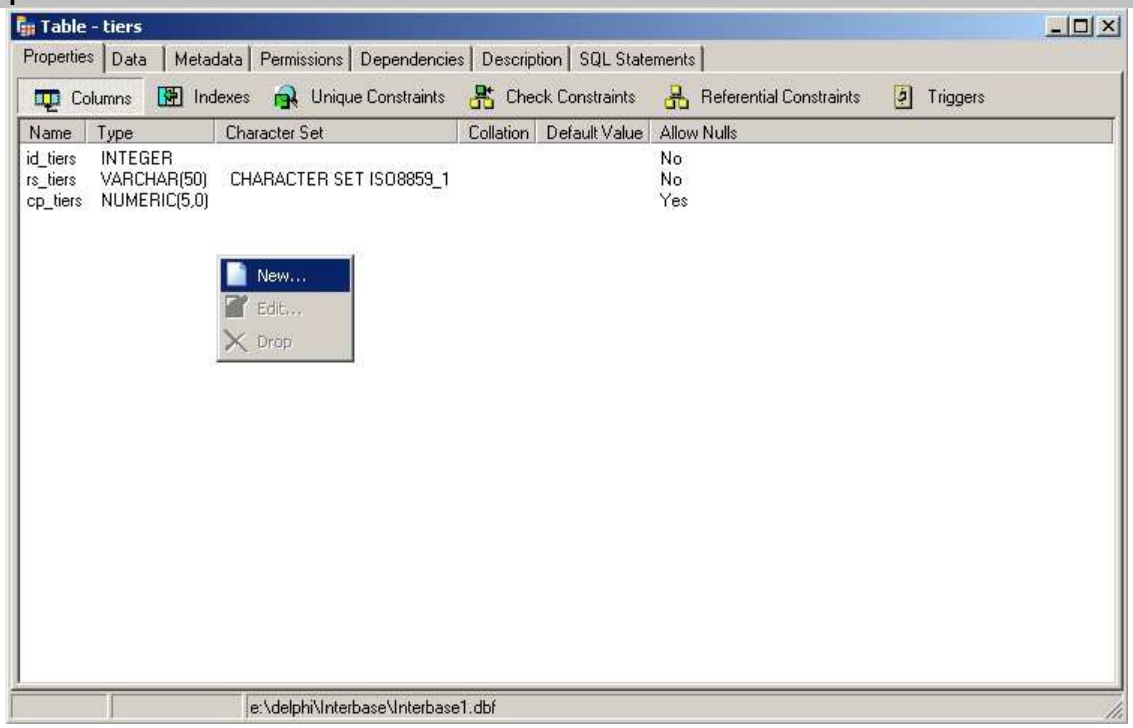
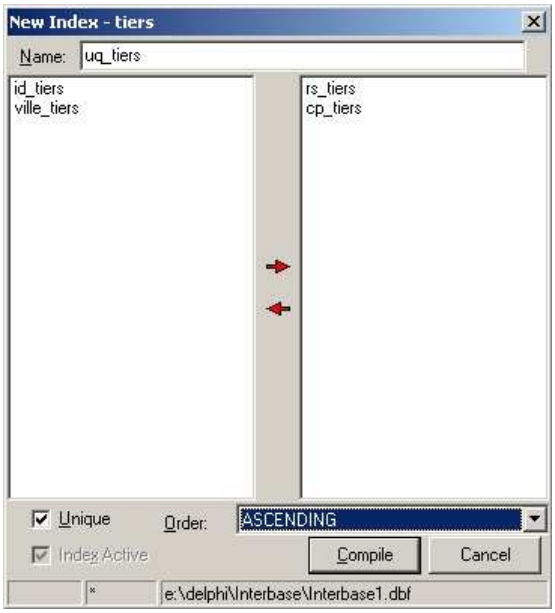


Table - tiers
Properties
Data
Metadata
Permissions
Dependencies
Description
SQL Statements
Columns
Indexes
Unique Constraints
Check Constraints
Referential Constraints
Triggers
Name
Type
Character Set
Collation
Default Value
Allow Nulls
id_tiers INTEGER No
rs_tiers VARCHAR(50) CHARACTER SET ISO8859_1 No
cp_tiers NUMERIC(5,0) Yes
New...
Edit...
Drop
e:\delphi\Interbase\Interbase1.dbf


New Index - tiers
Name:
uq_tiers
id_tiers
ville_tiers
rs_tiers
cp_tiers
Unique
Order:
ASCENDING
Index Active
Compile
Cancel
e:\delphi\Interbase\Interbase1.dbf

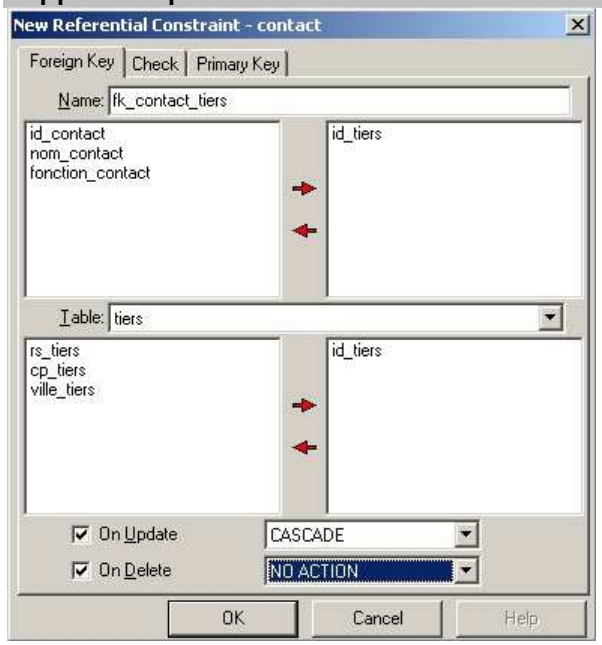

New Referential Constraint - contact
Foreign Key
Check
Primary Key
Name: fk_contact_tiers
id_contact
nom_contact
fonction_contact
id_tiers
Table: tiers
rs_tiers
cp_tiers
ville_tiers
id_tiers
On Update
CASCADE
On Delete
NO ACTION
OK
Cancel
Help


Table - tiers
Properties
Data
Metadata
Permissions
Dependencies
Description
SQL Statements
Close
Form View
Filter
Empty Table
Copy Record
id_tiers
rs_tiers
cp_tiers
ville_tiers
1
Dsfc
27800
Saint Eloi de Fourques
2
Winuxware
76300
Sotteville les Rouen
e:\delphi\Interbase\Interbase1.dbf

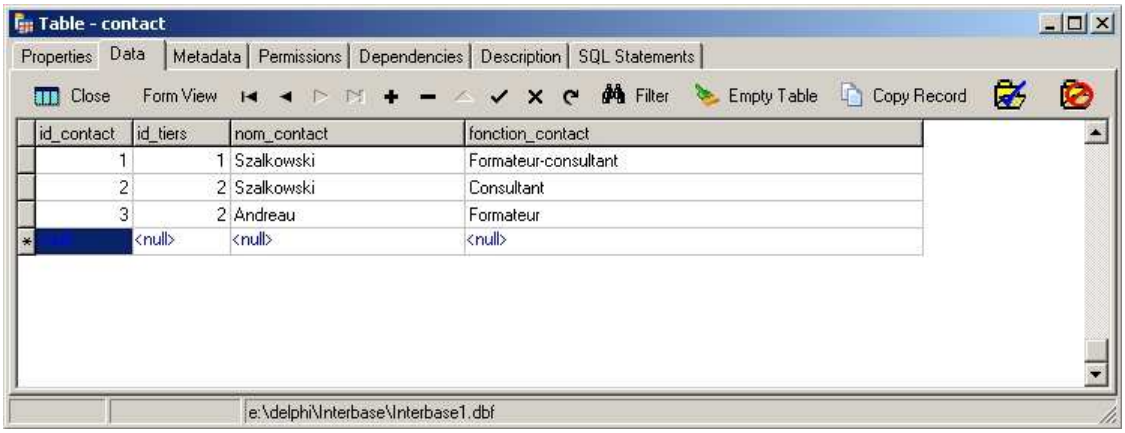
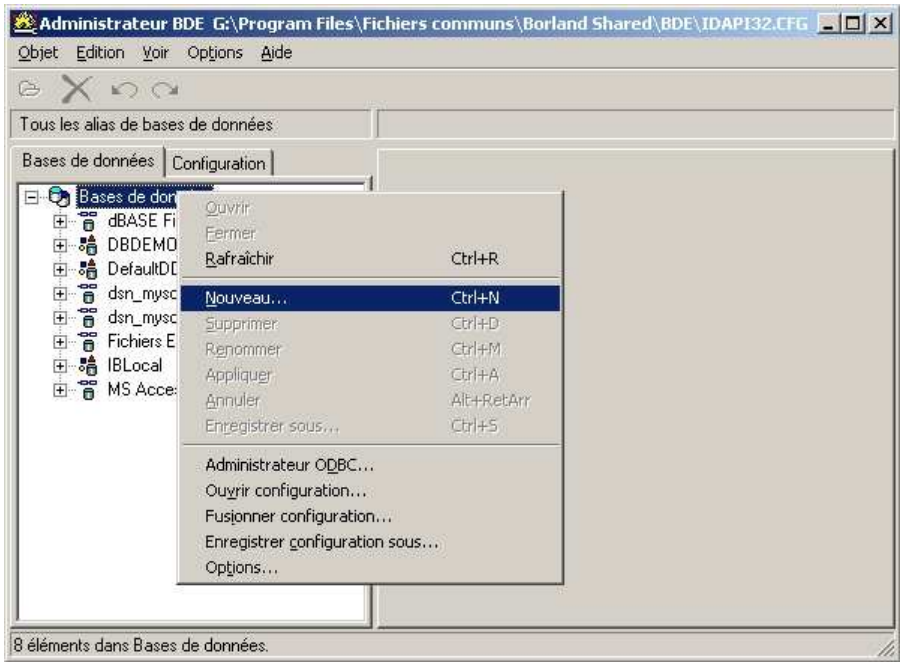
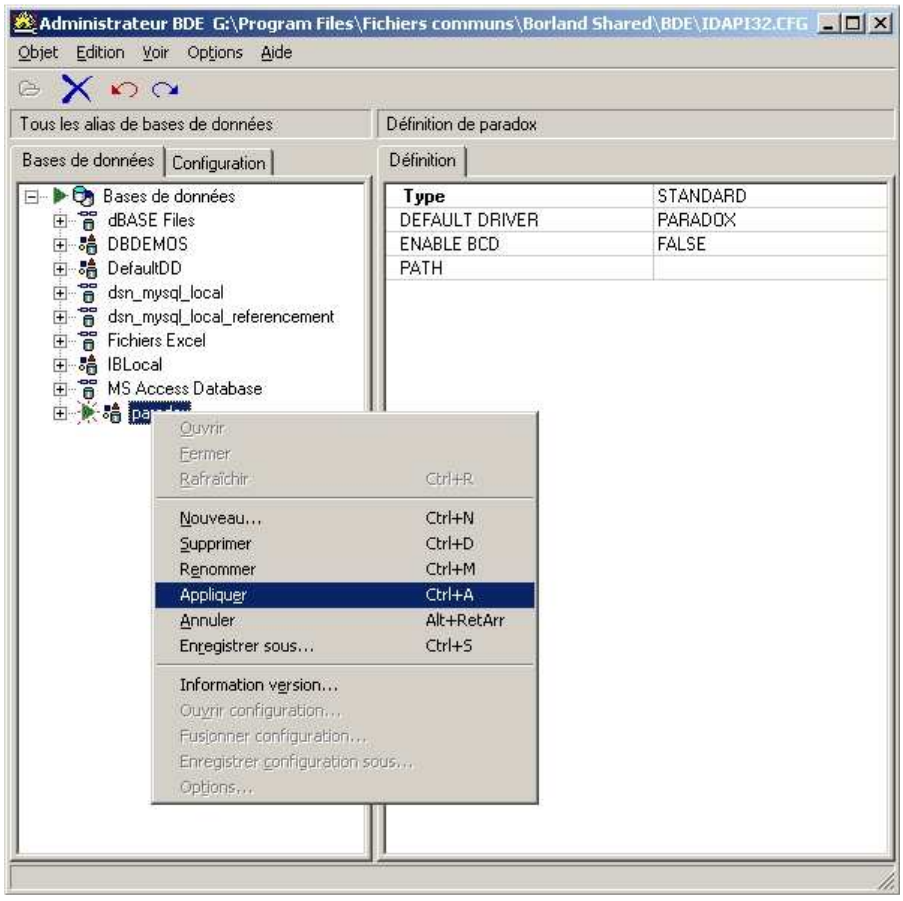


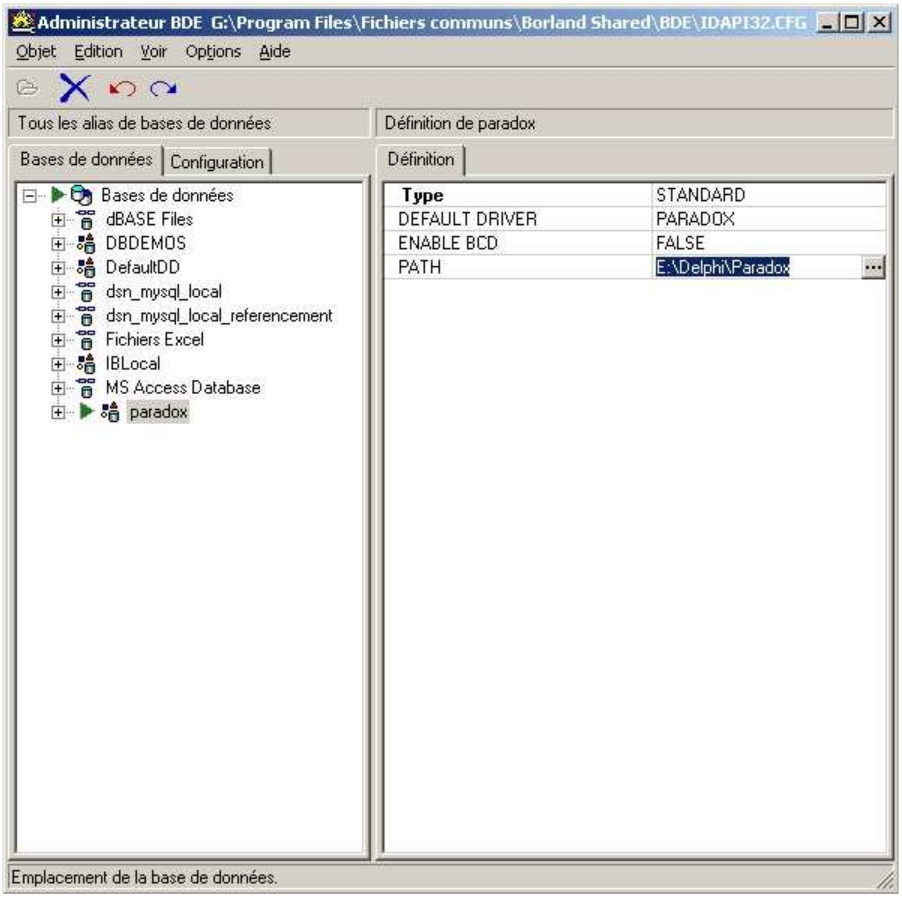
Table - contact
Properties
Data
Metadata
Permissions
Dependencies
Description
SQL Statements
Close
Form View
Filter
Empty Table
Copy Record
id_contact
id_tiers
nom_contact
fonction_contact
1
1
Szalkowski
Formateur-consultant
2
2
Szalkowski
Consultant
3
2
Andreau
Formateur
<null>
<null>
<null>
e:\delphi\Interbase\Interbase1.dbf

VI. Paradox

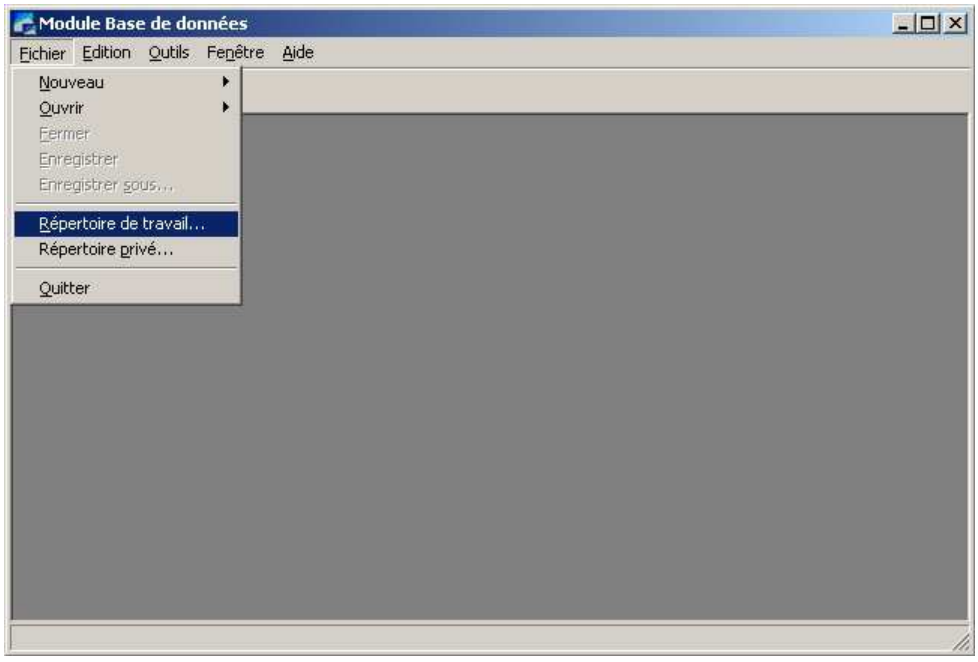
A. Création du connecteur BDE

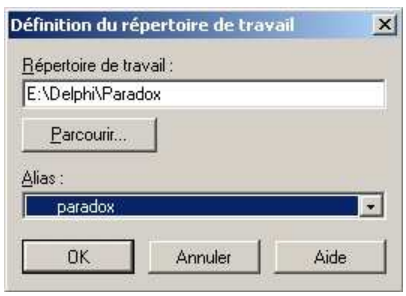


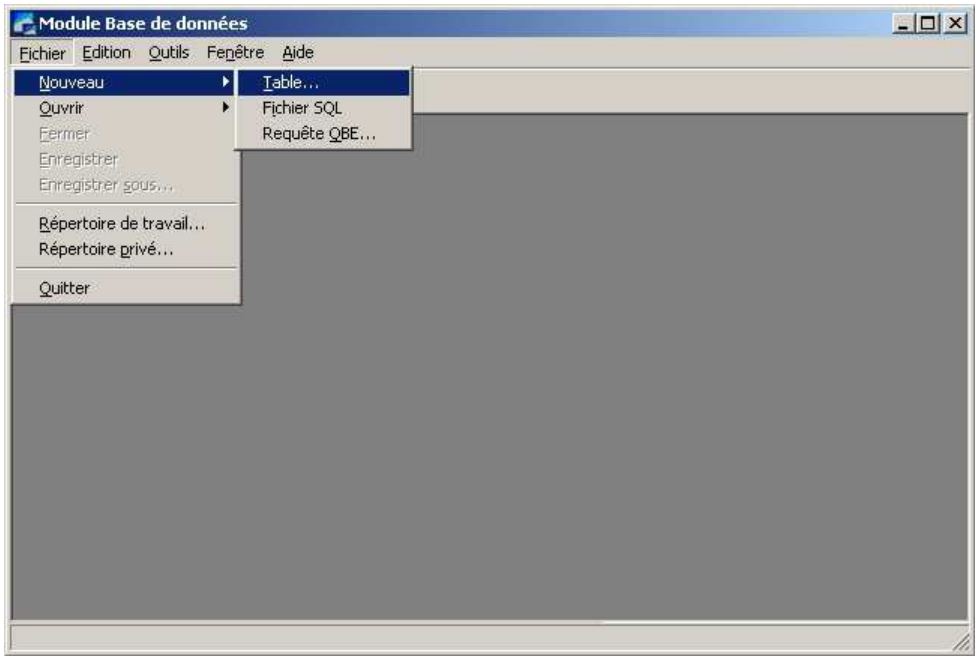


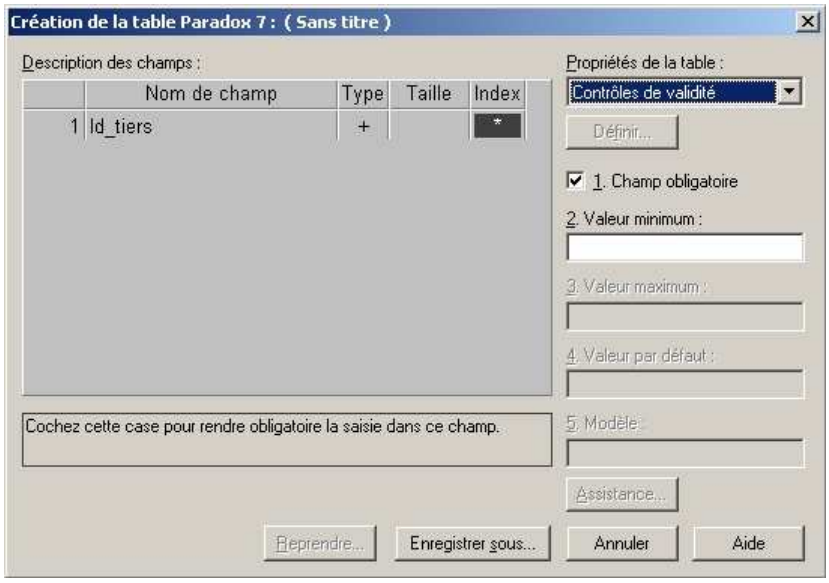


B. Création des tables






Module Base de données
Fichier Edition Outils Fenêtre Aide
Nouveau
Ouvrir
Fermer
Enregistrer
Enregistrer sous...
Répertoire de travail...
Répertoire privé...
Quitter
Table...
Fichier SQL
Requête QBE...

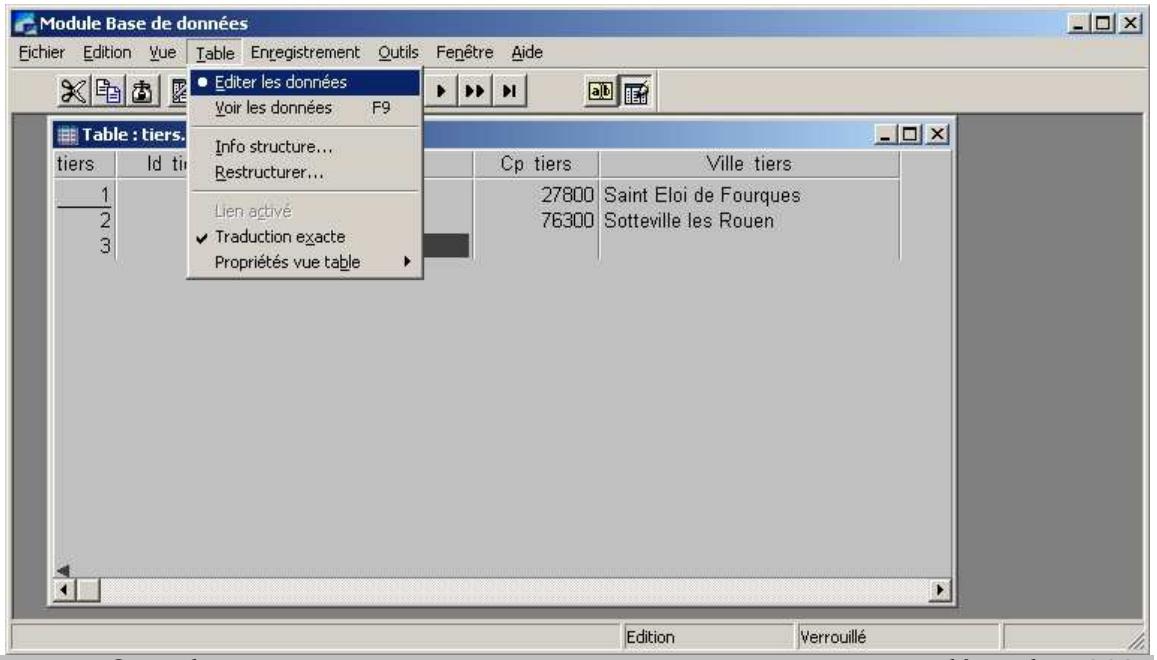

Création de la table Paradox 7 : (Sans titre)
Description des champs :
Nom de champ
Type
Taille
Index
1
Id_tiers
+
Propriétés de la table :
Contrôles de validité
Définir...
1. Champ obligatoire
2. Valeur minimum :
3. Valeur maximum :
4. Valeur par défaut :
5. Modèle :
Cochez cette case pour rendre obligatoire la saisie dans ce champ.
Assistance...
Reprendre...
Enregistrer sous...
Annuler
Aide

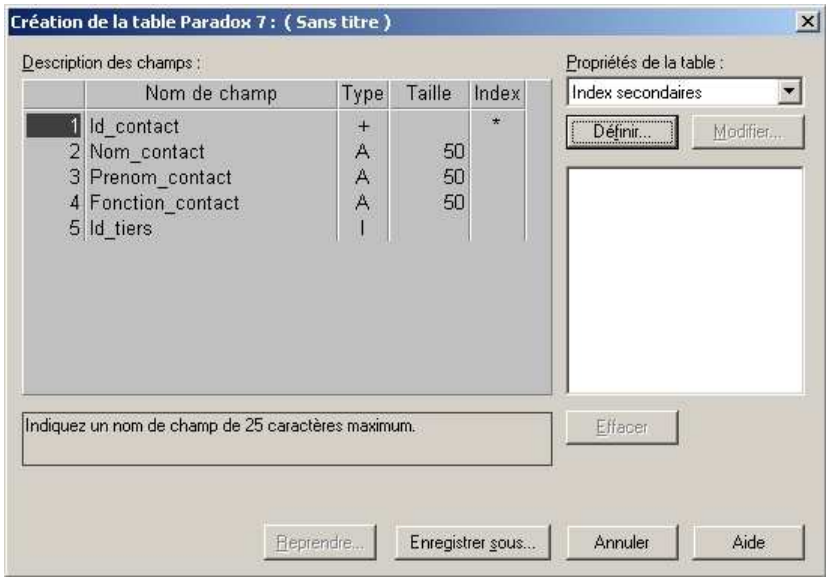

Enregistrement de la table sous
Enregistrer dans :
Paradox
Nom du fichier :
tiers
Type :
Paradox (*.db)
Alias :
TRAVAIL:
Options :
Afficher la table
Ajouter des données à la nouvelle table
Enregistrer
Annuler


Restructuration de la table Paradox 7 : tiers.db
Description des champs :
Nom de champ
Type
Taille
Index
1 Id_tiers + *
2 Rs_tiers A 50
3 Cp_tiers I
4 Ville_tiers A 50
Propriétés de la table :
Index secondaires
Définir...
Modifier...
Indiquez un nom de champ de 25 caractères maximum.
Effacer
Compactage
Enregistrer
Enregistrer sous...
Annuler
Aide

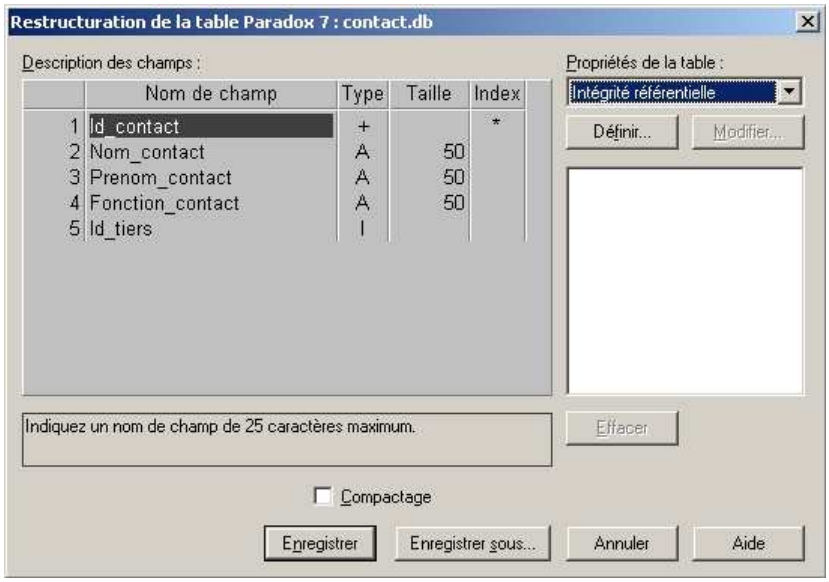

Définition d'un index secondaire
Champs :
Id_tiers
Rs_tiers
Cp_tiers
Ville_tiers
Champs indexés :
Rs_tiers
Cp_tiers
Modifier l'ordre :
Options de l'index
Unique
Maintenu
Différence MAJ/min
Décroissant
OK
Annuler
Aide

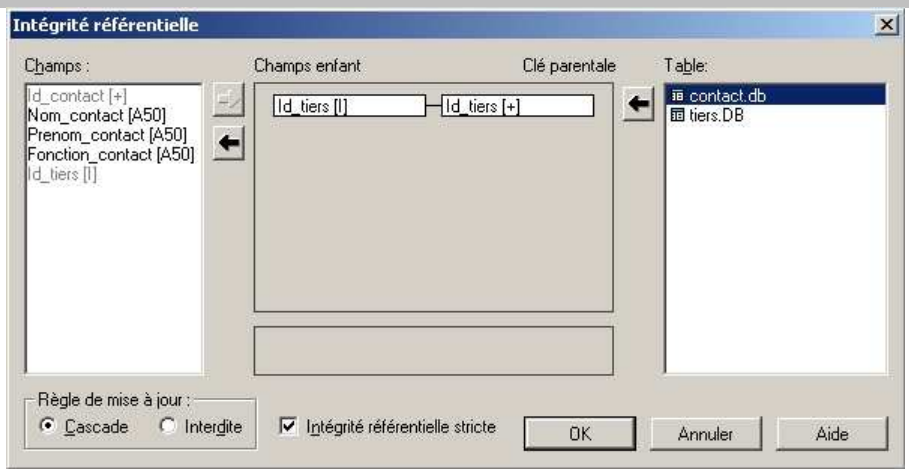

Enregistrement de l'index
Nom d'index :
uq_tiers
OK
Annuler
Aide


Module Base de données
Fichier Edition Vue Table Enregistrement Outils Fenêtre Aide
Editer les données
Voir les données F9
Info structure...
Restructurer...
Lien activé
Traduction exacte
Propriétés vue table
Table : tiers.
tiers
Id ti
Cp tiers
Ville tiers
1 27800 Saint Eloi de Fourques
2 76300 Sotteville les Rouen
3
Edition
Verrouillé

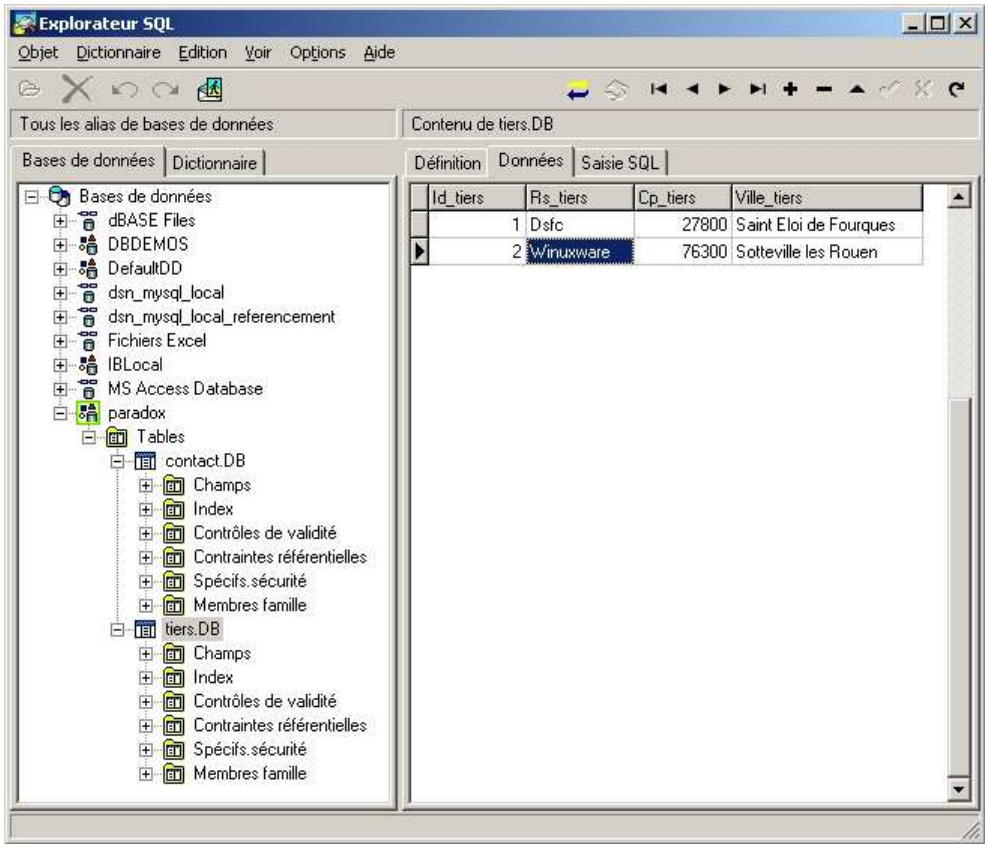

Création de la table Paradox 7 : (Sans titre)
Description des champs :
Nom de champ
Type
Taille
Index
1 Id_contact + *
2 Nom_contact A 50
3 Prenom_contact A 50
4 Fonction_contact A 50
5 Id_tiers I
Propriétés de la table :
Index secondaires
Définir...
Modifier...
Indiquez un nom de champ de 25 caractères maximum.
Effacer
Reprendre...
Enregistrer sous...
Annuler
Aide


Définition d'un index secondaire
Champs :
Id_contact
Nom_contact
Prenom_contact
Fonction_contact
Id_tiers
Champs indexés :
Nom_contact
Prenom_contact
Fonction_contact
Id_tiers
Modifier l'ordre :
Options de l'index
Unique
Maintenu
Différence MAJ/min
Décroissant
OK
Annuler
Aide


Restructuration de la table Paradox 7 : contact.db
Description des champs :
Nom de champ
Type
Taille
Index
1 Id_contact + *
2 Nom_contact A 50
3 Prenom_contact A 50
4 Fonction_contact A 50
5 Id_tiers I
Propriétés de la table :
Intégrité référentielle
Définir...
Modifier...
Indiquez un nom de champ de 25 caractères maximum.
Effacer
Compactage
Enregistrer
Enregistrer sous...
Annuler
Aide



C. L'explorateur SQL

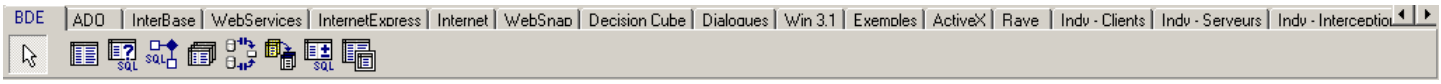


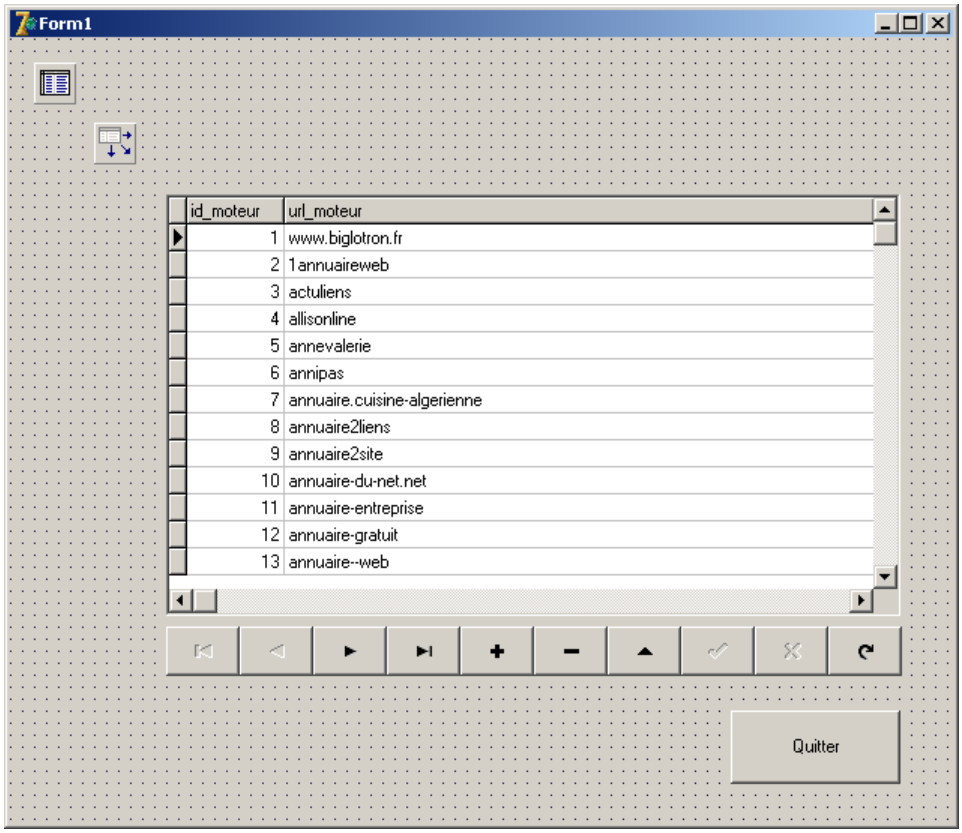
VII. Borland Database Engine, l'outil de connectivité signé Borland

A. Bde

Bde permet de connecter les applications Delphi à un moteur de base de données via Odbc.
Pour créer votre connexion Odbc à partir de la station de travail, allez dans Démarrer|Paramètres|Panneau de configuration|Outils d'administration|Source de données (ODBC).

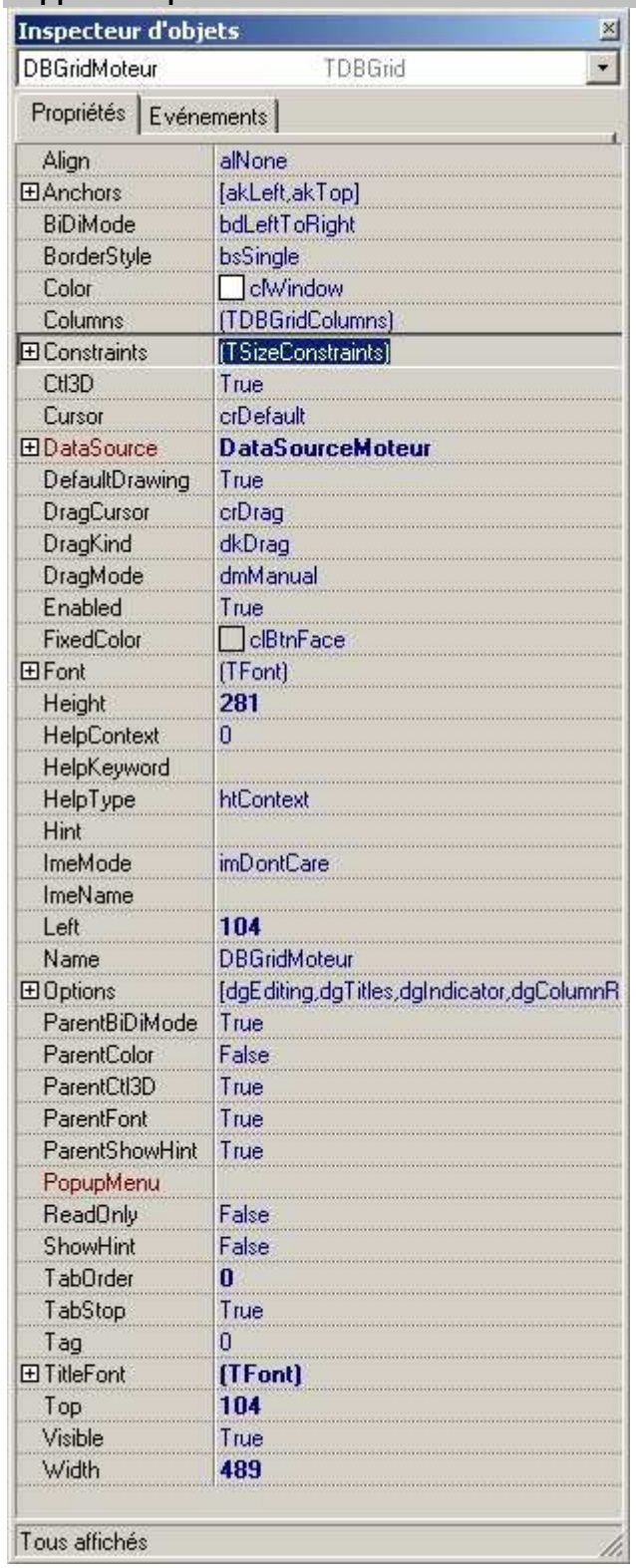
B. Connexion à partir d'une application Delphi

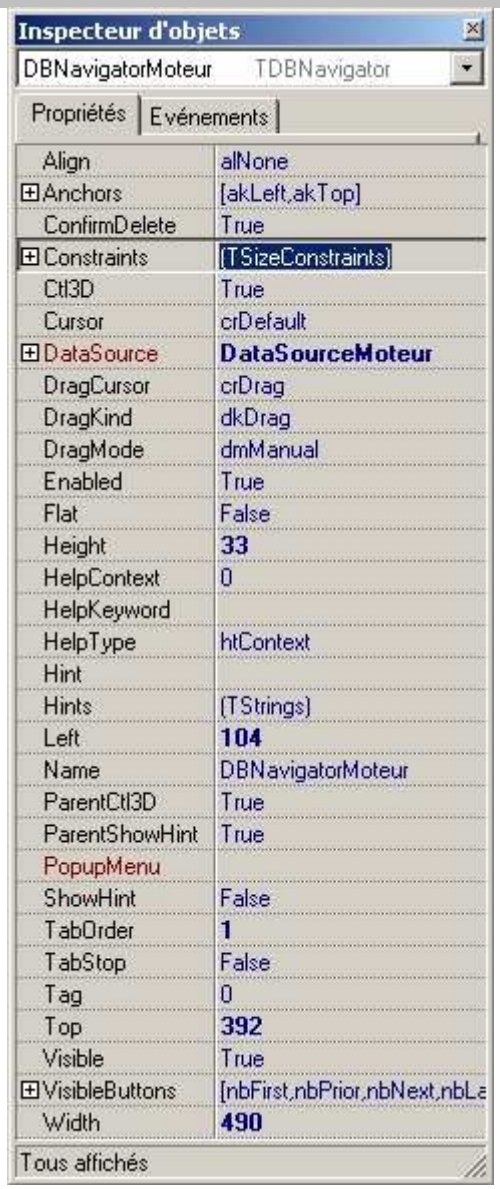





Inspecteur d'objets
TableMoteur TTable
Propriétés Evénements
Active True
AutoCalcFields True
AutoRefresh True
CachedUpdates False
Constraints (TCheckConstraints)
DatabaseName dsn_mysql_local_referencement
DefaultIndex True
Exclusive False
FieldDefs (TFieldDefs)
Filter
Filtered False
FilterOptions []
IndexDefs (TIndexDefs)
IndexFieldNames
IndexFiles (TIndexFiles)
IndexName
MasterFields
MasterSource
Name TableMoteur
ObjectView False
ReadOnly False
SessionName Default
StoreDefs False
TableName moteur
TableType ttDefault
Tag 0
UpdateMode upWhereAll
UpdateObject
Tous affichés


Inspecteur d'objets
DataSourceMoteur
TDataSource
Propriétés
Evénements
AutoEdit
True
DataSet
TableMoteur
Active
True
AutoCalcFields
True
AutoRefresh
True
CachedUpdates
False
Constraints
(TCheckConstraints)
DatabaseName
dsn_mysql_local_referencement
DefaultIndex
True
Exclusive
False
FieldDefs
(TFieldDefs)
Filter
Filtered
False
FilterOptions
[]
IndexDefs
(TIndexDefs)
IndexFieldName
IndexFiles
(TIndexFiles)
IndexName
MasterFields
MasterSource
ObjectView
False
ReadOnly
False
SessionName
Default
StoreDefs
False
TableName
moteur
TableType
ttDefault
Tag
0
UpdateMode
upWhereAll
UpdateObject
Enabled
True
Name
DataSourceMoteur
Tag
0
Tous affichés


Inspecteur d'objets
DBGridMoteur TDBGrid
Propriétés Evénements
Align alNone
Anchors [akLeft,akTop]
BiDiMode bdLeftToRight
BorderStyle bsSingle
Color clWindow
Columns (TDBGridColumns)
Constraints (TSizeConstraints)
Ctl3D True
Cursor crDefault
DataSource DataSourceMoteur
DefaultDrawing True
DragCursor crDrag
DragKind dkDrag
DragMode dmManual
Enabled True
FixedColor clBtnFace
Font (TFont)
Height 281
HelpContext 0
HelpKeyword
HelpType htContext
Hint
ImeMode imDontCare
ImeName
Left 104
Name DBGridMoteur
Options [dgEditing,dgTitles,dgIndicator,dgColumnR
ParentBiDiMode True
ParentColor False
ParentCtl3D True
ParentFont True
ParentShowHint True
PopupMenu
ReadOnly False
ShowHint False
TabOrder 0
TabStop True
Tag 0
TitleFont (TFont)
Top 104
Visible True
Width 489
Tous affichés



VIII. Sources

Sources	http://delphi.developpez.com/sources/
Comprendre CLX	http://fr.wikipedia.org/wiki/CLX_(Delphi)
Free Pascal Compiler	http://www.freepascal.org/download.var
Lazarus, un EDI FPC et FreeClx	http://www.lazarus.freepascal.org/
Outils d'aministration Firebird	http://www.ibphoenix.com/main.nfs?a=ibphoenix&page=ibp_admin_tools
Cours	http://delphi.developpez.com/cours/
Faq	http://delphi.developpez.com/faq/

IX. Annexes : Routines

A. Char

Chr,fonction	Renvoie le caractère correspondant àune valeur ASCII.
FillChar,procédure	Remplit une succession d'octets avec la valeur spécifiée.
UpCase,fonction	Convertit un caractère en majuscules.

B. Dates

CheckSqlTimeStamp,procédure	Vérifie si une valeur TSQLTimeStamp représente une date et une heure valides.
CompareDate,fonction	Indique la relation entre les parties date de deux valeurs TDateTime.
CompareDateTime,fonction	Indique la relation entre deux valeurs TDateTime.
CompareTime,fonction	Indique la relation entre les parties heure de deux valeurs TDateTime.
CurrentYear,fonction	Renvoie l'année en cours.
Date,fonction	Renvoie la date en cours.
DateOf,fonction	Enlève la partie heure d'une valeur TDateTime.
DateTimeToFileDate,fonction	Convertit un objet TDateTime en marque horaire OS.
DateTimeToSQLTimeStamp,fonction	Convertit une valeur TDateTime en valeur TSQLTimeStamp.
DateTimeToStr,fonction	Convertit une valeur TDateTime en chaîne.
DateTimeToString,procédure	Convertit une valeur TDateTime en chaîne àl'aide du format spécifié.
DateTimeToSystemTime,procédure	Convertit une valeur TDateTime en type heure système de l'API Win32.
DateTimeToTimeStamp,fonction	Convertit une valeur TDateTime en valeur TTimeStamp correspondante.
DateToStr,fonction	Convertit une valeur TDateTime en chaîne.
DayOf,fonction	Renvoie le jour du mois représentépar une valeur TDateTime.
DayOfTheMonth,fonction	Renvoie le jour du mois représentépar une valeur TDateTime.
DayOfTheWeek,fonction	Renvoie le jour de la semaine représentée par une valeur TDateTime.
DayOfTheYear,fonction	Renvoie le nombre de jours compris entre une valeur TDateTime spécifiée et le 31 décembre de l'année antérieure.
DayOfWeek,fonction	Renvoie le jour de la semaine d'une date spécifiée.
DaysBetween,fonction	Renvoie le nombre de jours entiers séparant deux valeurs TDateTime spécifiées.
DaysInAMonth,fonction	Renvoie le nombre de jours compris dans le mois spécifiéd'une année donnée.
DaysInAYear,fonction	Renvoie le nombre de jours compris dans une année spécifiée.
DaysInMonth,fonction	Renvoie le nombre de jours compris dans le mois d'une valeur TDateTime spécifiée.
DaysInYear,fonction	Renvoie le nombre de jours compris dans l'année d'une valeur TDateTime spécifiée.
DaySpan,fonction	Renvoie le nombre de jours,y compris les jours non entiers,séparant deux valeurs TDateTime spécifiées.
DecodeDate,procédure	Scinde TDateTime en valeurs Année,Mois et Jour.
DecodeDateDay,procédure	Renvoie l'année et le jour de l'année d'un objet TDateTime spécifié.
DecodeDateFully,fonction	Renvoie les valeurs Année,Mois,Jour et Jour de la semaine d'une valeur TDateTime.
DecodeDateMonthWeek,procédure	Renvoie l'année,le mois,la semaine du mois et le jour de la semaine d'un objet TDateTime spécifié.
DecodeDateTime,procédure	Scinde un objet TDateTime en valeurs

	Année,Mois,Jour,Heure,Minute,Seconde et Milliseconde.
DecodeDateWeek,procédure	Renvoie l'année,la semaine de l'année et le jour de la semaine d'une valeur TDateTime spécifiée.
DecodeDayOfWeekInMonth,procédure	Pour une valeur TDateTime donnée,renvoie l'année,le mois,le jour de la semaine et le compte de ce jour de la semaine dans le mois.
DecodeTime,procédure	Scinde TDateTime en heures,minutes,secondes et millisecondes.
EncodeDate,,fonction,fonction	Renvoie un type TDateTime pour une Année,un Mois et un Jour spécifiés.
EncodeDateDay,fonction	Renvoie un TDateTime qui représente un jour spécifiéde l'année pour une année donnée.
EncodeDateMonthWeek,fonction	Renvoie un TDateTime qui représente un jour d'une semaine d'un mois et d'une année spécifiés.
EncodeDateTime,fonction	Renvoie un TDateTime pour une année,un mois,un jour,une heure,une minute,une seconde et une milliseconde spécifiés.
EncodeDateWeek,fonction	Renvoie un TDateTime qui représente un jour d'une semaine d'une année spécifiés.
EncodeDayOfWeekInMonth,fonction	Renvoie un TDateTime qui représente un jour d'une semaine d'un mois et d'une année spécifiés.
EncodeTime,fonction	Renvoie une valeur TDateTime pour les heures,les minutes,les secondes et les millisecondes spécifiées.
EndOfADay,fonction	Renvoie un TDateTime qui représente la dernière milliseconde d'un jour spécifié.
EndOfAMonth,fonction	Renvoie un TDateTime qui représente la dernière milliseconde du dernier jour d'un mois spécifié.
EndOfAWeek,fonction	Renvoie une valeur TDateTime qui représente la dernière milliseconde d'un jour spécifiéd'une semaine donnée.
EndOfAYear,fonction	Renvoie un TDateTime qui représente la dernière milliseconde d'une année spécifiée.
EndOfTheDay,fonction	Renvoie un TDateTime qui représente la dernière milliseconde du jour identifiépar un TDateTime spécifié.
EndOfTheMonth,fonction	Renvoie un TDateTime qui représente la dernière milliseconde du dernier jour du mois identifiépar un TDateTime spécifié.
EndOfTheWeek,fonction	Renvoie un TDateTime qui représente la dernière milliseconde du dernier jour de la semaine identifiée par un TDateTime spécifié.
EndOfTheYear,fonction	Renvoie un TDateTime qui représente la dernière milliseconde du dernier jour de l'année identifiée par un TDateTime spécifié.
FormatDateTime,fonction	Formate un objet TDateTime.
HourOf,fonction	Renvoie l'heure du jour représentépar une valeur TDateTime.
HourOfTheDay,fonction	Renvoie l'heure du jour représentépar une valeur TDateTime.
HourOfTheMonth,fonction	Renvoie le nombre d'heures séparant une valeur TDateTime spécifiée et le premier jour du mois à00h00mn.
HourOfTheWeek,fonction	Renvoie le nombre d'heures séparant une valeur TDateTime spécifiée et le premier jour de la semaine à00h00mn.
HourOfTheYear,fonction	Renvoie le nombre d'heures séparant une valeur TDateTime spécifiée et le premier jour de l'année à00h00mn.
HoursBetween,fonction	Renvoie le nombre d'heures entières séparant deux valeurs TDateTime spécifiées.
HourSpan,fonction	Renvoie le nombre d'heures,y compris les heures non entières,séparant deux valeurs TDateTime spécifiées.
IncAMonth,procédure	Incrémente les données de date d'un mois.
IncDay,fonction	Renvoie une date décalée d'un certain nombre de jours.
IncHour,fonction	Renvoie une valeur date/heure décalée d'un certain nombre d'heures.
IncMilliSecond,fonction	Renvoie une valeur date/heure décalée d'un certain nombre de

	millisecondes.
IncMinute,fonction	Renvoie une valeur date/heure décalée d'un certain nombre de minutes.
IncMonth,fonction	Renvoie une date décalée d'un certain nombre de mois.
IncSecond,fonction	Renvoie une valeur date/heure décalée d'un certain nombre de secondes.
IncWeek,fonction	Renvoie une date décalée d'un certain nombre de semaines.
IncYear,fonction	Renvoie une date décalée d'un certain nombre d'années.
IsInLeapYear,fonction	Indique si la valeur TDateTime spécifiée appartient àune année bissextile.
IsLeapYear,fonction	Indique si l'année spécifiée est bissextile.
IsPM,fonction	Indique si la partie heure d'une valeur TDateTime spécifiée intervient l'après-midi.
IsSameDay,fonction	Indique si la valeur TDateTime spécifiée appartient au même jour qu'une date donnée.
IsToday,fonction	Indique si la valeur TDateTime spécifiée intervient àla date en cours.
IsValidDate,fonction	Indique si une année,un mois et un jour spécifiés représentent une date valide.
IsValidDateDay,fonction	Indique si une année et un jour de l'année spécifiés représentent une date valide.
IsValidDateMonthWeek,fonction	Indique si une année,un mois,une semaine du mois et un jour de la semaine spécifiés représentent une date valide.
IsValidDateTime,fonction	Indique si une année,un mois,un jour,une heure,une minute,une seconde et une milliseconde spécifiés représentent une valeur date/heure valide.
IsValidDateWeek,fonction	Indique si une année,une semaine de l'année et un jour de la semaine spécifiés représentent une date valide.
IsValidTime,fonction	Indique si une heure,une minute,une seconde et une milliseconde spécifiées représentent une valeur date/heure valide.
MilliSecondOf,fonction	Renvoie la milliseconde de la seconde représentée par une valeur TDateTime.
MilliSecondOfTheDay,fonction	Renvoie le nombre de millisecondes séparant une valeur TDateTime spécifiée et le début du même jour.
MilliSecondOfTheHour,fonction	Renvoie le nombre de millisecondes séparant une valeur TDateTime spécifiée et le début de la même heure.
MilliSecondOfTheMinute,fonction	Renvoie le nombre de millisecondes séparant une valeur TDateTime spécifiée et le début de la même minute.
MilliSecondOfTheMonth,fonction	Renvoie le nombre de millisecondes séparant une valeur TDateTime spécifiée et le début du mois.
MilliSecondOfTheSecond,fonction	Renvoie la milliseconde de la seconde représentée par une valeur TDateTime.
MilliSecondOfTheWeek,fonction	Renvoie le nombre de millisecondes séparant une valeur TDateTime spécifiée et 00h00mn00s00cs le premier jour de la semaine.
MilliSecondOfTheYear,fonction	Renvoie le nombre de millisecondes séparant une valeur TDateTime spécifiée et 00h00mn00s00cs le premier jour de l'année.
MilliSecondsBetween,fonction	Renvoie le nombre de millisecondes séparant deux valeurs TDateTime spécifiées.
MilliSecondSpan,fonction	Renvoie le nombre de millisecondes séparant deux valeurs TDateTime spécifiées.
MinuteOf,fonction	Renvoie la minute de l'heure représentée par une valeur TDateTime.
MinuteOfTheDay,fonction	Renvoie le nombre de minutes séparant une valeur TDateTime

	spécifiée et 00h00mn le même jour.
MinuteOfTheHour,fonction	Renvoie le nombre de minutes séparant une valeur TDateTime spécifiée et le début de la même heure.
MinuteOfTheMonth,fonction	Renvoie le nombre de minutes séparant une valeur TDateTime spécifiée et le premier jour du mois à00h00mn.
MinuteOfTheWeek,fonction	Renvoie le nombre de minutes séparant une valeur TDateTime spécifiée et le premier jour de la semaine à00h00mn.
MinuteOfTheYear,fonction	Renvoie le nombre de minutes séparant une valeur TDateTime spécifiée et le premier jour de l'année à00h00mn.
MinutesBetween,fonction	Renvoie le nombre de minutes séparant deux valeurs TDateTime spécifiées.
MinuteSpan,fonction	Renvoie le nombre de minutes,y compris les minutes non entières,séparant deux valeurs TDateTime spécifiées.
MonthOf,fonction	Renvoie le mois de l'année représentée par une valeur TDateTime.
MonthOfTheYear,fonction	Renvoie le mois de l'année représentée par une valeur TDateTime.
MonthsBetween,fonction	Renvoie le nombre approximatif de mois séparant deux valeurs TDateTime spécifiées.
MonthSpan,fonction	Renvoie le nombre approximatif de mois,y compris les mois non entiers,séparant deux valeurs TDateTime spécifiées.
MSecsToTimeStamp,fonction	Convertit un nombre spécifiéde millisecondes en une valeur TTimeStamp.
Now,fonction	Renvoie la date et l'heure en cours.
NthDayOfWeek,fonction	Renvoie l'occurrence du jour de la semaine représentépar une valeur TDateTime.
RecodeDate,fonction	Remplace la partie date d'une valeur TDateTime spécifiée.
RecodeDateTime,fonction	Remplace de façon sélective les parties d'une valeur TDateTime spécifiée.
RecodeDay,fonction	Remplace le jour du mois d'une valeur TDateTime spécifiée.
RecodeHour,fonction	Remplace l'heure du jour d'une valeur TDateTime spécifiée.
RecodeMilliSecond,fonction	Remplace la milliseconde de la seconde d'une valeur TDateTime spécifiée.
RecodeMinute,fonction	Remplace la minute de l'heure d'une valeur TDateTime spécifiée.
RecodeMonth,fonction	Remplace le mois de l'année d'une valeur TDateTime spécifiée.
RecodeSecond,fonction	Remplace la seconde de la minute d'une valeur TDateTime spécifiée.
RecodeTime,fonction	Remplace la partie heure d'une valeur TDateTime spécifiée.
RecodeYear,fonction	Remplace l'année d'une valeur TDateTime spécifiée.
ReplaceDate,procédure	Remplace la partie date d'une valeur TDateTime par une date spécifiée.
ReplaceTime,procédure	Remplace la partie heure d'une valeur TDateTime par une date spécifiée.
SameDate,fonction	Indique si deux valeurs TDateTime représentent les mêmes année,mois et jour.
SameDateTime,fonction	Indique si deux valeurs TDateTime représentent les mêmes année,mois,jour,heure,minute,seconde et milliseconde.
SameTime,fonction	Indique si deux valeurs TDateTime représentent la même heure du jour,en ignorant la partie date.
SecondOf,fonction	Renvoie la seconde de la minute représentée par une valeur TDateTime.
SecondOfTheDay,fonction	Renvoie le nombre de secondes séparant une valeur TDateTime spécifiée et 00h00mn00s le même jour.
SecondOfTheHour,fonction	Renvoie le nombre de secondes séparant une valeur TDateTime spécifiée et le début de la même heure.
SecondOfTheMinute,fonction	Renvoie le nombre de secondes séparant une valeur TDateTime

	spécifiée et le début de la même minute.
SecondOfTheMonth,fonction	Renvoie le nombre de secondes séparant une valeur TDateTime spécifiée et le premier jour du mois à00h00mn00s.
SecondOfTheWeek,fonction	Renvoie le nombre de secondes séparant une valeur TDateTime spécifiée et le premier jour de la semaine à00h00mn00s.
SecondOfTheYear,fonction	Renvoie le nombre de secondes séparant une valeur TDateTime spécifiée et le premier jour de l'année à00h00mn00s.
SecondsBetween,fonction	Renvoie le nombre de secondes séparant deux valeurs TDateTime spécifiées.
SecondSpan,fonction	Renvoie le nombre de secondes,y compris les secondes non entières,séparant deux valeurs TDateTime spécifiées.
SQLDayOfWeek,fonction	Indique le jour de la semaine quand une valeur TSQLTimeStamp est fournie.
SQLTimeStampToDateTime,fonction	Convertit une valeur TSQLTimeStamp en valeur TDateTime.
SQLTimeStampToStr,fonction	Convertit une valeur TSQLTimeStamp en chaîne.
StartOfADay,fonction	Renvoie un TDateTime qui représente 00h00mn00s00cs un jour spécifié.
StartOfAMonth,fonction	Renvoie un TDateTime qui représente 00h00mn00s00cs le premier jour d'un mois spécifié.
StartOfAWeek,fonction	Renvoie un TDateTime qui représente le premier moment d'un jour spécifiéd'une semaine donnée.
StartOfAYear,fonction	Renvoie un TDateTime qui représente le premier moment du premier jour d'une année spécifiée.
StartOfTheDay,fonction	Renvoie un TDateTime qui représente 00h00mn00s00cs le jour identifiépar un TDateTime spécifié.
StartOfTheMonth,fonction	Renvoie un TDateTime qui représente 00h00mn00s00cs le premier jour du mois identifiépar un TDateTime spécifié.
StartOfTheWeek,fonction	Renvoie un TDateTime qui représente 00h00mn00s00cs le premier jour de la semaine identifiée par un TDateTime spécifié.
StartOfTheYear,fonction	Renvoie un TDateTime qui représente 00h00mn00s00cs le premier jour de l'année identifiée par un TDateTime spécifié.
StrToDate,fonction	Convertit une chaîne en valeur TDateTime.
StrToDateDef,fonction	Convertit une chaîne en valeur TDateTime,avec renvoi de Default en cas d'erreur.
StrToDateTime,fonction	Convertit une chaîne en valeur TDateTime.
StrToDateTimeDef,fonction	Convertit une chaîne en valeur TDateTime avec renvoi de Default en cas d'erreur.
StrToSQLTimeStamp,fonction	Convertit une chaîne en valeur TSQLTimeStamp.
StrToTime,fonction	Convertit une chaîne en valeur TDateTime.
StrToTimeDef,fonction	Convertit une chaîne en valeur TDateTime avec renvoi de Default en cas d'erreur.
SystemTimeToDateTime,fonction	Convertit une valeur heure système en une valeur TDateTime.
Time,GetTime,,fonctions	Renvoie l'heure en cours.
TimeOf,fonction	Enlève la partie date d'une valeur TDateTime.
TimeStampToDateTime,fonction	Convertit une valeur TTimeStamp en la valeur TDateTime correspondante.
TimeStampToMSecs,fonction	Convertit une valeur TTimeStamp en un nombre absolu de millisecondes.
TimeToStr,fonction	Renvoie une chaîne qui représente une valeur TDateTime.
Today,fonction	Renvoie une valeur TDateTime qui représente la date en cours.
Tomorrow,fonction	Renvoie une valeur TDateTime qui représente la jour suivant.
TryEncodeDate,fonction	Crée une valeur TDateTime qui représente une année,un mois et un jour spécifiés.
TryEncodeDateDay,fonction	Calcule la valeur TDateTime qui représente un jour spécifiéde

	l'année pour une année donnée.
TryEncodeDateMonthWeek,fonction	Calcule la valeur TDateTime qui représente un jour d'une semaine d'un mois et d'une année spécifiés.
TryEncodeDateTime,fonction	Calcule la valeur TDateTime pour une année,un mois,un jour,une heure,une minute,une seconde et une milliseconde spécifiés.
TryEncodeDateWeek,fonction	Calcule la valeur TDateTime qui représente un jour d'une semaine d'une année spécifiés.
TryEncodeDayOfWeekInMonth,fonction	Calcule un TDateTime qui représente un jour d'une semaine d'un mois et d'une année spécifiés.
TryEncodeTime,fonction	Crée une valeur TDateTime pour les heures,les minutes,les secondes et les millisecondes spécifiées.
TryRecodeDateTime,fonction	Remplace de façon sélective les parties d'une valeur TDateTime spécifiée.
TryStrToDate,fonction	Convertit une chaîne en valeur TDateTime,avec renvoi d'un code de succès booléen.
TryStrToDateTime,fonction	Convertit une chaîne en valeur TDateTime avec renvoi d'un code de succès booléen.
TryStrToSQLTimeStamp,fonction	Convertit une chaîne en valeur TSQLTimeStamp.
TryStrToTime,fonction	Convertit une chaîne en valeur TDateTime avec renvoi de Default en cas d'erreur.
WeekOf,fonction	Renvoie la semaine de l'année représentée par une valeur TDateTime.
WeekOfTheMonth,fonction	Renvoie la semaine du mois représentépar une valeur TDateTime.
WeekOfTheYear,fonction	Renvoie la semaine de l'année représentée par une valeur TDateTime.
WeeksBetween,fonction	Renvoie le nombre de semaines entières séparant deux valeurs TDateTime spécifiées.
WeeksInAYear,fonction	Renvoie le nombre de semaines comprises dans une année spécifiée.
WeeksInYear,fonction	Renvoie le nombre de semaines comprises dans l'année d'une valeur TDateTime spécifiée.
WeekSpan,fonction	Renvoie le nombre de semaines,y compris les semaines non entières,séparant deux valeurs TDateTime spécifiées.
WithinPastDays,fonction	Indique si deux dates sont séparées l'une de l'autre par un nombre de jours inférieur ou égal au nombre spécifié.
WithinPastHours,fonction	Indique si deux valeurs date/heure sont séparées l'une de l'autre par un nombre d'heures inférieur ou égal au nombre spécifié.
WithinPastMilliSeconds,fonction	Indique si deux valeurs date/heure sont séparées l'une de l'autre par un nombre de millisecondes inférieur ou égal au nombre spécifié.
WithinPastMinutes,fonction	Indique si deux valeurs date/heure sont séparées l'une de l'autre par un nombre de minutes inférieur ou égal au nombre spécifié.
WithinPastMonths,fonction	Indique si deux valeurs date/heure sont séparées l'une de l'autre par un nombre de mois inférieur ou égal au nombre spécifié.
WithinPastSeconds,fonction	Indique si deux valeurs date/heure sont séparées l'une de l'autre par un nombre de secondes inférieur ou égal au nombre spécifié.
WithinPastWeeks,fonction	Indique si deux valeurs date/heure sont séparées l'une de l'autre par un nombre de semaines inférieur ou égal au nombre spécifié.
WithinPastYears,fonction	Indique si deux valeurs date/heure sont séparées l'une de l'autre par un nombre d'années inférieur ou égal au nombre spécifié.
YearOf,fonction	Renvoie l'année représentée par une valeur TDateTime.
YearsBetween,fonction	Renvoie le nombre approximatif d'années séparant deux valeurs TDateTime spécifiées.
YearSpan,fonction	Renvoie le nombre approximatif d'années,y compris les années

	non entières,séparant deux valeurs TDateTime spécifiées.
Yesterday,fonction	Renvoie une valeur TDateTime qui représente la jour précédent.

C. Trigo

ArcCos,fonction	Calcule le cosinus inverse d'une valeur donnée.
ArcCosh,fonction	Calcule le cosinus hyperbolique inverse d'une valeur donnée.
ArcCot,fonction	Calcule la cotangente inverse d'une valeur donnée.
ArcCotH,fonction	Calcule la cotangente hyperbolique inverse d'une valeur donnée.
ArcCsc,fonction	Calcule la cosécante inverse d'une valeur donnée.
ArcCscH,fonction	Calcule la cosécante hyperbolique inverse d'une valeur donnée.
ArcSec,fonction	Calcule la sécante inverse d'une valeur donnée.
ArcSecH,fonction	Calcule la sécante hyperbolique inverse d'une valeur donnée.
ArcSin,fonction	Calcule le sinus inverse d'une valeur donnée.
ArcSinh,fonction	Calcule le sinus hyperbolique inverse d'une valeur donnée.
ArcTan,fonction	Calcule l'arc-tangente de la valeur donnée.
ArcTan2,fonction	Calcule l'angle et le quadrant d'arc-tangente de la valeur donnée.
ArcTanh,fonction	Calcule la tangente hyperbolique inverse d'une valeur donnée.
Cos,fonction	Calcule le cosinus d'un angle.
Cosecant,fonction	Renvoie la cosécante d'un angle.
Cosh,fonction	Calcule le cosinus hyperbolique d'un angle.
Cot,fonction	Calcule la cotangente d'un angle.
Cotan,fonction	Calcule la cotangente d'un angle.
CotH,fonction	Calcule la cotangente hyperbolique d'un angle.
Csc,fonction	Renvoie la cosécante d'un angle.
CscH,fonction	Renvoie la cosécante hyperbolique d'un angle.
Hypot,fonction	Calcule la longueur de l'hypoténuse.
Sec,fonction	Calcule la sécante d'un angle.
Secant,fonction	Calcule la sécante d'un angle.
SecH,fonction	Calcule la sécante hyperbolique d'un angle.
Sin,fonction	Renvoie le sinus de l'angle en radians.
SinCos,procédure	Renvoie le sinus et le cosinus d'un angle.
Sinh,fonction	Sinh renvoie le sinus hyperbolique d'un angle.
Tan,fonction	Renvoie la tangente de X.
Tanh,fonction	Renvoie la tangente hyperbolique de X.

D. Conversion de types

BinToHex,procédure	Convertit une valeur binaire en sa représentation hexadécimale.
BoolToStr,fonction	Convertit une variable booléenne en chaîne.
Bounds,fonction	Renvoie le TRect d'un rectangle de dimensions données.
CompToCurrency,fonction	Convertit une valeur Comp en une valeur Currency.
CompToDouble,fonction	Convertit une valeur Comp en valeur double.
Dec,procédure	Convertit une valeur Currency en valeur Comp.
HexToBin,fonction	Convertit une chaîne de chiffres hexadécimaux en la valeur binaire correspondante.
OffsetRect,fonction	Modifie l'origine d'un rectangle selon la quantitéspécifiée.
Point,fonction	Crée une structure TPoint avec un couple de coordonnées.
Rect,fonction	Crée une structure TRect àpartir de coordonnées fournies.
Rect,fonction	Crée une structure TRect àpartir de coordonnées fournies.
SmallPoint,fonction	Crée une structure TSmallPoint àpartir d'un couple de coordonnées.
StrToBool,fonction	Convertit une chaîne en valeur booléenne.
StrToBoolDef,fonction	Convertit une chaîne en valeur booléenne,avec renvoi de Default en cas d'erreur.
StrToInt,fonction	Convertit en nombre une chaîne qui représente un entier (décimal ou hexadécimal).
StrToInt64,fonction	Convertit en nombre une chaîne qui représente un entier (décimal ou

	hexadécimal).
StrToInt64Def,fonction	Convertit en nombre une chaîne qui représente un entier (décimal ou hexadécimal).
StrToIntDef,fonction	Convertit en nombre une chaîne qui représente un entier (décimal ou hexadécimal),avec renvoi de Default en cas d'erreur.
TryStrToBool,fonction	Convertit une chaîne en valeur booléenne,avec renvoi d'un code de succès booléen.
TryStrToInt,fonction	Convertit en nombre une chaîne qui représente un entier (décimal ou hexadécimal),avec renvoi d'un code de succès booléen.
TryStrToInt64,fonction	Convertit en nombre une chaîne qui représente un entier (décimal ou hexadécimal).