5.11 Utilisation des fiches

Présentation

Une fiche est aussi appelée fenêtre. Toute application MS-WINDOWS comporte au moins une fiche. Lorsque vous créez une nouvelle **Application Fiches VCL**, Delphi crée par défaut une unité au sein de laquelle vous trouvez la déclaration **TForm1 = class(TForm)**

Cet objet TForm1 semble simple. Vous pouvez néanmoins afficher, masquer et redimensionner la fiche, ajouter ou retirer les icônes standard en haut de la fiche ou la configurer pour qu'elle fasse partie d'une application à **interface de document multiple** (**MDI**). Vous pouvez faire tout cela car la fiche a hérité de toutes les propriétés et méthodes du composant VCL TForm. Quand vous ajoutez une nouvelle fiche à votre projet, vous partez de TForm que vous personnalisez en lui ajoutant des composants, en modifiant la valeur des propriétés et en écrivant des gestionnaires d'événements.

Form1 représente une variable destinée à stocker une instance, du type de classe TForm1. Vous pouvez déclarer plusieurs instances d'un type de classe ; par exemple, pour créer plusieurs fenêtres enfant dans une application utilisant l'interface d'une application à documents multiples (MDI). Chaque instance a ses données propres, mais elle utilise le même code pour exécuter les méthodes.

Quelques propriétés et valeurs notables :

PROPRIETE	VALEUR	DESCRIPTION
FormStyle		Définit le style de la fiche
	sNormal	La fiche n'est ni une fenêtre parent MDI ni une fenêtre enfant MDI.
	fsMDIChild	La fiche est une fenêtre enfant MDI
	fsMDIForm	La fiche est une fenêtre parent MDI
Position		Agit sur la taille et la position de la fiche
	poDesigned	La fiche apparaît àl'écran àla position et avec les dimensions qu'elle avait àla conception
	poScreenCenter	La fiche conserve la taille définie àla conception mais elle est placée au centre de l'écran.
WindowState		
	wsNormal	La fiche est dans l'état normal (c'est-à-dire,ni réduite ni maximale)
	wsMinimized	La fiche est de taille réduite.
	wsMaximized	La fiche est de taille maximale.

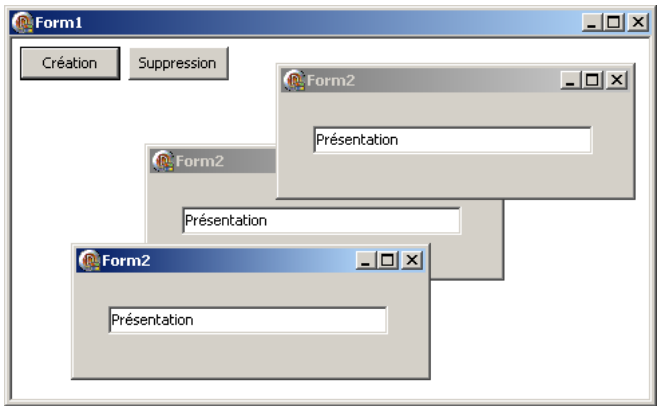
Quelques méthodes :

- La création d'une fiche est réalisée par **Application.CreateForm,**
- L'affichage d'une fiche se fait par la méthode **Show** ou **ShowModal**,
- La fermeture d'une fiche se fait par la méthode **Close**. Cependant, l'événement **OnClose** permet de réaliser des traitements avant fermeture.
- La variable **Action** de la méthode **FormClose** permet de valider ou non la fermeture de la fiche.

5.11.1 Créer une application multti-fiche MDI

Une application MDI (Multiple Document Interface) est constituée d'un certain nombre de fiches apparaissant dans une fiche principale. (Ex : un traitement de texte peut ouvrir plusieurs documents).

Nous allons en regarder les détails au travers cet exemple. Objectif : pouvoir ouvrir des fiches de type Tform2 à convenance et synchroniser les contenus de texte.



Ce projet nommé **ex10** utilise deux unités **ex10a** dans **'ex10a.pas'** et **ex10b** dans **'ex10b.pas'**. L'objet **form1** est dans ex10a.pas et **form2** dans ex10b.pas.

pour Form1 : FormStyle = fsMDIForm et WindowState = wsMaximized
pour Form2 : FormStyle = fsMDIChild

Visibilité entre les deux unités :

```
unit ex10a;

interface

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

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  public
    procedure MettreAJourTexte ;
  end;

var
  Form1: TForm1 ;
  Sl1 : TStringList ;
  MonTexte : string ;

implementation

uses ex10b;
```

```
unit ex10b;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics,   Controls, Forms, Dialogs, StdCtrls,
  ex10a;

type
  TForm2 = class(TForm)
    Edit1: TEdit;
    procedure Edit1Change(Sender: TObject);
  end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Edit1Change(Sender: TObject);
begin
  MonTexte := Edit1.Text ;
  Form1.MettreAJourTexte  ;
end;
```

Ex10a utilise des éléments de ex10b et réciproquement. Pour éviter les référencements circulaires dans un cas on place **uses** dans la partie interface , dans l'autre dans la partie implémentation. Ainsi :

- **Form2** pourra être utilisé depuis **ex10a**,
- **MonTexte** , variable globale, sera visible depuis les deux unités,
- **MettreAJourTexte** décrit dans la partie **public** de l'objet de type **TForm1** de ex10a sera accessible par **Form1.MettreAJourTexte** dans **uni10b**

TStringlist utilisé ici est un objet non visuel. La propriété **Items** du type **TlistBox** et **Lines** du type **Tmemo** sont des objets de cette nature. Dans cet exemple, il est utilisé pour mémoriser les adresses de création des objets de type **TForm2**.

EX10A.PAS

CODE	DESCRIPTION

implementation
uses ex10b;
{$R *.dfm}
 	
Var
  Form1: TForm1;
  Sl1 : TStringList ;
  MonTexte : String ;
 	
procedure TForm1.Button1Click(Sender: TObject);
begin
  Application.CreateForm(TForm2, Form2);
  sl1.Add(intToStr(integer(pointer(form2)))) ;
  Form2.Show ;
  MettreAJourTexte ;
end;
 	
procedure TForm1.Button2Click(Sender: TObject);
var
  i1 : integer ;
begin
  for i1 := 0 to sl1.Count -1 do
  begin
  Form2 := pointer(integer(strToInt(sl1[i1]))) ;
  Form2.free ;
  end;
  sl1.Clear ;
end;
 	
procedure TForm1.MettreAJourTexte ;
var
  i1 : integer ;
begin
  for i1 := 0 to sl1.Count -1 do
  begin
  Form2 := pointer(integer(strToInt(sl1[i1]))) ;
  Form2.edit1.text := montexte ;
  end;
end;
 	
initialization
  sl1 := TStringList.Create ;
finalization
  sl1.Free ;
end.
	Création d'une instance de form2 et mémorisation de l'adresse de son pointeur dans un élément de la **stringlist**
	
	Destruction de toutes les instance de form2 créés
	
	Synchronisation de tous les textes des objets Tedit de ficher form2
	
	initialization et **finalization** sont deux éléments du décrivant les instruction déclenchées respectivement en début et en fin d'exécution du programme

EX10B.PAS

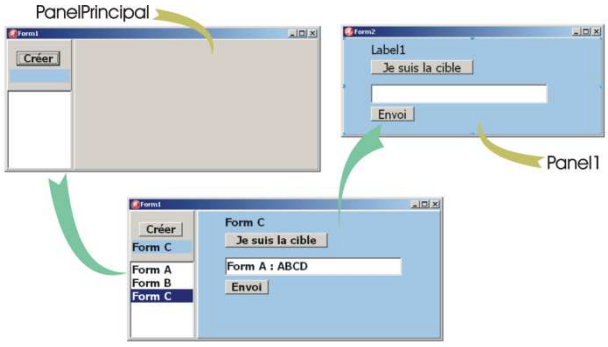
Form2 contient d'un objet **Tedit**. **Edit1Change** est associé à l'évènement **OnChange** de **Tedit**.

CODE	DESCRIPTION

procedure TForm2.Edit1Change(Sender: TObject);
begin
  MonTexte := Edit1.Text ;
  Form1.MettreAJourTexte ;
end;	

5.11.2 Créer une application multi-fiche

En partant des principes exposés avec l'exemple du damier nous allons maintenant voir un des mécanismes les plus étonnants Ce mécanisme peut être décrit de manière imagée comme une ***téléportation*** d'un groupe de composants d'une fiche à une autre. Nous allons voir l'usage qui en est fait dans une application multi-fiche.



procédures dans l'unité décrivant FORM1

CODE	DESCRIPTION

procedure TForm1.FormCreate(Sender: TObject);
begin
  lbCible.Tag := 0 ;
end;
	La propriété Tag du composant lbCible de type Tlabel est utilisé pour stocker l'adresse de l'objet fiche considéré comme cible

procedure TForm1.RemiseEnPlacePanel;
var
  i1 : integer ;
begin
  for i1 := 0 to ListBox1.Items.Count - 1 do
  begin
  form2 := ListBox1.Items.Objects[i1] as TForm2;
  form2.Panel1.Parent := form2 ;
  end;
end;
	Remise en place pour chaque fiche créée des objets Panel1 et de leur contenu dans leur objet parent initial qui est ici un objet form2

procedure TForm1.Button1Click(Sender: TObject);
begin
  RemiseEnPlacePanel ;
  form2 := Tform2.Create (form1) ;
  form2.Panel1.Parent := form1.PanelPrincipal ;
  form2.lbTitre.caption := 'Form '+char(ListBox1.Items.Count+$41) ;
  listBox1.Items.AddObject(form2.lbTitre.caption, TObject(form2))
end;
	Création d'une fiche
	
	Ajout de l'adresse en mémoire de la fiche créée (dans la **stringlist** de **listBox1**)

procedure TForm1.ListBox1DblClick(Sender: TObject);
begin
  if ListBox1.ItemIndex < 0 then exit ;
  RemiseEnPlacePanel;
  form2 := Pointer(ListBox1.Items.Objects[ListBox1.ItemIndex]) ;
  form2.Panel1.Parent := form1.PanelPrincipal ;
end;
	Visualisation du contenu da la fiche choisie

procédures dans l'unité décrivant FORM2

procedure TForm2.BtEnvoiCibleClick(Sender: TObject);
begin
  if Form1.lbCible.Tag = 0 then exit ;
  form2 := Pointer(Form1.lbCible.Tag) ;
  form2.edMessage.Text := lbTitre.caption+' : '+edMessage.Text ;
end;
 	
procedure TForm2.BtDefinirCibleClick(Sender: TObject);
begin
  Form1.lbCible.Caption := lbTitre.Caption ;
  Form1.lbCible.Tag := Integer(Pointer(self))
end;	