Delphi

بناء مكوّن (Component) جديد في دلفي (مثال عملي)

تأليف وإعداد المهندس: عصام علي

جميع الحقوق محفوظة

www.iss.almoja.net

إعداد : م . عصام علي

بناء مكوّن (Component) جديد في دلفي

سنستعرض مثالاً عن كيفية بناء مكوّن (component) جديد هو عبارة عن صندوق إدخال Inputbox مشابه لصندوق الإدخال الخاص بـــ فيجوال بيسك .

جميعنا يعرف أن هناك توابع خاصة سواء في دلفي أو في فيجوال بيسك لإظهار صندوق إدخال يقوم بعرض رسالة للمستخدم و يسمح له بادخال قيمة ما للبرنامج ليتم معالجتها . التابع المسؤول عن ذلك سواء في دلفي أو في فيجوال بيسك له نفس الاسم (INPUTBOX) .

المثال التالي يبين كيفية عمله في فيجوال بيسك :

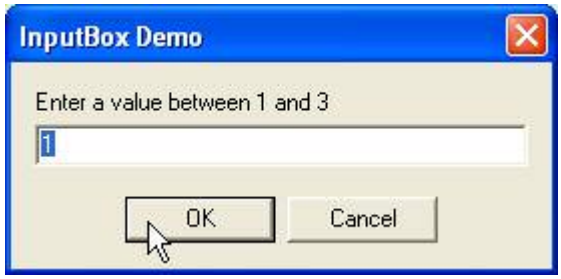
```
Private Sub CommandButton1_Click()
Dim Message, Title, Default, MyValue
Message = "Enter a value between 1 and 3"    ' Set prompt.
Title = "InputBox Demo"    ' Set title.
Default = "1"    ' Set default.
' Display message, title, and default value.
MyValue = InputBox(Message, Title, Default)

End Sub
```

لتكون النتيجة كما في الشكل :

أما في دلفي فتابع INPUTBOX هو أحد توابع الوحدة Dialogs و المثال التالي يبين طريقة استخدامه :

```
procedure TForm1.Button1Click(Sender: TObject);
VAR T_Message, Title, Default, MyValue :String;
begin
T_Message := 'Enter a value between 1 and 3';// set prompet
Title := 'InputBox Demo';    // Set title.
Default := '1';    // Set default.
// Display T_Message, title, and default value.
MyValue := InputBox(Title, T_Message, Default);

end;
```

لتكون النتيجة على الشكل التالي :



إعداد : م . عصام علي

وطبعاً نلاحظ الفرق الواضح بين شكل الصندوقين، الآن إذا أردنا على سبيل التدريب أن نصنع في دلفي صندوق إدخال مشابهاً للذي في فيجوال بيسك ولكن على شكل مكوّن (Component) وليس تابع (Function) .

بناء أي مكوّن يبدأ بالطريقة نفسها : إنشاء وحدة المكون (Component unit)، تعريف صنف المكون (Component class) تسجيل المكون (Rigister)، ترجمة المكون (Compile)، وأخيراً تثبيت المكون(Install) في أحد لوحات المكونات.

خلق وتسجيل المكوّن (Creating and registering the component) :

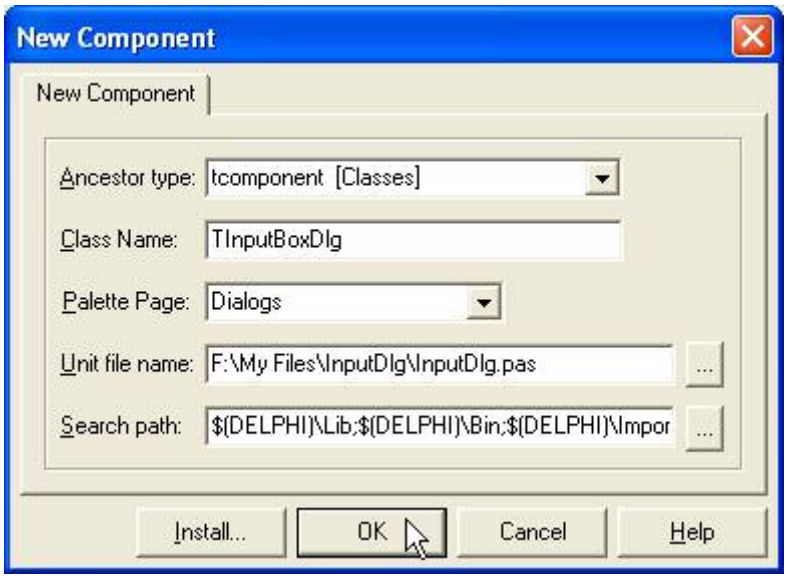
للقيام بذلك نتبّع الخطوات التالية :

سمي وحدة المكون InputDlg.

عرّف صنف جديد TInputBoxDlg .

سجل المكون الجديد في صفحة Dialogs .

إذا لم ترغب بإنشاء وحدة المكون يدوياً يمكنك أن تطلب من دلفي إنشائها عن طريق قائمة Component ثم نختار New Component و ندخل الخيارات كما في الشكل التالي :



اضغط OK وسيقوم دلفي بإنشاء وحدة المكوّن، الآن قم بإضافة الوحدات التالية إلى قائمة USES : { Controls, Forms}

بحيث يصبح شكل وحدة المكون كما يلي :

```
unit InputDlg;

interface

uses
  SysUtils, Classes,Controls, Forms,Windows;

type
  TInputBoxDlg= class(TComponent)
  private
    { Private declarations }
  protected
    { Protected declarations }
  public
```

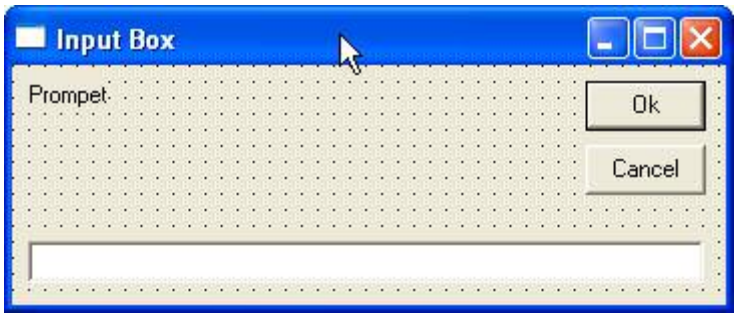
إعداد : م . عصام علي

```
  { Public declarations }
 published
  { Published declarations }
 end;

procedure Register;

implementation

procedure Register;
begin
 RegisterComponents('Dialogs', [TInputBoxDlg]);
end;

end.
```

بناء واجهة المكوّن (Creating the component interface) :

إنشاء النموذج:

أول خطوة من خطوات بناء واجهة هذا المكون هي إنشاء نموذج (Form) يكون مشابهاً لصندوق الإدخال في فيجوال بيسك وتضمينه في وحدة المكون لذلك ننشئ نموذج جديد ونحفظه في نفس الدليل الذي حفظنا فيه وحدة المكون. ولنحفظ هذا النموذج باسم (IDialog) أما اسم النموذج (Form Name) فليكن (IDlg) مثلاً، ولتكن أبعاده (Height = 154) (Width = 364) ونضيف إلى النموذج مكونات بحيث يصبح النموذج كما في الشكل التالي



وطبعاً يجب تغيير خاصية (BorderStyle) بحيث تصبح (BorderStyle = bsDialog).

ملاحظة : يفضل الاطلاع على نص المصدر (Source) الموجود على موقع www.iss.almoja.net ورابط تحميله هو : www.iss.almoja.net/Components/VBInputBox.zip

الآن وبعد الانتهاء من تصميم النموذج يجب تضمين وحدة هذا النموذج وهي (IDialog) في وحدة المكون عن طريق إضافتها إلى قسم الـــ USES، وبما أن وحدة النموذج تحوي تعريف للنموذج (IDlg) على شكل متحول :

```
var
 IDlg: TIDlg;
```

بالتالي فإن النموذج(IDlg) سيكون متاحاً للمكوّن (InputDlg)، و بما أننا الآن نتحدث عن النموذج فإن قسماً مهماً من واجهة المكوّن هو الطريقة التي سيفتح فيها الصندوق – أو النموذج – ويعيد النتيجة عندما يغلق (Execute method) كما هو الأمر في صناديق الحوار العادية الموجودة في صفحة (Dialogs)، من أجل ذلك نستخدم تابع ذو نتيجة منطقية (Boolean Function) يسمى (Execute) والذي يعيد نتيجة True عندما يضغط المستخدم OK أو موافق و يعيد نتيجة False عندما

إعداد : م . عصام علي

يضغط المستخدم Cancel أو إلغاء الأمر، وبما أن هذا التابع هو method وسيستدعى من الخارج لذلك يجب التصريح عنه في قسم الـــ public كما يلي :

```
public
   Function Execute: Boolean;
  end;
```

أما بالنسبة لكيفية عمل هذا التابع فإننا نكتبه ضمن قسم التطبيق **implementation** كما يلي:

```
Function TInputBoxDlg.Execute: Boolean;
begin
IDlg:=TIdlg.Create(Application);
 try
 result := (IDlg.ShowModal = IDOK);
 finally
 IDlg.Free;
 end;
End;
```

إضافة الخصائص (Add Properties):

إذا أمعنا النظر في الغاية المرجوة من هذا المكوّن نجد أن هناك ثلاث خصائص رئيسية وضرورية لعمل المكوّن وهي : Title (عنوان صندوق الإدخال)، Prompet (الرسالة التي ستطالب المستخدم بإدخال قيمة ما)، IValue (القيمـــة التـــي أدخلها المستخدم) .

معلومة:

عند التصريح عن خاصية نحتاج إلى ثلاثة أشياء :

* اسم الخاصية. * نوع (Type) الخاصية. * الطرق (methods) المستخدمة لقراءة وكتابة هذه الخاصية.

الخصائص المصرح عنها في قسم published تكون متاحة للتعديل من خلال Object Inspector في وقت التصميم و هـــذه القيم المدخلة تحفظ مع المكون في ملف النموذج (Form file)، أما الخصائص المصرح عنها في قسم Public فتكون متاحـــة خلال وقت التشغيل ويمكن قراءتها أو تعديلها من خلال شفرة البرنامج.

ليس هناك أية شروط حول كيفية حفظ بيانات الخاصية ولكن هناك تقاليد متبعة في دلفي مثل :

بيانات الخاصية تخزن في حقول صنف (Class Field).

الحقل المستخدم لتخزين بيانات الخاصية يجب أن يصرح عنه في قسم Private ولا يجب أن يستخدم إلا من داخل المكوّن.

اسم هذا الحقل يتألف من حرف F متبوعاً باسم الخاصية مثلاً خاصية Height يكون الحقل FHeight .

بالنسبة للطرق المستخدمة لقراءة وكتابة بيانات هذه الخاصية فإن أبسط هذه الطرق هي طريقـــة الوصـــول المباشـــر (Direct Access) في هذه الطريقة إن عملية قراءة أو كتابة القيمة المخزنة للخاصية تتم بطريقة مباشرة عبرالحقل المخـــصص لـــذلك وبدون استدعاء لطرق أو إجرائيات إضافية.

في العموم عملية القراءة **(Read)** تكون بطريقة الوصول المباشر أما القراءة **(Write)** فعادة نستخدم طرق (Methods) مـــن أجل تحديث قيمة الخاصية

بالنسبة للخاصتين (Title,Prompet) فإننا نحتاج إلى تعديلهما ربما أثناء وقت التصميم ولـــذلك نـــصرح عنهمـــا فـــي القـــسم **published** أما الخاصية IValue فنصرح عنها في قسم **public** ليصبح قسم Interface كما يلي:

```
interface

uses
```

إعداد : م . عصام علي

```
  SysUtils, Classes,Controls,Forms,windows,IDialog;

type
  TInputBoxDlg = class(TComponent)
  private
    { Private declarations }
  FTitle,FPrompet,FIValue:string;
  Procedure SetIValue (Value:String);

  protected
    { Protected declarations }
  public
    { Public declarations }
  property IValue :String read FIValue Write SetIValue;
  function Execute: Boolean;
  published
    { Published declarations }
  property Title :String read FTitle Write FTitle;
  property Prompet :String read FPrompet Write FPrompet;

  end;

procedure Register;
```

أما في قسم التطبيق **implementation** فنحتاج إلى تعريف الإجرائية SetIValue كما يلي:

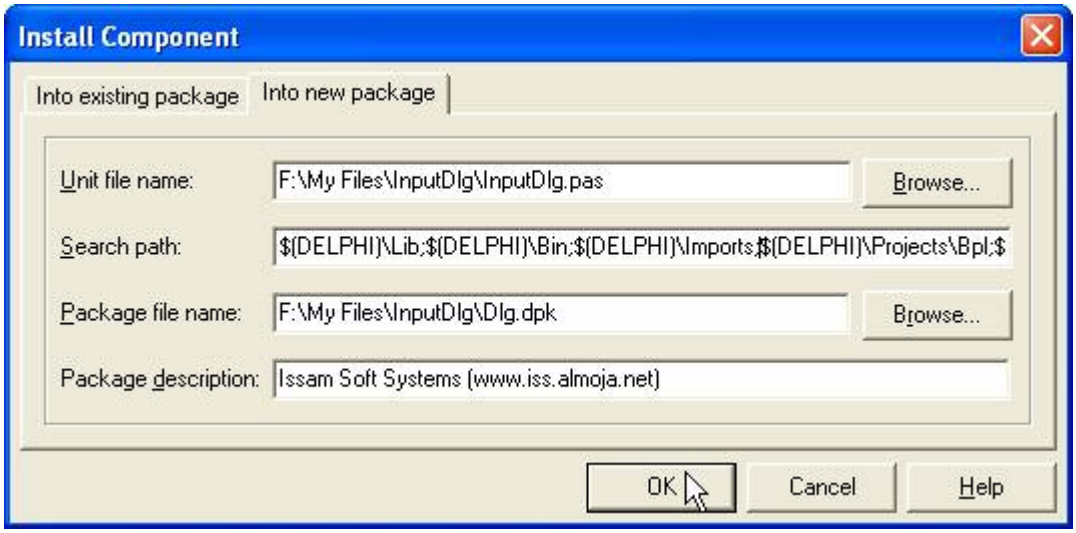
```
Procedure TInputBoxDlg.SetIValue (Value:String);
Begin
  if Value <> FIValue then
  begin
    FIValue := Value;
  end;
End;
```

كما ندخل بعض التعديلات على التابع **Execute** بحيث يصبح على الشكل التالي:

```
function TInputBoxDlg.Execute: Boolean;
begin
IDlg:=TIdlg.Create(Application);
 try
 if Title = '' then Title := 'Input Box';
 IDlg.Caption:=Title;
 IDlg.Prompet.Caption:=Prompet;
 result := (IDlg.ShowModal = IDOK);
 finally
 SetIValue(IDlg.InputValue.Text);
 IDlg.Free;
 end;
End;
```

بهذا يكون المكون جاهزاً للتجريب و نقوم بذلك عن طرق خيار Install Component... في قائمة Component وندخل الخيارات المبينة في الشكل:

إعداد : م . عصام علي



نبدأ تطبيقاً جديداً ونضع على النموذج الرئيسي زراً ونضيف مكوّن InputBoxDlg جديد والذي نجده في صفحة Dialogs ومكوّن Label جديد ونضيف الشيفرة التالية إلى الحدث Button1Click:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
InputBoxDlg1.Title:='InputBox Demo';
InputBoxDlg1.Prompet:='Enter a value between 1 and 3';
if InputBoxDlg1.Execute then
label1.Caption:=InputBoxDlg1.IValue;
end;
```

إنشاء صورة للمكوّن الجديد (Create Component Image):

الخطوة الأخيرة لبناء المكوّن هي صنع صورة Image وربطها مع المكوّن و نقوم بذلك عن طريق برنامج Image Editor المرفق مع دلفي.

نفتح البرنامج و من قائمة File نختار New ثم Component recourse file ثم ضمن هذا الملف ننشئ صورة Bitmap وندخل خصائصها كما يلي:

ثم نسمي هذه الصورة بنفس اسم صنف المكون وفي حالتنا هذه (TINPUTBOXDLG) و لاحظ أن الأحرف المستخدمة يجب أن تكون كبيرة، ثم نرسم صورة تعبر عن هذا المكون ونحفظ ملف الـ Recourse في نفس دليل المكون ونسميه مثلاً InputBoxDlg.dcr ، الآن نعود إلى وحدة المكون ونضيف السطر التالي قبل قسم **implementation**:

```
{$R InputBoxDlg.dcr}
implementation
```

إعداد : م . عصام علي

الآن نعيد ترجمة الحزمة التي تحتوي المكوّن الجديد وهي هنا Dlg.dpk ونعيد تثبيت المكون لتظهر الصورة التي رسمتها والتي تعبر عن هذا المكون.

تأليف وإعداد المهندس: عصام علي

جميع الحقوق محفوظة

www.iss.almoja.net