

# الفصل الأول

## كتابة أول برنامج لك في فيجول بيسك

سنكتب في هذا الفصل، أول برنامج لك في فيجول بيسك. تمر مرحلة كتابة البرنامج في فيجول بيسك بخطوتين:

- خطوة التصميم المرئي للبرنامج.
- سندعوها عبر الكتاب بالتمثيل المرئي Visual Implementation للنموذج.
- خطوة كتابة نص البرنامج.

يُصمم المستخدم خلال الخطوة الأولى، البرنامج باستخدام الأدوات التي تأتي مع برمجية فيجول بيسك. يمكنك هذه الأدوات من تصميم البرنامج باستخدام الفأرة ولوحة المفاتيح.

لا يلزمنا خلال مرحلة البرمجة المرئية (أي التصميم المرئي) كتابة أي نص برمجي!، وكل ما عليك معرفته هو كيف تشغل وتستخدم الأدوات البرمجية التي تأتي مع فيجول بيسك. وسوف تلاحظ أن عملية التصميم المرئي فيها قدر كبير من المتعة وتعتمد بكثرة على النقر بالفأرة. يركز هذا الفصل على معرفة كيفية استخدام أدوات التصميم المرئي في فيجول بيسك.

أما في خطوة كتابة نص البرنامج، فيستخدم محرر نصوص لكتابة البرنامج. وتتألف البرامج من عبارات مكتوبة بلغة البرمجة فيجول بيسك. تتشابه عملية كتابة نصوص البرامج في فيجول بيسك مع كتابة البرامج في اللغات الأخرى. إلا أن كتابة البرامج في فيجول بيسك أسهل بكثير من كتابتها باللغات الأخرى.

## حول هذا الفصل

ستلاحظ إذا استعرضت باقي فصول الكتاب، أن هذا الفصل ليس نموذجياً، فهو يركز على ناحية البرمجة المرئية Visual Programming للغة فيجول بيسك، ولهذا فهو يركز على كيفية استخدام الأدوات البرمجية للغة (عناصر التحكم). بينما تتولى فصول الكتاب المتبقية تعليمك، كيفية كتابة نص البرامج في فيجول بيسك.

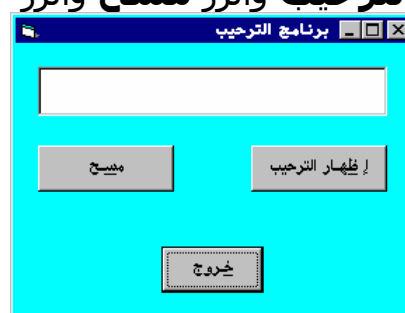
## إنشاء دليل حفظ الملفات

قبل البدء بعملية كتابة أول برنامج لك في لغة فيجول بيسك، سننشئ دليلاً يحوي على ملفات العمل المنجز. وسنفترض عبر هذا الفصل، أن الدليل موجود لديك على القرص الصلب، وسنطلب منك فقط حفظ الملفات فيه، لهذا أنشئ الدليل التالي C:\VB5Prg\Ch01.

## برنامج الترحيب

سنكتب برنامجاً يدعى برنامج الترحيب. وقبل كتابة برنامج الترحيب بنفسك دعنا ندرس أولاً مواصفاته، وبهذه الطريقة سنتمكن من فهم ما يفترض أن ينجزه هذا البرنامج بشكل أفضل. يظهر الإطار المبين في الشكل 1-1، عند بدء تشغيل برنامج الترحيب وكما تلاحظ يحوي إطار البرنامج على ثلاثة أزرار أوامر (الزر إظهار الترحيب والزر مسح والزر خروج) ومربع نص فارغ.

الشكل 1-1  
برنامج الترحيب.



تظهر الرسالة **مرحباً بكم** ضمن مربع النص، عند النقر على الزر **إظهار الترحيب**.

الشكل 2-1  
إظهار مرحباً

بكم

في مربع النص.



يمحو برنامج الترحيب محتوى مربع النص، عند النقر بالفأرة على الزر **مسح**.

ينتهي تنفيذ برنامج الترحيب عند النقر على الزر **خروج**.

## إنشاء مشروع جديد

الآن وقد علمت ما ينجزه برنامج الترحيب، نستطيع الشروع بكتابته.

### ملاحظة

برنامج الترحيب عبارة عن برنامج بالغ البساطة، لكن مع ذلك يتوجب عليك كتابته بنفسك، لأنه يمثل برنامجاً نموذجياً في فيجول بيسك. وفي الواقع، حال تعلمك كيفية كتابة برنامج الترحيب بنفسك، تستطيع فهم ما هي لغة فيجول بيسك! طبعاً هنالك كم هائل من المعلومات الأخرى في فيجول بيسك لا بد لك من معرفتها، إلا أن كتابة هذا البرنامج بنفسك يعني أنك تعرفت على أساسيات فيجول بيسك.

أولى الأوليات التي يجب عليك إنجازها، هي إنشاء مشروع جديد New Project لبرنامج الترحيب باتباع الخطوات التالية:

□ شغل فيجول بيسك. إذا رأيت مربع الحوار المبين في الشكل 1-1-3، فأغلق هذا المربع بنقر الزر **إلغاء الأمر**.

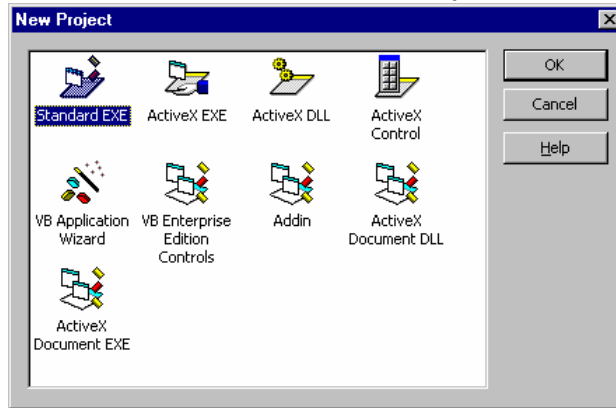
□ اختر البند **New Project** من القائمة **File** لفيجول بيسك. يستجيب فيجول بيسك بإظهار مربع الحوار **New Project** المبين في الشكل 1-4.

الشكل 3-1  
مربع الحوار New Project  
الذي قد يظهر عند  
تشغيل فيجول بيسك.



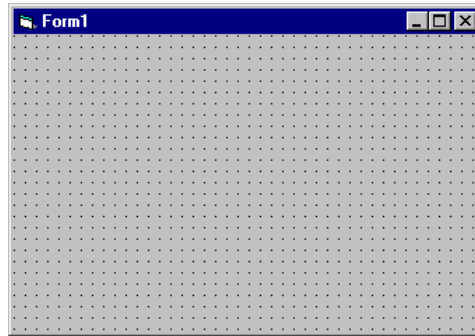
كما يبدو من الشكل 4-1. يمكنك مربع الحوار New Project من إنشاء  
أنواع مختلفة من المشاريع (Project). لكننا في الوقت الراهن، نرغب  
بإنشاء تطبيق تنفيذي قياسي (Standard EXE).

الشكل 4-1  
مربع الحوار New Project  
الذي يظهره فيجول  
بيسك بعد اختيار New  
Project من القائمة  
.File



لهذا أخبر فيجول بيسك بذلك باتباع الخطوة التالية:  
□ انقر الرمز Standard EXE الذي يظهر ضمن مربع الحوار New  
Project ثم انقر الزر **OK** لمربع الحوار New Project.  
يظهر عند ذلك إطار خال يدعى النموذج Form1 (انظر الشكل 5-1).  
سنستخدم أدوات لغة فيجول بيسك المتنوعة حتى يصبح النموذج  
الخالى،  
في الشكل 1-1.

### الشكل 5-1 النموذج Form1 الفارغ.



### حفظ المشروع الجديد

رغم أننا لم نجر أي تعديلات بعد على النموذج الفارغ، لكن يتوجب عليك حفظ المشروع في هذه المرحلة المبكرة من التصميم. يؤدي حفظ المشروع إلى تخزين ملفين:

- ملف المشروع Project File، ويمتلك الامتداد .vbp. يحتوي هذا الملف على المعلومات التي يستخدمها فيجول بيسك لبناء المشروع.

- ملف النموذج Form File، ويمتلك الامتداد .frm. ويحتوي على معلومات تتعلق بالنموذج نفسه.

استخدم الآن الخطوات التالية لحفظ الملفين: Hello.vbp (ملف المشروع)، و Hello.frm (ملف النموذج).

- تحقق من تمرکز الإضاءة عند عنوان الإطار Form1، ثم اختر **Save Form1 As** من القائمة **File** لفيجول بيسك. (يعني تمرکز الإضاءة عند نموذج ما، أنه تم اختياره).

يستجيب فيجول بيسك بإظهار مربع الحوار *Save File As*.

- استخدم مربع الحوار *Save File As* لاختيار الدليل C:\VB5Prg\Ch01 من أجل حفظ الملف فيه. بدل الاسم الافتراضي للنموذج من Hello.frm إلى Form1.frm

(انظر الشكل 6-1).

- انقر الزر **حفظ** في مربع الحوار *Save File As*.

يستجيب فيجول بيسك بحفظ النموذج باسم *Hello.frm* في الدليل *C:\VB5Prg\Ch01*.

### ملاحظة

لا تستخدم الاسم الافتراضي الذي يقدمه فيجول بيسك عند حفظ نموذج ما. بل احفظ النموذج باسم مناسب للتطبيق الذي تصممه.

لاحظ مثلاً، أننا استخدمنا الاسم Hello.frm كاسم لنموذج برنامج الترحيب.

الشكل 6-1  
حفظ النموذج  
بالاسم Hello.frm.



والآن، احفظ ملف المشروع:

- اختر **Save Project As** من القائمة **File** لفيجول بيسك.
- يستجيب فيجول بيسك بإظهار مربع الحوار *Save Project As*.
- الاسم الافتراضي الذي يقدمه فيجول بيسك للمشروع هو Project1.vbp.
- لكن لا بد من تغيير الاسم الافتراضي إلى اسم يتناسب مع التطبيق الذي تعمل على تطويره.
- استخدم مربع الحوار *Save Project As* لحفظ المشروع بالاسم Hello.vbp في الدليل C:\VB5Prg\Ch01.

#### ملاحظة

لا تستخدم الاسم الافتراضي الذي يعطيه فيجول بيسك للمشروع عند حفظ ملف المشروع. ولكن بدلاً من ذلك أطلق على المشروع اسماً يتناسب مع التطبيق الذي تصممه. لاحظ مثلاً أننا أطلقنا على مشروع برنامج الترحيب تسمية Hello.vbp. نكون حتى هذه اللحظة قد أنهينا حفظ الملفين Hello.vbp (ملف المشروع) و Hello.frm (ملف النموذج).

### فحص إطار المشروع Project Window

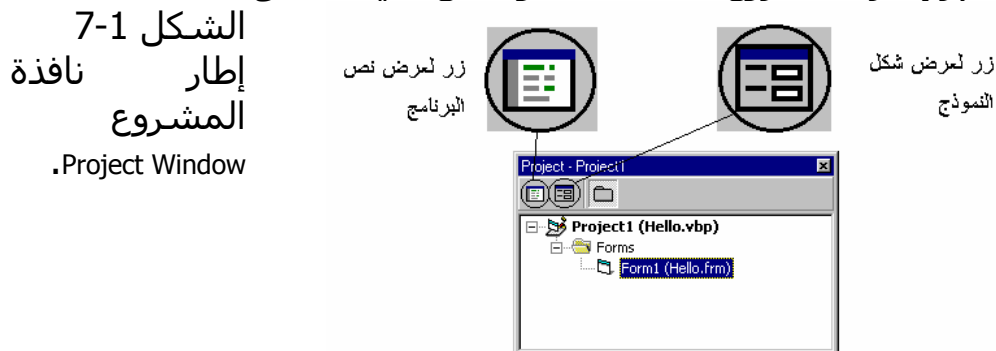
حتى هذه النقطة، يدعى المشروع بالاسم Hello.vbp ويتألف من ملف نموذج واحد هو الملف Hello.frm. سنمر عبر الفصول القادمة على مشاريع تحوي أكثر من ملف نموذج.

يعتبر إطار المشروع Project Window من الأدوات الهامة التي يقدمها فيجول بيسك، ويمكنك من رؤية الملفات المختلفة الموجودة في المشروع. (ستشعر بقيمة هذه الميزة بشكل أفضل كلما زاد تعقيد المشروع).

اتبع الخطوات التالية لفهم محتويات إطار المشروع Project Explorer:

□ اختر **Project Explorer** من القائمة **View** لفيجول بيسك.

يظهر إطار المشروع حسب ما هو مبين في الشكل 7-1.



ملف المشروع، حسب ما يظهر من إطار المشروع هو Hello.vbp. وهو يحتوي على نموذج وحيد باسم Hello.frm. يعتبر المشروع الحالي بسيط جداً، بحيث لا تغدو أهمية إطار المشروع واضحة للعيان. لكن مع تزايد تعقيد المشاريع التي ستكتبها مستقبلاً، سوف تلاحظ مدى أهمية هذا الإطار.

## تغيير الخاصية Caption للنموذج (تغيير عنوان النموذج)

اتفقنا أن النموذج الفارغ الذي أنشأه فيجول بيسك يحمل العنوان Form1 (انظر الشكل 5-1). هذا العنوان يمثل العنوان الافتراضي الذي يعطيه فيجول بيسك للنموذج الفارغ عند إنشائه. تستطيع ترك هذا العنوان على حاله، ولكننا نفضل أن نطلق عنواناً على النموذج يكون مناسباً لموضوعه. فمثلاً حسب الشكل 1-1، يحمل النموذج عنوان برنامج الترحيب، وفي هذا دلالة على أن البرنامج هو برنامج ترحيب.

### ملاحظة

النموذج هو إطار (Window) بنفس الوقت. فمثلاً، النموذج Form1 المبين في الشكل 5-1 يدعى نموذج ويدعى إطار (Window) بذات الوقت. يطلق مصطلح النموذج، على النافذة التي أنشأتها في مرحلة التصميم،

ويستخدم مصطلح الإطار (أو نافذة)، عند تنفيذ البرنامج. بكلام آخر، عند ذكر مصطلح النموذج، يكون المقصود مرحلة التصميم، وعند ذكر مصطلح الإطار أو النافذة، يكون المقصود مرحلة التنفيذ.

نبين لك الآن طريقة تغيير عنوان النموذج الفارغ لبرنامج الترحيب:

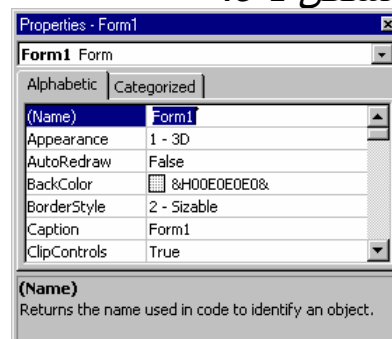
- تحقق من اختيار النموذج الفارغ. تستطيع التأكد بسهولة من اختيار نموذج، بتفحص شريط عنوانه. فإذا كان شريط العنوان (Caption) مضاءً، فهذا يعني أن النموذج تم اختياره. أما إذا لم يكن قد تم اختيار النموذج، فيكفي النقر النموذج في أي مكان على سطحه لاختياره.

أو تستطيع بدلاً من ذلك اللجوء إلى إطار المشروع Project Window (اختر Project Explorer من القائمة View لفيجول بيسك)، ثم اختيار العنصر Hello.frm بنقر الفأرة عليه، ثم نقر الرمز View Object الذي يظهر أعلى إطار المشروع. (انظر الشكل 7-1. يظهر الرمز View Object أعلى نافذة المشروع وثاني رمز من جهة اليمين).

بعد اختيار النموذج، اختر إطار الخصائص Properties Window من القائمة View لفيجول بيسك.

يستجيب فيجول بيسك بإظهار إطار الخصائص (Properties Window)، الشكل 8-1.

الشكل 8-1  
إطار الخصائص  
Properties Window.



### ملاحظة

يمكنك في فيجول بيسك نقل شتى الإطارات إلى أي موقع ضمن سطح مكتب فيجول بيسك وذلك بسحب عناوين هذه الإطارات بواسطة الفأرة.

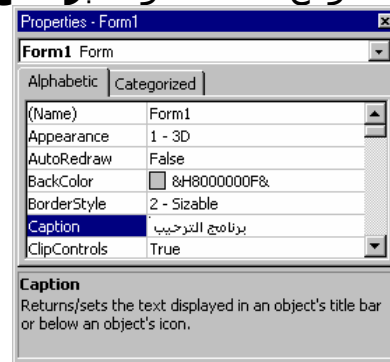
قد يتغير شكل الإطارات قليلاً تبعاً للموقع الذي تأخذه على سطح المكتب. فمثلاً قد يتغير شكل إطار الخصائص Properties Window إلى حد ما، عما هو مبين في الشكل 8-1، بحسب الموقع الذي يحتله على سطح المكتب، لكن مهما كان موقعه، تستطيع التعرف



عليه، بسبب احتوائه على كلمة الخصائص Properties في عنوانه.

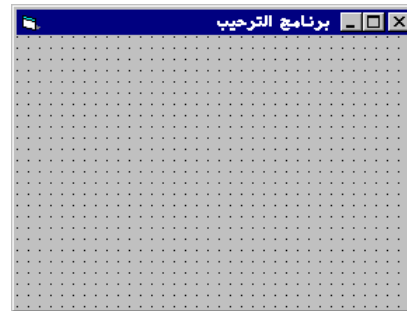
- انقر على الخلية التي تظهر يمين اسم الخاصية Caption (خلية العنوان) في إطار الخصائص (Properties Window).
- الآن ستجد، أن الخلية الواقعة يمين الخلية Caption تملك النص Form1.
- استخدم المفتاح Delete ومفاتيح الأسهم على لوحة المفاتيح، لحذف النص Form1، واستبدله بكتابة النص **برنامج الترحيب**.
- يفترض أن يبدو إطار الخصائص كما في الشكل 9-1.
- تهانينا! لقد أنهيت للتو عملية تبديل الخاصية Caption (العنوان) للنموذج.
- ألق نظرة على النموذج الفارغ (انظر الشكل 10-1). يحمل هذا النموذج الآن العنوان **برنامج الترحيب**.

الشكل 9-1  
تبديل الخاصية  
Caption للنموذج.



الشكل 10-1  
النموذج frmHello وهو  
يحمل الآن عنواناً

جديداً.



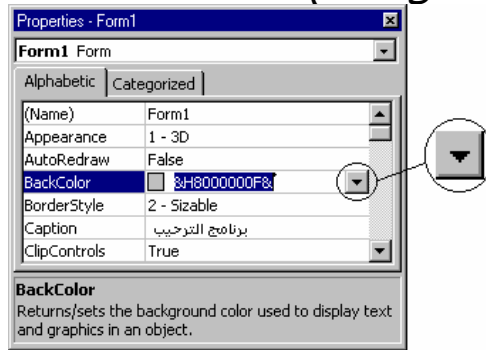
## ما هي الخاصية؟!

الخاصية Caption ما هي إلا إحدى خصائص النموذج، فكما تشاهد من إطار الخصائص Properties Window، فإن النموذج يمتلك الكثير من الخصائص الأخرى. ولفهم معنى الخاصية لا بد لك من فهم كيفية تعامل فيجول بيسك مع الكائنات Objects مثل النماذج Forms وأزرار الأوامر Command Buttons وأشرطة التمرير Scroll bars ومربعات الاختيار Check Boxes . . . الخ.

تُعرّف خصائص الكائن (Object) كيف يبدو هذا الكائن وكيف يتصرف. فمثلاً النموذج عبارة عن كائن. تحدد الخاصية Caption للنموذج، النص الذي يظهر في شريط عنوانه. خذ مثلاً الخاصية BackColor للنموذج، تحدد هذه الخاصية لون خلفية النموذج. اتبع الخطوات التالية لتغيير الخاصية BackColor للنموذج:

- تحقق من اختيار النموذج. (انقر أي مكان من النموذج لاختياره).
- اختر **Properties Window** من القائمة **View** لإظهار إطار الخصائص.
- انقر على الخلية الواقعة يمين الخلية BackColor في إطار الخصائص.
- يضع فيجول بيسك عند نقر هذه الخلية، رمز سهم نازل فيها، (انظر الشكل 11-1).

الشكل 11-1  
الخاصية BackColor.

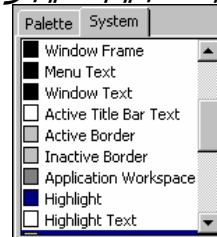


- انقر رمز السهم النازل الذي يظهر في الخلية.

يستجيب فيجول بيسك بإظهار مربع الحوار المبين في الشكل 12-1.

الشكل 12-1  
مربع الحوار الذي يظهر عند  
نقر رمز السهم النازل

الموجود



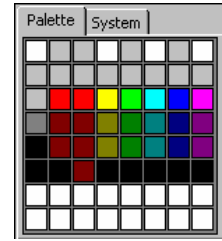
جانب الخاصية BackColor.

لاحظ أن مربع الحوار المبين في الشكل 12-1 يمتلك صفحتين: الصفحة Palette، والصفحة System. الصفحة التي تظهر وفق الشكل 12-1 هي الصفحة System.

- انقر على الصفحة Palette في مربع الحوار المبين في الشكل 12-1.

يستجيب فيجول بيسك بإظهار صفحة الألوان Palette (انظر الشكل 13-1).

### الشكل 13-1 الصفحة Palette صفحة الألوان.



□ اختر اللون الذي تحبذه بنقره في الصفحة Palette. لنفترض الآن أن اللون الذي اخترته هو اللون الذي يظهر في المربع الواقع عند العمود الثالث والسطر الثالث من جهة الأعلى. (أو اختر أي لون تفضل).

#### ملاحظة

تفحص الخلية التي تقع يمين اسم الخاصية في نافذة الخصائص والتي ترغب بتعيينها. فإذا كانت تلك الخلية تحمل رمز سهم نازل داخلها، أو زر يحوي ثلاث نقاط متجاورة، انقر على السهم أو الزر، فيظهر إطار آخر أو لائحة، يمكنك من اختيار قيمة ما بواسطة الفأرة للخاصية المحددة. جرب عدة ألوان بتكرار العملية، إلى أن تشعر بالرضا عن اللون الذي تختاره.

### تبديل الخاصية Name للنموذج (اسم النموذج البرمجي)

يجب أن يمتلك كل كائن (Object) في فيجول بيسك اسماً، يتحدد ذلك الاسم بواسطة الخاصية Name لذلك الكائن. فمثلاً، عندما أنشأنا النموذج الجديد لبرنامج الترحيب، أطلق فيجول بيسك من تلقاء نفسه الاسم Form1 على النموذج (أي أسند الاسم Form1 إلى الخاصية Name لنموذج برنامج الترحيب).

#### ملاحظة

لا تخط بين الخاصية Caption، والخاصية Name للنموذج. تستخدم الخاصية Caption لإظهار عنوان ما، في شريط عنوان النموذج. أما الخاصية Name فتستخدم لإسناد اسم برمجي خاص بالنموذج نفسه. مع أن فيجول بيسك يفترض القيمة Form1 لكلتا الخاصيتين عند إنشاء نموذج جديد فارغ.

غير الآن الخاصية Name للنموذج:

□ انقر على النموذج في أي مكان منه لاختياره.

- اختر **Properties Window** من القائمة **View**.  
يستجيب فيجول بيسك بإظهار إطار خصائص النموذج *Form1*.  
يملك إطار الخصائص **Properties** صفحتين هما: الصفحة **Alphabetic** والصفحة **Categorized** (انظر الشكل 1-8). عند اختيار الصفحة **Alphabetic**، ترتب الخصائص أبجدياً (باستثناء أهم خاصية وهي **Name** التي تظهر أولاً). بينما تظهر الخصائص مصنفة حسب مواضيعها، عند اختيار الصفحة **Categorized**.
- اختر الصفحة **Alphabetic** لإطار الخصائص.  
تظهر الخاصية **Name** في مقدمة لائحة الخصائص.
- انقر الخلية التي تظهر يمين الخاصية **Name**، يمكنك فيجول بيسك من تعديل الخاصية **Name**.
- استبدل الاسم الافتراضي **Form1** بالاسم **frmHello**.  
غيرنا في الخطوة السابقة الخاصية **Name** للنموذج من **Form1** إلى **frmHello**. تشير الأحرف الثلاث الأولى من قيمة الخاصية **Name** للكائنات **Objects** إلى نوع الكائن.  
وهكذا فالأحرف الثلاث الأولى من الخاصية **Name** لنموذج ما، هي **frm**، كما في مثالنا الحالي **frmHello**.

#### ملاحظة

غير الأسماء الافتراضية للكائنات بحيث تعكس أسماؤها ووظائفها في البرنامج. فمثلاً **frmHello** عبارة عن اسم النموذج الذي يستخدم من قبل برنامج الترحيب. يؤدي بدء اسم النموذج بالأحرف الثلاث **frm** إلى تسهيل فهم البرنامج وتصحيحه.  
ألق نظرة على الاسم **frmHello**، بما أنه يبدأ بالأحرف **frm**. تستطيع بسهولة (أنت أو من يقرأ الاسم) أن تعلم مباشرة أن نوع الكائن **frmHello** عبارة عن نموذج. لا يعتبر هذا الأمر من متطلبات البرمجة في لغة فيجول بيسك، لكنه كما قلنا يسهل قراءة وفهم البرنامج.

### تبدل الخاصية **RightToLeft** للنموذج (تعريب النموذج)

يمكنك فيجول بيسك، من إنشاء برامج عربية المظهر والمضمون، بشرط تطوير برنامجك في نظام تشغيل يدعم اللغة العربية، مثل Windows95 العربي أو WinNT الداعم للغة العربية.

وما عليك سوى تغيير قيمة الخاصية RightToLeft من القيمة False إلى القيمة True، لأي كائن من كائنات فيجول بيسك، حتى يظهر الكائن بشكل مقبول للمستخدم العربي.

#### ملاحظة

للحصول على معلومات كاملة، عن موضوع إنشاء التطبيقات العربية في فيجول بيسك، اقرأ الفصل الثاني والعشرين (إنشاء تطبيقات عربية السمة مع فيجول بيسك 5). وكل ما يهمنا معرفته الآن، أن الخاصية RightToLeft = True تعني تعريب الكائن.

غير الآن الخاصية RightToLeft للنموذج:

- انقر على النموذج في أي مكان منه لاختياره.
- اختر **Properties Window** من القائمة **View**.
- يستجيب فيجول بيسك بإظهار إطار خصائص النموذج *Form1*.
- انقر خلية الخاصية RightToLeft نقرًا مزدوجًا، لتغيير قيمتها من القيمة False إلى القيمة True.
- يستجيب فيجول بيسك، بتعريب النموذج بأن يظهر عنوانه على يمين المستخدم، بدلاً من يساره.
- أصبح النموذج الآن، نموذجاً عربياً. وعند إضافة أي عنصر تحكم (أداة من أدوات فيجول بيسك)، سيقوم فيجول بيسك بإسناد القيمة True للخاصية RightToLeft للعنصر الجديد آلياً.

#### حفظ العمل المنجز

لم تنته بعد من النموذج (تذكر أن النموذج سيبدو عند انتهائه كما في الشكل 1-1)، لكن رغم ذلك يفضل حفظ العمل الذي أنجزته حتى هذه اللحظة، حتى لا تضطر إلى إعادة العمل مرة ثانية، إذا انهار الحاسب لديك لسبب ما. لهذا اتبع الخطوات التالية لحفظ العمل:

- اختر **Save Project** من القائمة **File**.
- يستجيب فيجول بيسك بحفظ كل التغييرات المنجزة على ملف المشروع أو أي من الملفات التابعة للمشروع (مثال ذلك، الملف *Hello.frm*).

#### إضافة الزر خروج إلى النموذج frmHello

حسب ما يظهر من الشكل 1-1، فالنموذج المكتمل سيحتوي ثلاثة أزرار أوامر داخله، وهي: **إظهار الترحيب و مسح و خروج**.

لوضع زر أمر ما ضمن النموذج، لا بد لك من تحديده أولاً من مربع الأدوات.

### إطار مربع الأدوات Toolbox Window

يحتوي إطار مربع الأدوات، رموز جميع الكائنات المتاحة لمشروعك الحالي. ومهمتك هي التقاط الكائن من مربع الأدوات، ووضعه على النموذج.

□ أظهر إطار مربع الأدوات، باختيار **Toolbox** من القائمة **View** لفيجول بيسك.

يستجيب فيجول بيسك بإظهار مربع الأدوات (الشكل 14-1).

الشكل 14-1  
إطار مربع الأدوات  
Toolbox.



### ملاحظة

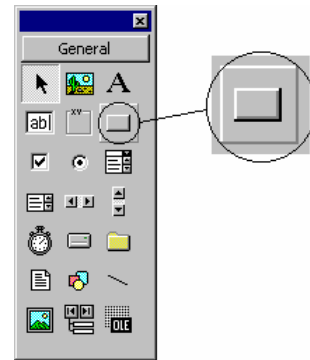
قد يختلف شكل إطار مربع الأدوات قليلاً عما هو عليه في الشكل 1-14، وذلك تبعاً للموقع الذي يأخذه على سطح مكتب فيجول بيسك (أي حسب المكان الذي تضعه فيه). كما أن إطار مربع الأدوات، قد يحوي رموزاً أكثر أو أقل، تبعاً لنوع نسخة فيجول بيسك 5 التي تملكها، وتبعاً لإعدادات فيجول بيسك 5.

يبين الشكل 1-15 رمز زر الأمر مكبراً، وهو طبعاً أحد الرموز التي تظهر في مربع الأدوات. تستطيع التعرف بسهولة على مختلف الرموز في مربع الأدوات بوضع مؤشر الفأرة فوق أي رمز بدون النقر عليه، ليظهر مستطيل أصفر يحمل بداخله اسم الكائن الذي يمثله هذا الرمز.

فمثلاً، عند وضع مؤشر الفأرة فوق رمز زر الأمر دون النقر عليه، يظهر مستطيل أصفر يحمل الرسالة `CommandButton` داخله.

الشكل 15-1  
رمز زر الأمر  
CommandButton  
في إطار مربع

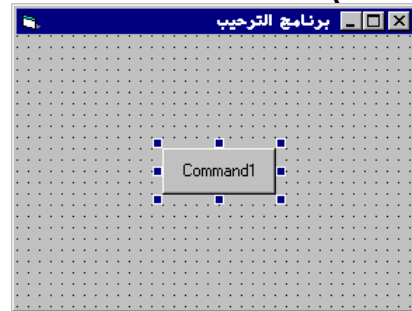
الأدوات.



### وضع الزر خروج على النموذج

اتبع الخطوات التالية لوضع زر أمر على النموذج:  
☐ انقر نقرًا مزدوجًا على رمز زر الأمر في مربع الأدوات. (انظر الشكل 15-1 للتعرف على شكل زر الرمز).  
 يستجيب فيجول بيسك بوضع زر أمر في مركز النموذج (انظر الشكل 16-1).

الشكل 16-1  
النموذج مع  
زر الأمر بداخله.

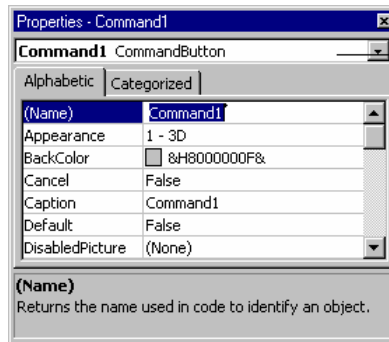


يتولى فيجول بيسك تعيين مختلف القيم الافتراضية لخصائص زر الأمر CommandButton الذي وضعته على النموذج. فمثلاً العنوان الافتراضي (Caption) لذلك الزر هو Command1.

### تغيير الخاصية Name للزر خروج (تغيير الاسم البرمجي)

ستغير اسم زر الأمر من Command1 إلى cmdExit باعتبار أن هذا الزر سيعمل عمل زر الإنهاء **خروج**:  
☐ اختر **Properties Window** من القائمة **View** لفيجول بيسك.  
 يستجيب فيجول بيسك بإظهار إطار الخصائص.  
☐ تحقق أن مربع السرد عند قمة إطار الخصائص يظهر البند التالي:  
 Command1 CommandButton. (انظر الشكل 17-1 لتحديد موقع مربع السرد).

الشكل 17-1  
مربع السرد عند قمة  
إطار  
الخصائص  
ويظهر  
البند  
التالي:  
**Command1**  
CommandButton.



البند المختار حالياً هو  
CommandButton

### ملاحظة

يملك النموذج الآن كائنين: النموذج frmHello وزر الأمر Command1. يظهر إطار الخصائص، خصائص الكائن الذي يظهر اسمه حالياً في مربع السرد الواقع عند قمة إطار الخصائص. للانتقال بين الكائنات، انقر رمز السهم النازل، الموجود يمين مربع السرد واختار الكائن المطلوب من اللائحة المنسدلة للأسفل.

غير الخاصية Name للزر Command1 إلى cmdExit. لاحظ كيف وضعنا الحروف الثلاثة cmd قبل كلمة Exit، وذلك كما اتفقنا سابقاً، للدلالة على نوع الكائن ووظيفته في آن واحد (تعتبر وظيفة زر الأمر في مثالنا هذا، تنفيذ أمر الخروج من البرنامج)، هذا العمل ليس من متطلبات لغة فيجول بيسك، ولكن لتسهيل قراءة وفهم البرنامج، فعندما نشاهد الاسم cmdExit، نستطيع القول مباشرة، أن هذا الاسم يخص زر أمر وأن وظيفته هي إنهاء البرنامج.

### تغيير الخاصية Caption للزر خروج (تغيير العنوان)

العنوان الافتراضي الذي يعطيه فيجول بيسك لزر الأمر هو Command1. وبما أن وظيفة زر الأمر هذا الخروج من البرنامج، فأنسب عنوان له هو خروج:

غير الخاصية Caption لزر الأمر cmdExit من Command1 إلى خروج.

للحصول على الرمز (&)، اضغط المفاتيح Shift+7 على لوحة المفاتيح. عند استخدام الرمز & (مثل استخدامه قبل الحرف خ في الخطوة السابقة)، يتسبب بقيام فيجول بيسك بوضع خط تحت الحرف الذي يليه (الحرف الذي يأتي بعد الرمز &)، ولاحظ وجود خط تحت الحرف خ في عنوان الزر خروج. انظر الشكل 1-1.



والآن، عند تنفيذ البرنامج، يؤدي الضغط على المفاتيح (خ+Alt) من لوحة المفاتيح إلى نفس تأثير النقر على الزر **خروج**.

#### ملاحظة

ينصح دائماً باستخدام الرمز & قبل أحد حروف عنوان زر ما. يتسبب هذا الرمز & بظهور خط تحت الحرف الذي يليه مباشرة، وأثناء التنفيذ يصبح المستخدم قادراً، إما على نقر الزر بالفأرة أو ضغط المفتاح Alt إضافة لضغط الحرف المحدد (ضغط المفتاح Alt والمفتاح الذي تحته خط). والذي يمثل حرف وصول سريع.

### تبديل موقع الزر خروج

كما ترى من الشكل 1-1، يجب أن يقع الزر **خروج** بقرب الحافة السفلى من النموذج. اسحب الزر **خروج** إلى الموقع المطلوب بضغط زر الفأرة الأيسر على أي مكان من الزر، ثم سحب وتحريك مؤشر الفأرة دون تحرير الزر الأيسر للفأرة، ثم تحرير زر الفأرة عند الوصول للموقع المناسب.

### تبديل خصائص الخط (Font) للزر خروج

كما تلاحظ من الشكل 1-1، يختلف نوع الخط المستخدم في الزر **خروج** عن الخط الافتراضي الذي استخدمه فيجول بيسك لعنوان الزر، الذي وضعته على النموذج.

اتبع الخطوات التالية لتبديل نوع الخط للزر **خروج**:

□ انقر على الخلية الواقعة يمين الخاصية Font التابعة للزر cmdExit. كما تلاحظ تحوي الخلية على مربع داخله ثلاث نقاط، يؤدي نقره إلى فتح مربع حوار تحديد الخطوط، يمكنك من اختيار قيم معينة بواسطة.

□ انقر على الزر ذي الثلاث نقط الموجود جانب الخاصية Font للزر cmdExit.

يستجيب فيجول بيسك بإظهار مربع الحوار Font.

□ غير نوع الخط إلى System.

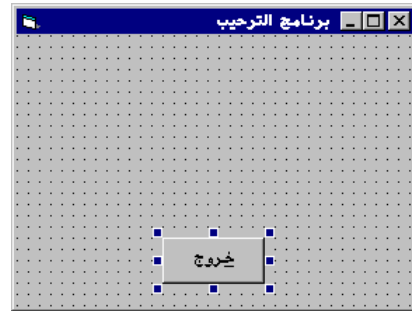
□ غير حجم الخط إلى 10.

□ انقر على الزر **Ok** في مربع الحوار Font.

□ ألق نظرة على الزر **خروج** الآن، ولاحظ أن العنوان مكتوب بالخط الذي حددته مسبقاً بواسطة مربع الحوار Font.

يشبه النموذج الذي نبينه حتى الآن، ذاك المبين في الشكل 1-18.

الشكل 1-18  
النموذج يحتوي الزر  
خروج على سطحه.



#### ملاحظة

لعل إحدى المزايا الرئيسية لفيجول بيسك، أنه يمكنك من رؤية نتائج البرمجة المرئية لحظياً. تمرّن دوماً وجرب خيارات مختلفة (كأن تجرب أنواع خطوط مختلفة، وأحجام وألوان مختلفة) حتى تقنع بالنتائج.

احفظ العمل الذي أنجزته:

□ اختر **Save Project** من القائمة **File** لفيجول بيسك.

**إضافة أزرار أخرى إلى النموذج frmHello**  
لعل الوقت قد حان، لإضافة زرّين جديدين إلى النموذج؛ الزر **إظهار الترحيب** والزر **مسح**.

#### وضع الزرين على النموذج

يحتوي النموذج حسب ما يفترضه الشكل 1-1، زرّين أمر آخرين:  
الزر **إظهار الترحيب** والزر **مسح**، سنضع هذين الزرين على النموذج:

□ أضف الزر **إظهار الترحيب** إلى النموذج بالنقر المزدوج على رمز زر

Command Button ضمن مربع الأدوات. اسحب زر الأمر الجديد إلى اليمين (سيأخذ هذا الزر دور الزر **إظهار الترحيب**).

□ انقر نقراً مزدوجاً رمز زر الأمر في مربع الأدوات مجدداً، ثم اسحب زر الأمر الجديد إلى اليسار. (يأخذ هذا الزر دور الزر **مسح**).

#### تغيير حجم الأزرار

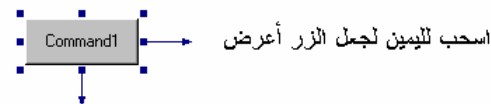
الأحجام الافتراضية للزّرين **إظهار الترحيب** و **مسح** أصغر مما هي عليه في الشكل 1-1.

□ كبر الزرين الجديدين اللذين وضعتهما للتو في النموذج، يتم تكبير أو تصغير كائن باختياره أولاً. يحيط فيجول بيسك الكائن حال اختياره

بمستطيل يتألف من ثماني مربعات سوداء، تدعى بالمقابض. اسحب أحد المقابض حتى يصل الكائن إلى الحجم المطلوب، فمثلاً لتغيير حجم الكائن أفقياً اسحب أحد المقابض أفقياً، أما لتغيير حجم الكائن عمودياً فاسحب أحد مقابضه عمودياً (انظر الشكل 1-19).

الشكل 1-19

تغيير حجم كائن  
بسحب مقابضه.



اسحب للأسفل لجعل الزر أصغر

## تبديل خاصية الاسم Name للزرين السابقين

الاسمان الافتراضيان الذين يطلقهما فيجول بيسك على الزرين الذين وضعتهما للتو في النموذج هي Command1 و Command2، ولكننا سنسبب لهما باسمين يناسبان عملهما أكثر:

- بَدِّل الخاصية Name للزر الأمر اليميني إلى cmdHello.
- بَدِّل الخاصية Name للزر الأمر اليساري إلى cmdClear.

## تبديل عنواني الزرين السابقين

حسب ما يوضحه الشكل 1-1، يجب أن يكون عنوان الزر الأيمن هو **إظهار الترحيب**، وأن يكون عنوان الزر الأيسر هو **مسح**.

- بَدِّل الخاصية Caption للزر الأيمن إلى إظهار الترحيب.
- بَدِّل الخاصية Caption للزر الأيسر إلى مسح.

## تبديل نوع الخط المستخدم في الزرين

- بَدِّل الخاصية Font للزر cmdHello إلى System واجعل حجم الخط مساوياً إلى 10.
- بَدِّل الخاصية Font للزر cmdClear إلى System واجعل حجم الخط مساوياً إلى 10.

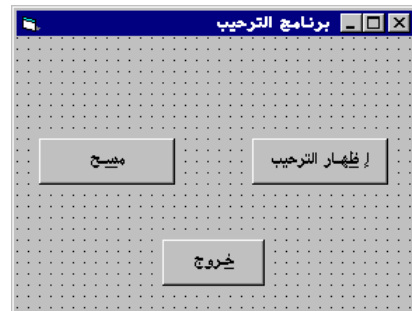
ستكتشف بعد هاتين الخطوتين أن مساحة الزر صغيرة جداً، لهذا كبر مساحة الزر، بسحب مقابضه.

يفترض أن يبدو النموذج لدى اكتماله كما في الشكل 1-20.

الشكل 20-1  
النموذج بعد انتهاء

تصميمه

مع ثلاثة أزرار داخله.



احفظ العمل المنجز حتى الآن:

□ اختر **Save Project** من القائمة **File** لفيجول بيسك.

## إضافة كائن مربع نص (Text Box) إلى النموذج frmHello

هناك كائن آخر يجب إضافته إلى النموذج، وهو كائن مربع النص. مربع النص عبارة عن مساحة مستطيلة، يتم إظهار نصوص أو كتابة نصوص فيها. يسمى مربع النص أحياناً بمربع تحرير.

### وضع مربع النص في النموذج

يبين الشكل 21-1 موقع رمز مربع النص Text Box في مربع الأدوات، وطبعاً قد تختلف مواقع الرموز في إطار مربع الأدوات عما هي عليه في الشكل 21-1 تبعاً للإصدار المستخدم للغة ولإعداداتها.

#### ملاحظة

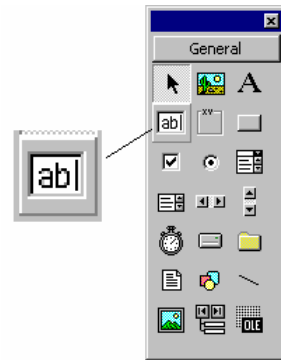
يظهر مستطيل أصفر يحوي النص TextBox بداخله، عند وضع مؤشر الفأرة فوق رمز مربع النص (دون ضغط أزرار الفأرة).

اتبع الخطوتين التاليتين لوضع مربع نص Text Box على النموذج:

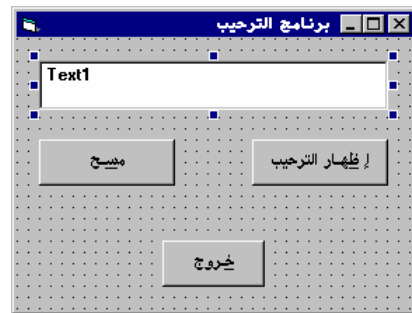
□ انقر نقرًا مزدوجاً على رمز مربع النص Text Box ضمن إطار مربع الأدوات.

□ انقل وغير حجم مربع النص حتى يظهر كما في الشكل 22-1.

الشكل 21-1  
رمز أداة مربع النص  
Text Box في إطار  
مربع الأدوات.



الشكل 22-1  
النموذج بعد وضع  
مربع نص بداخله.



## تبديل خصائص مربع النص

اتبع الخطوات التالية لتعديل بعض خصائص مربع النص:

- غير الخاصية Name لمربع النص من الاسم الافتراضي Text1 إلى txtDisplay.

خاصية Text الافتراضية لمربع النص هي Text1، وبالتالي فعند تنفيذ برنامج الترحيب يظهر النص Text1 ضمن مربع النص. نرغب أن يكون هذا المربع خالياً عند بدء تشغيل البرنامج، لهذا، احذف النص الذي يظهر في الخلية الواقعة يمين الخاصية Text للكائن txtDisplay.

- غير الخاصية Font لمربع النص txtDisplay إلى System واجعل حجم الخط مساوياً إلى 10.

تساوي القيمة الافتراضية للخاصية Alignment لمربع النص إلى 0-LeftJustify مما يعني أن النص في مربع النص يبدأ من اليسار. غير هذه الخاصية إلى 2-Center لأننا نريد للنص الظهور في وسط مربع النص.

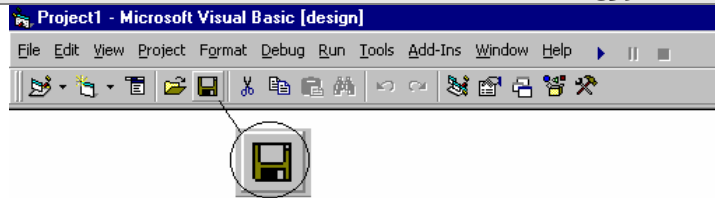
يرفض فيجول بيسك وضع النص في مركز مربع النص Text Box، ما لم يتم إسناد القيمة True إلى الخاصية MultiLine لمربع النص. لهذا يجب تغيير الخاصية MultiLine إلى True إلى جانب تغيير الخاصية Alignment إلى 2-Center. (يؤدي إسناد القيمة True إلى الخاصية MultiLine إلى

تمكين فيجول بيسك من إظهار أكثر من سطر واحد من مربع النص).  
إذاً بدل الخاصية MultiLine لمربع النص txtDisplay إلى True.  
احفظ العمل:  
□ اختر **Save Project** من القائمة **File** لفيجول بيسك.

### ملاحظة

حسب ما ذكرنا سابقاً، من الهام جداً حفظ العمل المنجز من وقت لآخر. طريقة أولى لحفظ العمل تتمثل باختيار **Save Project** من القائمة **File** لفيجول بيسك. لكن هنالك طريقة أسهل بكثير لحفظ العمل. ألق نظرة على الشكل 23-1، يظهر هذا الشكل شريط أدوات لغة فيجول بيسك. (إذا لم يكن شريط الأدوات ظاهراً، اذهب إلى القائمة **View** واختر **Toolbars** ثم انقر على العنصر Standard الذي يظهر في القائمة المنبثقة). نعود إلى الشكل 23-1. يوجد على شريط الأدوات رمز يظهر كقرص مرن، إذا وضعت مؤشر الفأرة (دون النقر على أحد أزرارها) فوق هذا الرمز، يظهر مستطيل أصفر يحمل الرسالة Save Project من القائمة **File**. أجل، لقد سهل مصممو فيجول بيسك كثيراً عملية حفظ المشروع، فنقرة واحدة من وقت لآخر تضمن لك السلامة. هناك طريقة سريعة أخرى لحفظ آخر التعديلات التي أجريتها على النموذج النشط الحالي، اضغط فقط المفاتيح Ctrl+S سوياً، ليحفظ فيجول بيسك تعديلات النموذج الحالي فقط، وليس كامل ملفات المشروع.

الشكل 23-1  
الرمز Save Project  
الواقع ضمن شريط  
أدوات لغة فيجول  
بيسك.



## عملية بناء النماذج انطلاقاً من الرسوم التوضيحية والجداول

انتهى جزء التصميم المرئي الآن.  
سنطالبك في كتابنا هذا ببناء عدد هائل من النماذج. لكن لن نستمر بهذه الطريقة في بناء النماذج، أي لن نبني النموذج خطوة بخطوة، بل سنزودك بشكل النموذج المكتمل (كما في الشكل 1-1)، كما سنعطيك جدولاً يدعى جدول خصائص النموذج. يحوي جدول خصائص

النموذج كل الكائنات المحتواة في النموذج ويستعرض كل الخصائص التي تختلف عن الخصائص الافتراضية لهذه الكائنات. عملك هو اللحاق بالجدول، سطرًا سطرًا وتغيير قيم الخصائص إلى القيم التي تظهر في الجدول. الجدول 1-1 هو جدول خصائص النموذج frmHello الذي أنهيته بناءً للتو.

الجدول 1-1. جدول خصائص النموذج frmHello.

القيمة	الخاصية	الكائن
<b>FrmHello</b>	<b>Name</b>	<b>Form</b>
Blue	BackColor	
برنامج الترحيب	Caption	
True	RightToLeft	
<b>cmdExit</b>	<b>Name</b>	<b>CommandButton</b>
&خروج	Caption	
System	FontName	
10	FontSize	
True	RightToLeft	
<b>cmdClear</b>	<b>Name</b>	<b>CommandButton</b>
&مسح	Caption	
System	FontName	
10	FontSize	
True	RightToLeft	
<b>cmdHello</b>	<b>Name</b>	<b>CommandButton</b>
&إظهار الترحيب	Caption	
System	FontName	
10	FontSize	
True	RightToLeft	
<b>txtDisplay</b>	<b>Name</b>	<b>TextBox</b>
2-Center	Alignment	
System	FontName	
10	FontSize	
True	MultiLine	
True	RightToLeft	

### ربط الكائنات بنصوص برمجية

باعتبار أننا وضعنا الكائنات في النموذج وحددنا خصائصها، فقد انتهى جزء التصميم المرئي (البرمجة المرئية Visual Programming). الآن، حان الوقت لكتابة النصوص البرمجية لهذه الكائنات.



تعتبر لغة فيجول بيسك لغة مقادة بالأحداث event-driven programming language. هذا يعني أن نص البرنامج ينفذ استجابة لحادثة ما. فمثلاً يؤدي نقر الزر **خروج** أثناء تنفيذ برنامج الترحيب إلى توليد الحادثة Click آلياً، وبالتالي ينفذ نص البرنامج المرافق لحادثة نقر الزر **خروج** آلياً أيضاً.

كذلك، تتولد الحادثة Click (حادثة النقر) أيضاً عند نقر الزر **إظهار الترحيب**، وينفذ نص البرنامج المرافق لحادثة نقر هذا الزر آلياً. عملك هو كتابة النص المناسب وربطه بالكائن والحادثة. هل يبدو هذا الأمر معقداً؟ بالواقع إنه سهل جداً، إذاً لنبدأ بربط بعض النص البرمجي بالحادثة Click للزر **خروج**.

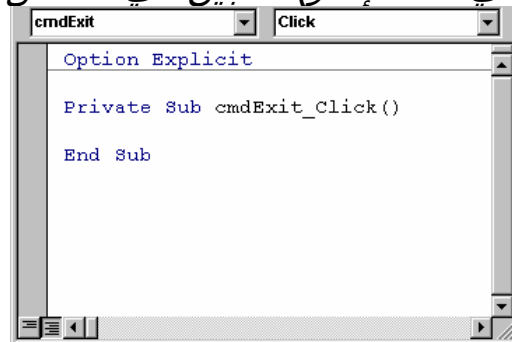
### ربط الزر خروج بنص برمجي

اتبع الخطوات التالية لربط نص برمجي بالزر cmdExit:

- انقر نقرأ مزدوجاً على الزر cmdExit.

يستجيب فيجول بيسك بإظهار إطار نص البرنامج (سنكتب برنامجاً في هذا الإطار) المبين في الشكل 24-1.

الشكل 24-1  
ربط الزر **خروج** بنص  
البرنامج المرافق، في  
إطار نص  
البرنامج Code Window.



يُسهّل فيجول بيسك إلى حد كبير التعرف على النص الظاهر حالياً في إطار نص البرنامج.

فكما يتبين من الشكل 24-1، يُظهر مربع السرد اليساري اسم الكائن cmdExit، ويظهر مربع السرد اليميني اسم الحادثة Click. كما يظهر من الشكل 24-1 أيضاً، فقد أضاف فيجول بيسك مسبقاً سطري نص يمثلان بداية ونهاية الإجراء الخاص بالكائن المحدد والحادثة المختارة، في إطار نص البرنامج:

```
Private Sub cmdExit_Click()
```

```
End Sub
```

سنكتب الآن نص البرنامج ضمن هذين السطرين.

## السطران الأول والأخير من نص البرنامج

السطر الأول من النص (الذي كتبه عنك فيجول بيسك) يبدأ بالكلمتين Private Sub. الكلمة Sub عبارة عن كلمة محجوزة في لغة فيجول بيسك، تدل على أن الإجراء Procedure يبدأ من هنا. والإجراء Procedure عبارة عن نص برنامج مكرس لحادثة خاصة. اسم الإجراء في مثالنا هو cmdExit\_Click(). السطر الأخير من النص، كُتب أيضاً من قبل لغة فيجول بيسك. وتشير العبارة End Sub لنهاية الإجراء.

## اسم الإجراء

اسم الإجراء هو cmdExit\_Click(). ما السبب الذي دفع فيجول بيسك إلى تعيين اسم الإجراء بنفسه؟. تسبب النقر المزدوج على الزر cmdExit بمعرفة فيجول بيسك أنك تحاول ربط ذلك الزر بنص برمجي، وبذلك حدد القسم الأول من اسم الإجراء وهو cmdExit. يمثل القسم الثاني من اسم الإجراء نوع الحادثة الناجمة (الحادثة Click).

أيضاً ما الذي جعل فيجول بيسك يحدد هذا الاسم (Click)؟! السبب طبعاً أنها تمثل الحادثة التي ترغب بكتابة برنامج لها. لاحظ أن القسمين الأول والثاني من اسم الإجراء يفصلان عن بعضهما برمز الخط التحتي ( \_ )، وبذلك يصبح اسم الإجراء كالتالي: cmdExit\_Click(). ولاحظ أيضاً أن آخر رمزين في اسم الإجراء هما القوسين ().

## نص برنامج الإجراء cmdExit\_Click()

ما هو نص البرنامج الذي يتوجب عليك كتابته في الإجراء cmdExit\_Click(). باعتبار أن هذا الإجراء ينفذ عند نقر الزر خروج (إنهاء البرنامج)، فالنص الواجب عليك كتابته ضمن هذا الإجراء يجب أن يتسبب بإنهاء البرنامج. تستخدم العبارة End لإنهاء البرنامج، وبالتالي فنص البرنامج اللازم كتابته ما هو إلا العبارة End فحسب. □ اكتب End في الإجراء cmdExit\_Click(). يُفترض أن يبدو الإجراء بعد كتابة هذه العبارة بالشكل التالي:

```
Private Sub cmdExit_Click()
```

End

**End Sub**

هذا كل ما في الأمر، لقد انتهيت للتو من ربط الإجراء cmdExit\_Click() بنص البرنامج المناسب.

### تنفيذ برنامج الترحيب

رغم أنك لم تنته بعد من ربط باقي الكائنات بنصوصها البرمجية، لكن هذا لا يمنع من تنفيذ برنامج الترحيب ورؤية كيف يعمل نص البرنامج الذي ربطناه بالزر **خروج**.

□ احفظ المشروع باختيار **Save Project** من القائمة **File**، أو انقر رمز

**Save Project** (رمز القرص المرن) على شريط أدوات فيجول بيسك.

□ اختر **Start** من القائمة **Run** لفيجول بيسك.

يستجيب فيجول بيسك بتنفيذ البرنامج، يظهر إطار هذا البرنامج بشكل يماثل ذلك المبين في الشكل 1-1.

تستطيع النقر على الزر **إظهار الترحيب** أو على الزر **مسح**، لكن لن يحدث شيء، والسبب طبعاً أننا لم نربط بعد هذين الزرين بأي نص برمجي.

□ انقر الآن الزر **خروج**، فينفذ الإجراء cmdExit\_Click() استجابة لحادثة النقر، وباعتبار أن نص هذا الإجراء يحوي العبارة End، فسيؤدي تنفيذه إلى إنهاء عمل برنامج الترحيب.

### إضافة المزيد إلى نص الإجراء cmdExit\_Click()

اتبع الخطوات التالية لربط الإجراء cmdExit\_Click() بالمزيد من النص البرمجي:

□ انقر نقرًا مزدوجاً على الزر cmdExit\_Click().

يستجيب فيجول بيسك بإظهار cmdExit\_Click() جاهزاً للتعديل من قبلك.

□ أضف العبارة Beep قبل العبارة End كما يلي:

**Private Sub cmdExit\_Click()**

Beep

End

**End Sub**

- اِحفظ العمل المنجز.
- نفذ برنامج الترحيب باختيار **Start** من القائمة **Run** لفيجول بيسك.
- تتسبب العبارة `Beep` بإصدار رنين (صافرة) من قبل الحاسب الشخصي. وبالتالي يؤدي نقر الزر **خروج** إلى إصدار صوت صافرة، ثم إنهاء برنامج الترحيب.
- انقر الزر **خروج** وتحقق من أن البرنامج أصدر رنيناً، ثم أنهى نفسه.

### ربط الزر إظهار الترحيب بنص البرنامج المناسب

اتبع الخطوات التالية لربط الزر **إظهار الترحيب** بنص برنامج:

- أظهر النموذج، ثم انقر نقرًا مزدوجاً على الزر **إظهار الترحيب**، (افتراضاً أنك ستظهر النموذج باختيار **Project Explorer** من القائمة **View**، ثم نقر البند `frmHello` في إطار المشروع **Project**، ثم نقر الرمز **View Object** الذي يظهر كثاني رمز على اليسار عند قمة إطار المشروع).

يستجيب فيجول بيسك بعد النقر المزدوج على الزر **إظهار الترحيب** بإظهار الإجراء `cmdHelloDisplay_Click()` مع سطري البرنامج التاليين:

```
Private Sub cmdHelloDisplay_Click()  
End Sub
```

يُنفذ الإجراء السابق عند نقر الزر **إظهار الترحيب** أثناء تنفيذ برنامج الترحيب.

يا ترى ما هو نص البرنامج الواجب ربطه بهذا الإجراء؟ يعتمد هذا على ما سيحصل عند نقر الزر **إظهار الترحيب**. ففي مثالنا هذا يُطلب من برنامج الترحيب إظهار رسالة الترحيب **مرحباً بكم** في مربع النص.

□ أدخل النص التالي ضمن الإجراء `cmdExit_Click()`:

```
txtDisplay.Text = "مرحباً بكم"
```

يفترض أن يبدو الإجراء كما يلي عند الانتهاء:

```
Private Sub cmdHelloDisplay_Click()
```

```
txtDisplay.Text = "مرحباً بكم"
```

```
End Sub
```

يمثل `txtDisplay` اسم الكائن `Text Box` (مربع النص) الذي سيظهر الكلمتين **مرحباً بكم** داخله).

تسند العبارة التالية:

```
txtDisplay.Text = "مرحباً بكم"
```

القيمة **مرحباً بكم** إلى الخاصية Text لمربع النص txtDisplay. (تمثل قيمة الخاصية Text النص الذي سيظهر في مربع النص txtDisplay).

#### ملاحظة

يُمكن استخدام الصيغة التالية لإسناد قيمة جديدة إلى خاصية ما، ضمن نص برنامج:

قيمة جديدة = ObjectName.Property

فمثلاً، لتبديل الخاصية Text لمربع النص txtDisplay إلى القيمة **مرحباً بكم** استخدم العبارة التالية:

```
txtDisplay.Text = "مرحباً بكم"
```

انتبه لكتابة النقطة (.) بين اسم الكائن txtDisplay واسم الخاصية Text وبدون فراغات بينهما.

### ربط الزر مسح بنص برنامج مناسب

اتبع الخطوات التالية لربط الزر **مسح** بنص برنامج الحادثة Click:

□ أظهر النموذج، وانقر نقراً مزدوجاً على الزر **مسح**.  
يستجيب فيجول بيسك للنقر المزدوج على الزر **مسح**، بإظهار إطار نص البرنامج *Code Window* عند الإجراء *cmdClear\_Click()* وجاهزاً للتعديل.

يفترض أن يقوم هذا الإجراء بمسح محتويات مربع النص. أي بمعنى آخر يجب تحويل قيمة الخاصية Text لمربع النص، إلى سلسلة صفرية (فراغ "")، يمكن إنجاز ذلك بإضافة العبارة التالية إلى الإجراء *cmdClear\_Click()*

```
txtDisplay.Text = ""
```

□ اكتب العبارة التالية في الإجراء *cmdClear\_Click()*:

```
txtDisplay.Text = ""
```

يفترض أن يبدو الإجراء لدى اكتماله كما يلي:

```
Private Sub cmdClear_Click()
```

```
txtDisplay.Text = ""
```

```
End Sub
```

□ احفظ العمل المنجز باختيار **Save Project** من القائمة **File** لفيجول بيسك (أو انقر على رمز القرص المرن على شريط أدوات لغة فيجول بيسك).

## تنفيذ برنامج الترحيب مرة ثانية

اكتمل برنامج الترحيب الآن. اتبع الخطوات التالية لتنفيذه:  
 □ اختر **Start** من القائمة **Run** لفيجول بيسك أو اضغط المفتاح F5 لبدء البرنامج.

### ملاحظة

إحدى الأسباب الرئيسية التي تجعل فيجول بيسك بهذا الشروع حقيقة، أنك تستطيع تطوير برنامجك قليلاً ثم تنفيذه ورؤية نتائج عملية التطوير على أرض الواقع، ثم متابعة تطوير البرنامج بعض الشيء وتشغيل التطبيق لاختيار عملية التطوير . . . وهكذا دواليك. ذكرنا، أنك تستطيع تنفيذ التطبيق المكتوب بلغة فيجول بيسك باختيار **Start** من القائمة **Run** لفيجول بيسك.

كما تلاحظ، تتطلب عملية تطوير التطبيق تكرار تنفيذه كثيراً، ولذلك راعى مصممو اللغة هذا الجانب، فوضعوا رمز التنفيذ على شريط أدوات لغة فيجول بيسك ويدعى هذا الرمز **Start** (انظر الشكل 1-25). ويكفي نقر الرمز **Start** على شريط الأدوات لتنفيذ التطبيق الراهن.

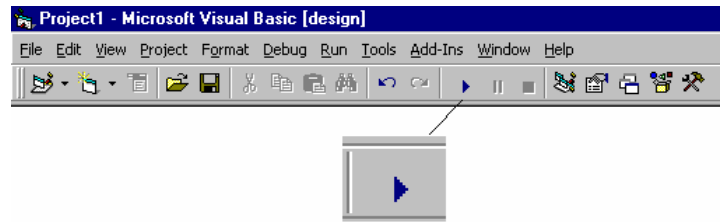
إذا كان شريط الأدوات غير ظاهر على سطح مكتب فيجول بيسك لسبب ما، اختر **Toolbars** من القائمة **View** لفيجول بيسك ثم اختر **Standard** من القائمة التي تظهر.

إذاً، تعتبر ميزة تنفيذ البرامج أثناء عملية التطوير، أمراً بالغ الأهمية، ففي لغات البرمجة الأخرى مثل Visual C++، يتوجب أولاً ترجمة ثم ربط البرنامج (Compile ثم Link) قبل التمكن من تنفيذه. وهذه العملية تأخذ وقتاً طويلاً نوعاً ما. يسهل فيجول بيسك عملية تنفيذ التطبيق الذي يجري تطويره إلى حد بعيد، لكن قد يظهر خطأ ما في البرنامج ويتسبب بانتهاء النظام كاملاً، مما يجبرك على إعادة تشغيل Windows أو فيجول بيسك مرة أخرى. يعني هذا أن عدم حفظ العمل قد يتسبب بضياعه وسنضطر إلى بدء عملية التطوير من جديد.

لهذا كن حكيماً، واجعل عملية حفظ العمل قبل تنفيذه قانوناً تسير عليه. والأمر جداً بسيط، فكما ذكرنا يقع رمز حفظ المشروع **Save** **Project** على شريط الأدوات. ويكفي نقره لحفظ العمل عند المرحلة الراهنة. لهذا انقر أولاً على الرمز

**Save Project** ثم انقر بعد ذلك الرمز **Start** (انظر الشكل 25-1) لتنفيذ التطبيق. وبهذه الطريقة، فإن وقوع خطأ ما أثناء تنفيذ البرنامج، لن يتسبب بضياع المشروع أو العمل، وتستطيع إعادة تشغيل فيجول بيسك مجدداً، وفتح المشروع المخزن وتصحيح الخطأ ومعاودة تنفيذ التطبيق.

الشكل 25-1  
رمز التنفيذ على  
شريط  
أدوات  
فيجول  
بيسك.



- انقر الزر **إظهار الترحيب** وأيضاً الزر **مسح** لإظهار ثم مسح الرسالة مرحباً بكم، في مربع النص. يجب أن يظهر إطار برنامج الترحيب كما في الشكل 2-1، بعد
- نقر الزر **إظهار الترحيب**. بينما سيظهر كما في الشكل 1-1 عند نقر الزر **مسح**.
- تستطيع استخدام المفاتيح **Alt+ظ** و **Alt+س** للحصول على نفس الاستجابة عند نقر الزر **إظهار الترحيب** وعند نقر الزر **مسح**. (طبعاً، بسبب وجود خط تحت الحرف **ظ** في عنوان الزر **إظهار الترحيب**، وخط تحت الحرف **س** في عنوان الزر **مسح**).
- يمكن إنهاء البرنامج بنقر الزر **خروج**، أو بضغط المفاتيح **Alt+خ** على لوحة المفاتيح.

## أحداث أخرى

يستخدم برنامج الترحيب الحادثة **Click** لأزرار الأوامر، (فمثلاً، تقع الحادثة **Click** لزر الأمر **cmdExit** عند نقر الزر **خروج** مما يتسبب بالتنفيذ الآلي للإجراء **cmdExit\_Click()**. هنالك حوادث أخرى يمكن استخدامها في البرنامج. وكل حادثة تمتلك بدورها إجراءاتها الخاص.

## الحادثة **KeyDown**

- سنتعرف على الإجراء الذي يرافق الحادثة **KeyDown**، والتي تحصل عند ضغط مفتاح ما على لوحة المفاتيح.
- اتبع الخطوات التالية لرؤية الإجراء **KeyDown** للزر **خروج**.
- انقر نقرًا مزدوجاً على الزر **خروج**.

يظهر فيجول بيسك في الحالة الافتراضية الإجراء المرافق للحادثة *Click*.

□ باعتبار أننا لا نريد تعديل الإجراء المرافق للحادثة *Click* وإنما نريد تعديل إجراء الحادثة *KeyDown*، لهذا انقر على مربع السرد الذي يظهر في الجانب الأيمن عند قمة إطار النص.

يستجيب فيجول بيسك بسرد لائحة تحوي كل الحوادث المتاحة للكائن المدعو *cmdExit* (انظر الشكل 26-1).

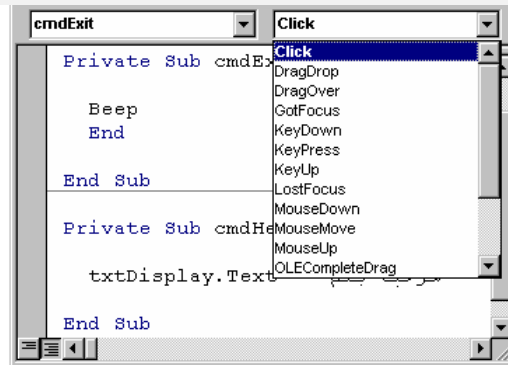
□ اختر البند *KeyDown* من اللائحة.

يستجيب فيجول بيسك بإظهار الإجراء *cmdExit\_KeyDown()*:

```
Private Sub cmdExit_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
End Sub
```

الشكل 26-1  
إظهار الحوادث المتاحة  
للزر *cmdExit*.



لاحظ أن السطر الأول من الإجراء والذي يكتبه فيجول بيسك آلياً، يختلف قليلاً عن السطر الأول للحادثة *Click* فقوسي الإجراء *cmdExit\_Click()* لا يحويان شيئاً داخلهما. أما قوسي الإجراء *cmdExit\_KeyDown()* فيحويان قدرّاً من النص البرمجي. سنتعرف خلال الفصول القادمة على هذا الاختلاف بشكل أوضح.

أما عند هذه النقطة، فلا تُضف شيئاً إلى الإجراء *cmdExit\_KeyDown()* (مضينا في هذا المثال، لمجرد إطلاعك على أشكال أخرى من الحوادث التي ترافق كائن ما في فيجول بيسك).

## إنشاء الملف التنفيذي للمشروع (Hello.exe)

نفذنا سابقاً برنامج الترحيب باختيار *Start* من القائمة *Run*. طبعاً يختلف الأمر عند الانتقال إلى الواقع العملي، فلا أحد يرضى أن يشغل التطبيق الذي بحوزته بهذه الطريقة، بل قد لا يملك المستخدم الذي اقتنى التطبيق، لغة فيجول بيسك أصلاً.



يجب أن تكون قادراً إذاً على تحويل البرنامج إلى ملف تنفيذي، حتى تتمكن من توزيع التطبيق الذي طورناه.

فمثلاً يجب تحويل برنامج الترحيب إلى ملف تنفيذي Hello.exe:

- اختر Make Hello.exe File من القائمة **File** لفيجول بيسك.
- يستجيب فيجول بيسك بإظهار مربع الحوار *Make*. (لا تنقر الزر **OK** في مربع الحوار في هذه اللحظة).
- استخدم مربع الحوار لحفظ الملف Hello.exe في الدليل C:\VB5Prg\Ch01.
- انقر الزر **OK** الآن.
- يستجيب فيجول بيسك بحفظ الملف *Hello.EXE* في الدليل C:\VB5Prg\Ch01.
- تستطيع الآن تنفيذ Hello.exe كأى برنامج آخر في Windows!، تستطيع مثلاً استخدام مستكشف Windows ثم النقر المزدوج على الملف Hello.exe.

### ملف تنفيذي EXE صغير الحجم

ألق نظرة على الملف Hello.exe الموجود حالياً ضمن الدليل C:\VB5Prg\Ch01. كيف يتمكن ملف بحجم 8 كيلو بايت فقط من إنجاز كل الأشياء التي يستطيع Hello.exe إنجازها؟

فكر بالأمر، ينجز البرنامج Hello.exe أموراً كثيرة، فيمكنك نقر الأزرار الثلاثة، ويسمح مثلاً بسحب إطار البرنامج، يمتلك البرنامج أيضاً مربع نص داخله، بمعنى آخر، يمتلك هذا البرنامج الصغير كل معالم البرامج القياسية في Windows. والسبب وراء صغر حجم الملف التنفيذي Hello.exe، أن البرنامج يفترض وجود الملف Msvbvm50.DLL في الدليل System. والدليل System موجود طبعاً ضمن دليل Windows.

فمثلاً ترى الدليل C:\Windows\System إذا كان نظام التشغيل المستخدم هو Windows 95 أو في الدليل C:\WinNT\System32 عندما يكون نظام التشغيل المستخدم هو Windows NT. المهم بغض النظر عن نظام التشغيل المستخدم يحتاج البرنامج Hello.exe أن يكون الملف Msvbvm50.DLL موجوداً ضمن دليل النظام System حتى يعمل. تستطيع العودة إلى ذلك الدليل والتأكد من وجود الملف المذكور. لقد تم تثبيت هذا الملف عند تثبيت فيجول بيسك 5 على قرصك الصلب.

حجم الملف Msvbvm50.DLL يساوي 1.3 ميغا بايت تقريباً، طبعاً يعتبر كبيراً، لكن حال توفر هذا الملف يصبح بوسع المستخدمين تنفيذ برامج بلغة القوة كتبت بلغة فيجول بيسك 5. النبأ السار أن هذه البرامج القوية صغيرة الحجم جداً. تستطيع أيضاً رؤية الملف Hello.vbw في الدليل الذي حفظنا فيه الملف Hello.exe. الملف Hello.vbw يستخدم من قبل فيجول بيسك، ولا يتوجب عليك تزويد المستخدمين بهذا الملف.

#### ملاحظة

يتوجب عليك تزويد البرامج التي توزعها بملفات DLL اللازمة لها. فمثلاً يجب أن يكون الملف Msvbvm50.DLL موجوداً ضمن الدليل System على الحاسب الذي ستنفذ فيه البرامج المطورة بلغة V.B.5.

#### الخلاصة

كتبنا في هذا الفصل أول برنامج لك بلغة Visual Basic فتعلمت الخطوتين اللازمتين لكتابة البرنامج، وهما خطوة البرمجة المرئية Visual Programming Step وخطوة كتابة نص البرنامج Code Programming Step. يضع المبرمج الكائنات Objects في النموذج ويسند خصائصها بالقيم المناسبة أثناء خطوة البرمجة المرئية. أما في خطوة كتابة النص Code Programming، فيتم انتقال الإجراء المرافق لكائن ما باختيار الكائن والحادثة ثم يكتب نص البرنامج المناسب في الإجراء. ينفذ جزء البرنامج المكتوب هذا زمن التنفيذ عند وقوع الحادثة.

## الفصل الثاني

---

Option Button

Text Boxes

Scroll Bar

.Command Button

)

.(

.

.User Interface

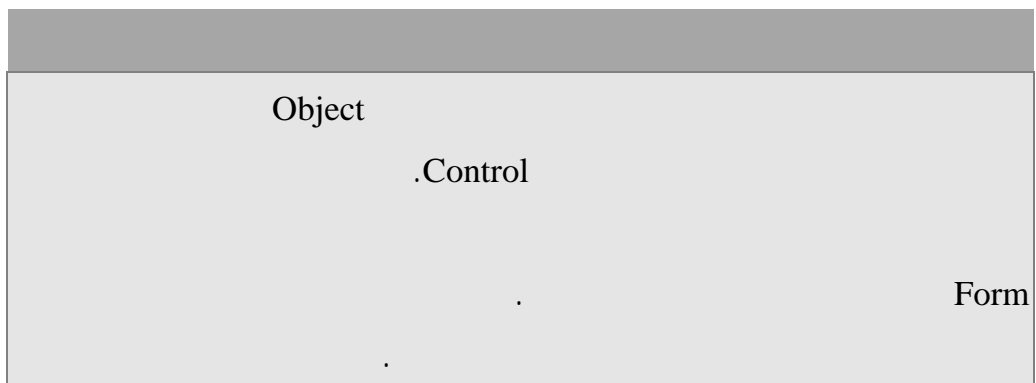
Controls

.(

)

.

Scroll Bar

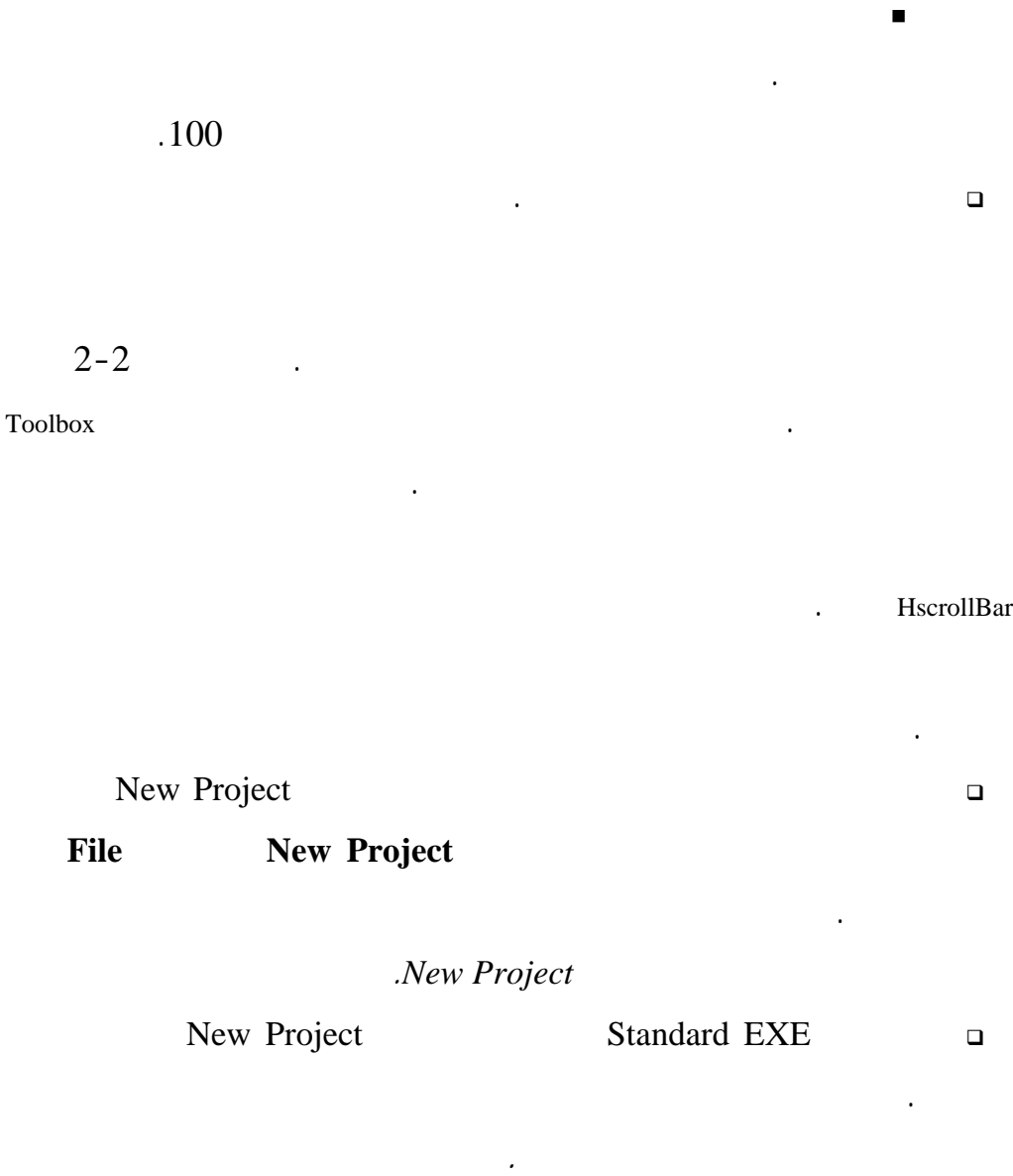


1-2

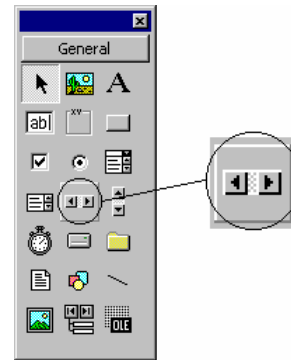
50" / " ( )

1-2





2-2



:

.C:\VB5Prg\Ch02

**Save Form1 As**

Form1

**.File***.Save File As*

Speed.Frm

Save File As

**File****Save Project As**

C:\VB5Prg\Ch02

Speed.vbp

Save Project As

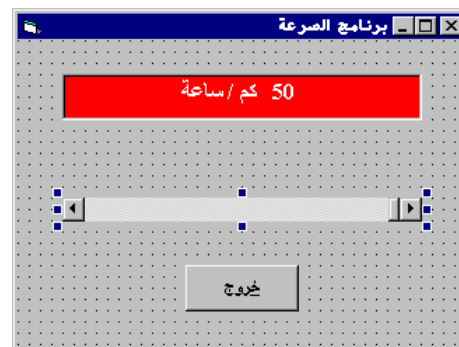
.1-2

Form1

*.3-2*

3-2

frmSpeed

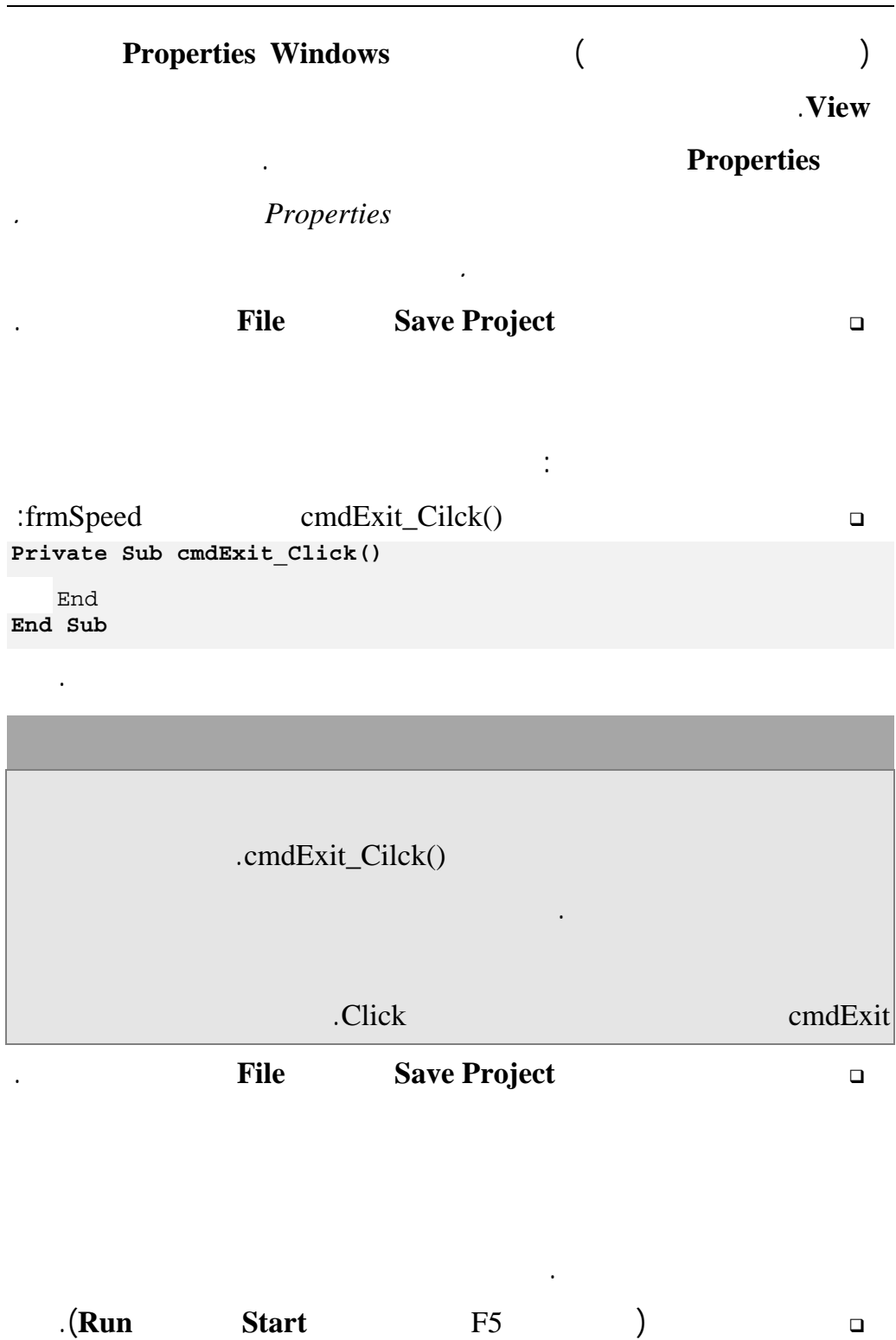


**.frmSpeed****.1-2**

<b>Form</b>	<b>Name</b>	<b>FrmSpeed</b>
	BackColor	Light gray
	Caption	
	RightToLeft	True
<b>CommandButton</b>	<b>Name</b>	<b>cmdExit</b>
	Caption	&
	RightToLeft	True
<b>Horizontal Scroll Bar</b>	<b>Name</b>	<b>hsbSpeed</b>
	Min	0
	Max	100
	RightToLeft	True
<b>TextBox</b>	<b>Name</b>	<b>txtSpeed</b>
	Alignment	2-Center
	Font	(            )
	BackColor	Red
	ForeColor	White
	Text	/    50
	MultiLine	True
	RightToLeft	True

□

□





4-2.

4-2



□

□

Value    Max    Min

:

Max    Min

Min

1-2

Min

Max

100

Max

.100

Value

Value

100

)

Value

(

---

) Value  
.(Value  
50 50  
:  
Value  
.Value 50 □  
.(100 )  
Text " / 50" 1-2  
50"  
.(Value = 50) " /  
**Focus**  
Tab  
.  
)  
.( ..  
.  
( )  
.  
.  
.  
:  
) . □  
.( Value 50  
Tab □

---

```
.( )

.

End Home .

PgDn PgUp

Space Tab □

.Enter

.!.
)

(

.

.

( )

.

:

□

.hsbSpeed_Change()

hsbSpeed

Change

.hsbSpeed_Change()

:hsbSpeed_Change() □
```

```
Private Sub hsbSpeed_Change()
```

```
    txtSpeed.Text = Str(hsbSpeed.Value) + " كم / ساعة "
```

```
End Sub
```

```
        ( ) hsbSpeed_Change()
```

```
        . Value .
```

```
        Value
```

```
        .Min
```

```
True
```

```
    RightToLeft
```

```
    .RightToLeft
```

```
    .( )
```

```
        Value
```

```
        .
```

```
        :
```

```
        Text
```

```
txtSpeed.Text = Str(hsbSpeed.Value) + " كم / ساعة "
```

```
Text
```

```
20
```

```
Value
```

```
    . / 20
```

```
Value
```

```
( )
```

```
Text
```

```
Str()
```

```
    .
```

```
Str()
```

```
    .
```

```
Value
```

```
Str(11)
```

```
Str(12345)
```

```
    ."11"
```

```
11
```

```
    ."12345"
```

```
12345
```

```
Val()
```

```
Str()
```

```
    .
```

.(1234)	("1234")
---------	----------

hsbSpeed.Value

:

```
Str(hsbSpeed.Value)
```

32

: (hsbSpeed.Value = 32)

```
txtSpeed.Text = Str(hsbSpeed.Value) + " كم / ساعة "
```

/ 32 : Text

:

.File Save Project ☐

. ☐

. ☐

. ☐

. ☐

.

:

. ☐

( ) ☐

.

!

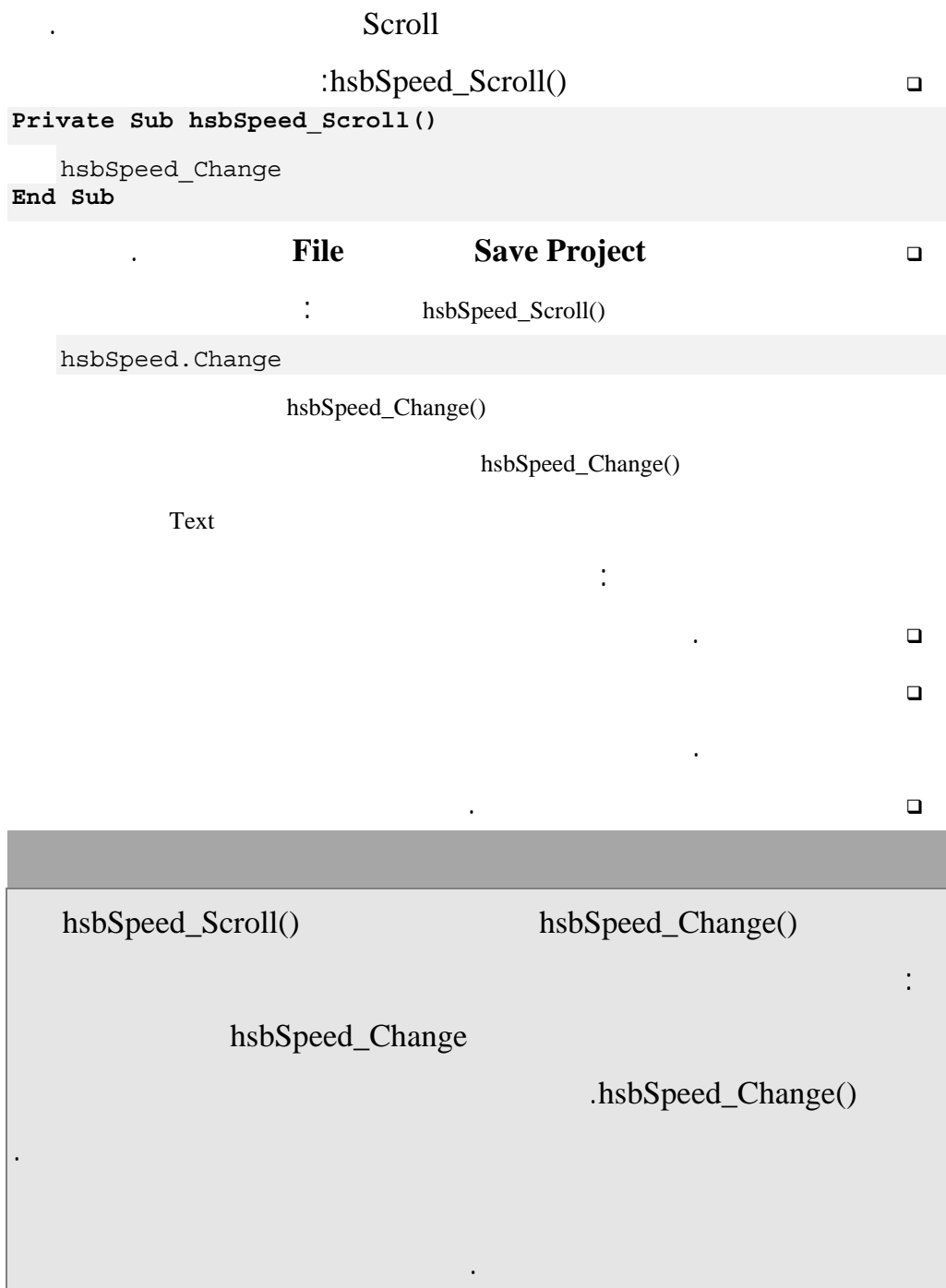
.

:

hsbSpeed\_Scroll()

☐

hsbSpeed\_Scroll()



100

Feedback

)

(

.Options

5 - 2

.Options Button

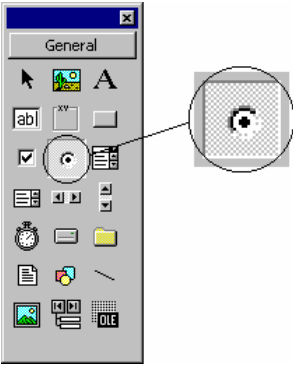
.(

Options Button

)

5-2

Options



**File**

**New Project**



.New Project

Standard EXE

) Form1 □  
**File Save Form1 As** .(  
 Save File As .  
 .C:\VB5Prg\Ch02 Options.Frm  
**File Save Project As** □  
 Options.Vbp Save Project  
 .C:\VB5Prg\Ch02  
 .2-2 frmOptions □

.6-2

6-2

frmOptions



.frmOptions

.2-2

Form	Name	frmOptions
	BackColor	Red
	Caption	
	RightToLeft	True
CommandButton	Name	cmdExit
	Caption	&
	RightToLeft	True
Check Box	Name	chkSound



	BackColor	Red
	Caption	&
	Font	( )
	ForeColor	White
	RightToLeft	True
<b>Check Box</b>	<b>Name</b>	<b>chkMouse</b>
	BackColor	Red
	Caption	&
	Font	( )
	ForeColor	White
	RightToLeft	True

<b>Check Box</b>	<b>Name</b>	<b>chkColors</b>
	BackColor	Red
	Caption	&
	Font	( )
	ForeColor	White
	RightToLeft	True
<b>Option Button</b>	<b>Name</b>	<b>optLevel1</b>
	BackColor	Red
	Caption	1&
	Font	( )
	ForeColor	White
	RightToLeft	True
<b>Option Button</b>	<b>Name</b>	<b>optLevel2</b>
	BackColor	Red
	Caption	2&
	Font	( )
	ForeColor	White
	RightToLeft	True

2-2 frmOptions . . .

(General)



(Declarations)

*General Declarations*

.7-2

:7-2

```
Private Sub Form_Load()  
  
End Sub
```

7-2

:

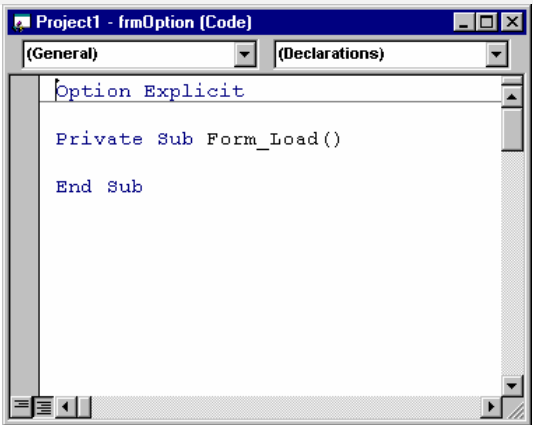
Form\_Load()

Option Explicit

7-2

General Declarations

.Code Window



:

Option Explicit

Option Explicit

**Click**

:frmOptions cmdExit\_Click()



```
Private Sub cmdExit_Click()  
  
End
```

•

•



**.1**



) **1**

**. (1**

**.2**



) **1**

) **2**

**(1**

.

**.3**



*.3*

2

$$).$$

## Radio Button

.(





:optLevel3\_Click()



Private Sub OptLevel3\_Click()

UpdateLabel

End Sub

:

!

UpdateLabel

UpdateLabel

.

.

:

:

UpdateLabel

.



.(Code Window)

.

Tool

Add Procedure



.Add Procedure

) .Add Procedure

Name

UpdateLabel



.(

UpdateLabel

.8-2

Add Procedure

8-2

.UpdateLabel

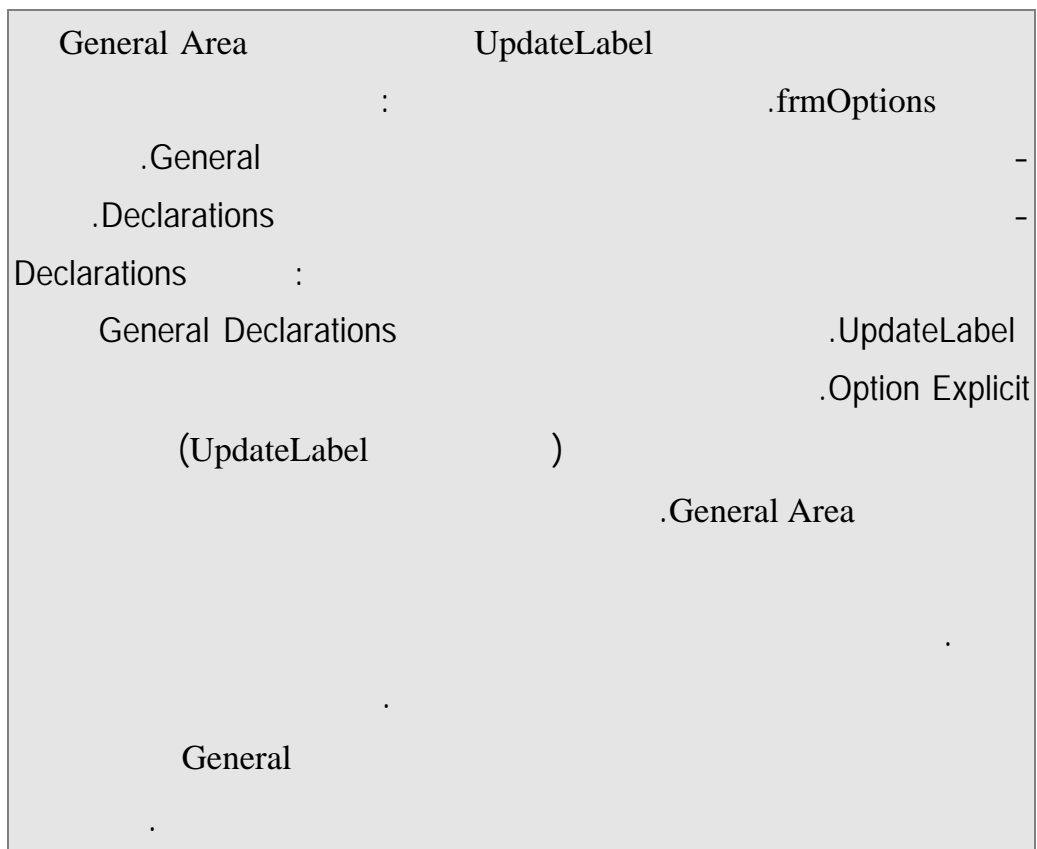
.Add Procedure

OK



UpdateLabel

.



:

:UpdateLabel


**Public Sub UpdateLabel()**

```

Dim Info
Dim LFCR
LFCR = Chr(13) + Chr(10)
' الصوت
If chkSound.Value = 1 Then
    Info = "الصوت : تشغيل"
Else
    Info = "الصوت : إيقاف"
End If
' الفأرة

```

```

If chkMouse.Value = 1 Then
    Info = Info + LFCR + "الفأرة : تشغيل"
Else
    Info = Info + LFCR + "الفأرة : إيقاف"
    f' الألوان
If chkColors.Value = 1 Then
    Info = Info + LFCR + "الألوان : تشغيل"
Else
    Info = Info + LFCR + "الألوان : إيقاف"
End If
' المستوى 1
If optLevel1.Value = True Then
    Info = Info + LFCR + "المستوى : 1"
End If
'
If optLevel2.Value = True Then
    Info = Info + LFCR + "المستوى : 2"
End If
If optLevel3.Value = True Then
    Info = Info + LFCR + "المستوى : 3"
End If

lblChoice.Caption = Info
End Sub

```

**.File**

**Save Project**



:UpdateLabel



.(9-2

) lblChoice



9-2



UpdateLabel

**frmOptions****chkColors\_Click()**

.chkColors

**Private Sub** chkColors\_Click()

UpdateLabel

**End Sub**

UpdateLabel

( )

.UpdateLabel

**UpdateLabel**

Add Procedure

```

    .
    (
        ) cmdExit_Click()
        cmdExit_Click() UpdateLabel
    .!
    Click .
    !
    .cmdExit_Click()
    UpdateLabel
    pdateLabel
    :
    UpdateLabel
    .UpdateLabel
    .
    Info
    :
    Info UpdateLabel
    Dim Info
    (
        Info) Dim
    Info
    .
    .UpdateLabel
    :
    Dim Info As String

```

```

:
Time = 10
Velocity = 50
Distance = Velocity * Time
lblDistance.Caption = " = المسافة " + Str(Distance)
50      Time      10
Distance      Velocity
.lblDistance      Caption      Distance
a      )
:      (Distance      t
lblDistance.Caption = " = المسافة " + Str(Distance)
(      ) Distnce
Distance
:      lblDistance
0 = المسافة
.
.
:
Dim Time
Dim Velocity
Dim Distance
Time = 10
Velocity = 50

```

```
Distance = Velocity * Time
lblDistance.Caption = " = المسافة " + Str(Distance)
```

Distnce

:

!

Option Explicit

General Declarations

Option Explicit

Option Explicit

Option Explicit

**LFCR**

:UpdateLabel

LFCR

Dim LFCR

:

LFCR

LFCR = Chr(13) + Chr(10)

Chr(10)

Enter

Chr(13)

.(

)

lblChoice

.LFCR

**Value**

:If .Else.End      UpdateLabel

```

الصوت'
If chkSound.Value = 1 Then
    Info = "الصوت : تشغيل"
Else
    Info = "الصوت : إيقاف"
End If

```

:      .Rem      (')

الصوت'

:

Rem الصوت

.

:

MyVariable = 1 'تهيئة المتحول

(')

.

.(')

chkSound

Value

UpdateLabel

Value

.1

Else    If

Else

If

:

```

Info = "الصوت : تشغيل"
Value
.If Then .Info " :
Value
chkSound
: Info
"الصوت : تشغيل"
Value End If Else
chkSound
Value
: Info End If Else
"الصوت : إيقاف"
: Info
"الصوت : تشغيل"
:
"الصوت : إيقاف"
Value If.Else.End If
: chkMouse
' الفأرة
If chkMouse.Value = 1 Then
Info = Info + LFCR + "الصوت : تشغيل"
Else
Info = Info + LFCR + "الصوت : إيقاف"
End If
chkSound
If.Else.End If chkMouse
:Info UpdateLabel
"الفأرة : إيقاف" + LFCR + "الصوت : تشغيل"
:

```

الصوت : تشغيل  
الفأرة : إيقاف

LFCR

Value UpdateLabel If.Else.End If  
:  
Info chkColors

```
الألوان '
If chkColors.Value = 1 Then
    Info = Info + LFCR + " تشغيل : الألوان "
Else
    Info = Info + LFCR + " إيقاف : الألوان "
End If
```

Value UpdateLabel If.End.If

:True optLevel1

```
If optLevel1.Value = True Then
    Info = Info + LFCR + " المستوى : 1 "
End If
```

Value Value  
Info True  
True optLevel1 Value

.

	2	1	0
			Value
			.
2		Value	
			.

True	.False	True	Value

False	.	.
-------	---	---

Value

Info

If.End.If

:optLevel3 optLevel2

```
If optLevel2.Value = True Then
```

```
    Info = Info + LFCR + "2 : المستوى"
```

```
End If
```

```
If optLevel3.Value = True Then
```

```
    Info = Info + LFCR + "3 : المستوى"
```

```
End If
```

Caption

Info

UpdateLabel

:lblChoice

```
lblChoice.Caption = Info
```

-2

lblChoice

Info

.9

!

.

.

Property

( ) Caption

( ) Label

Caption

Value

.

Value

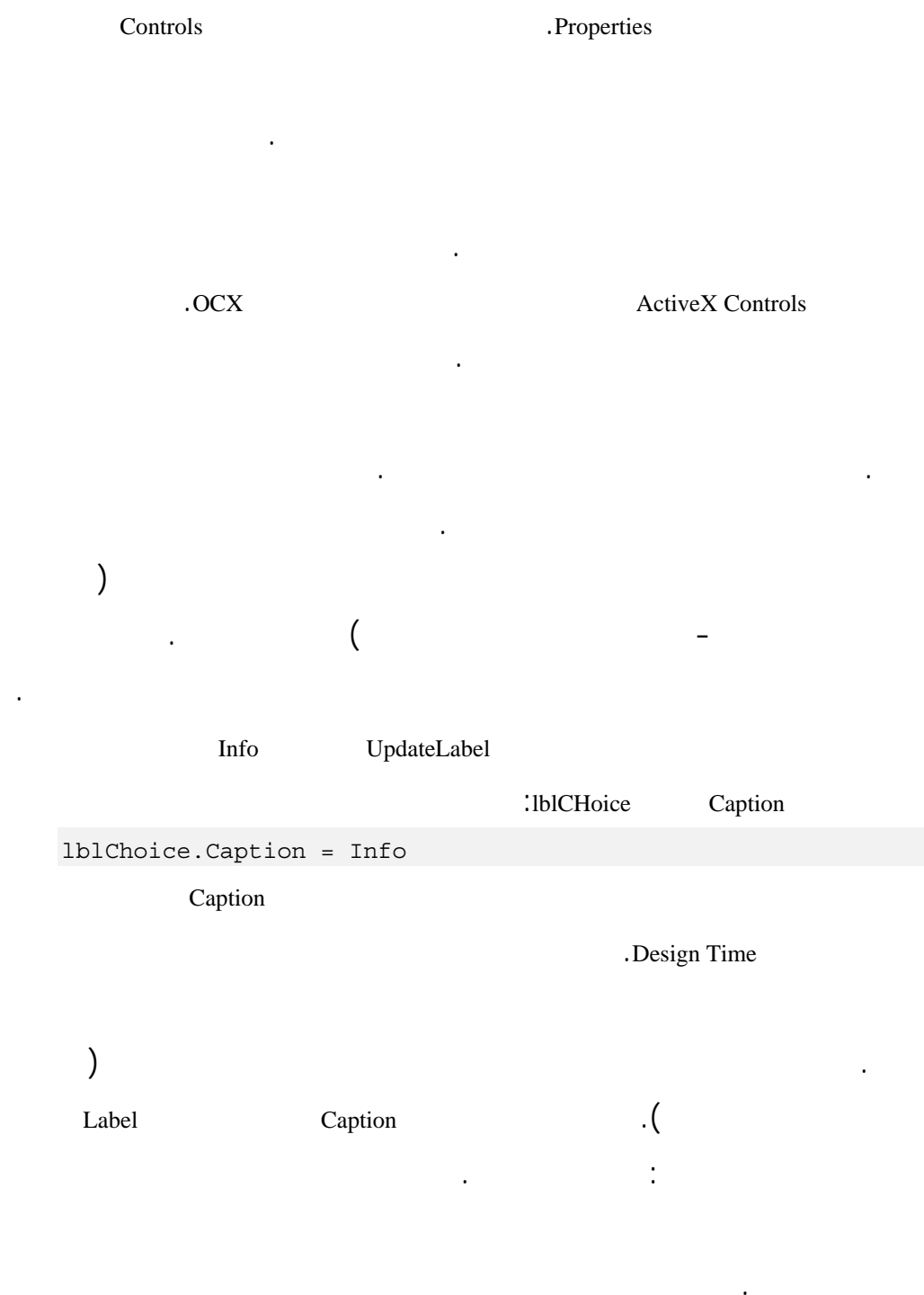
.

Value

.

.





## .3-2

txt (cmdMyButton ) cmd  
 .(txtMyTextBook )

## .Controls

## .3-2

Check box	chk	chkReadOnly
Combo box	cbo	cboEnglish
Command button	cmd	cmdExit
Common dialog	dlg	dlgFileOpen
Communications	com	comFax
Data control	dat	datBiblio
Directory list box	dir	dirSource
Drive list box	drv	drvTarget
File list box	fil	filSource
Form	frm	frmEntry
Frame	fra	fraLanguage
Grid	grd	grdPrices
Horizontal scroll bar	hsb	hsbVolume
Image	img	imgIcon
Label	lbl	lblHelpMessage
Line	lin	linVertical
List box	lst	lstPolicyCodes
MCI	mci	mciVideo
MDI child form	mdi	mdiNote
Menu	mnu	mnuFileOpen
Picture	pic	picVGA
Shape	shp	shpCircle
Text box	txt	txtLastName
Timer	tmr	tmrAlarm
UpDown	upd	updDirection

Vertical scroll bar	vsb	vsbRate
---------------------	-----	---------

3-2

Name

Label

lblChoice

.

Name

Choice

.

UpdateLabel

lblChoice

:

```
Choice.Caption = Info
```

:

```
lblChoice.Caption = Info
```

3-2

.

:

```
Choice.Caption = Info
```

.Label

Choice

.Choice

:

```
lblChoice.Caption = Info
```

lblChoice

."

Caption

Info

"

.lbl

:

```
MyVariable = 1 + 2 + 3
```

:

```
MyVariable = 1 + _
              2 + 3
```

:

```
MyVariable = 1 _
              + 2 _
```

---

+ 3

.Underscore ( \_ )

```
lblMyLabel.Caption = "This is my string "
:
:
lblMyLabel.Caption = "This is
my string_"
:
:
lblMyLabel.Caption = "This is _
& "my string "
```

```

:
.
Value
.
.(FrmOpptioon Update Label )
```