الفصل الحادى عشر
أدوات التحكم فى عرض البيانات

سنتعرف في هذا الفصل على مجموعة من أقوى عناصر التحكم في Visual Basic وهي المجموعة الخاصة بالتحكم فى العروض وإظهار البيانات المساعدة للمستخدم. تشمل هذه المجموعة شريط المعلومات ومقياس التقدم وأداة التعقب (المنزلق) وأخيراً عارض القوائم والعارض الشجري.

بانتهاء هذا الفصل ستتعرف على:

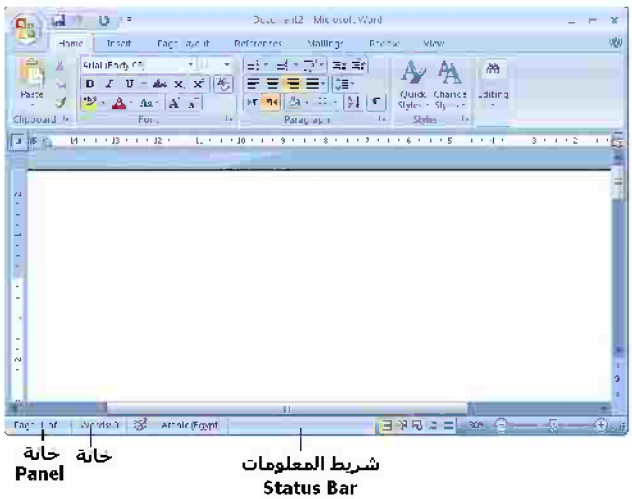
- *إنشاء شريط معلومات للمستخدم.*
- *بيان نسبة إنجاز الأعمال باستخدام مقياس التقدم.*
- *تسهيل إدخال الأرقام باستخدام المنزلق.*
- *إظهار قائمة من الكائنات باستخدام العارض الشجرى وعارض القوائم.*

يحتوي Visual Basic على مجموعة من أدوات التحكم القوية المستخدمة فى عرض البيانات بطرق مختلفة والتي تمكنك من إنشاء برامج لها نفس صفات البرامج التي تنتجها الشركات الكبيرة.

تتضمن هذه الأدوات: شريط المعلومات Status Bar والذي يمد المستخدم بمعلومات سريعة وشريط التقدم Progress Bar الذى يبين معدل إنجاز الأعمال وأداة التعقب (المنزلق) TrackBar المستخدم لإدخال البيانات بالإضافة إلى عارض القوائم والعارض الشجرى الذى يمكنك من عرض البيانات فى صورة قائمة أو فى صورة هرمية على الترتيب بنفس الأسلوب المستخدم فى مستكشف Windows. في هذا الفصل سنناقش هذه الأدوات وغيرها بشيءٍ من التفصيل.

إظهار معلومات عن البرنامج باستخدام شريط المعلومات Status Bar

يحب المستخدم دائماً أن يعرف معلومات عن البرنامج في أي لحظة، دون أن يقوم بنقر زر أو رؤية رسالة في نافذة منفصلة، وإنما يكون ذلك من خلال نفس واجهة البرنامج وذلك من خلال شريط المعلومات Status Bar. مثال لشريط المعلومات الشريط الذى يظهر أسفل نافذة المستند فى برنامج Microsoft Word (انظر شكل ١١-١).

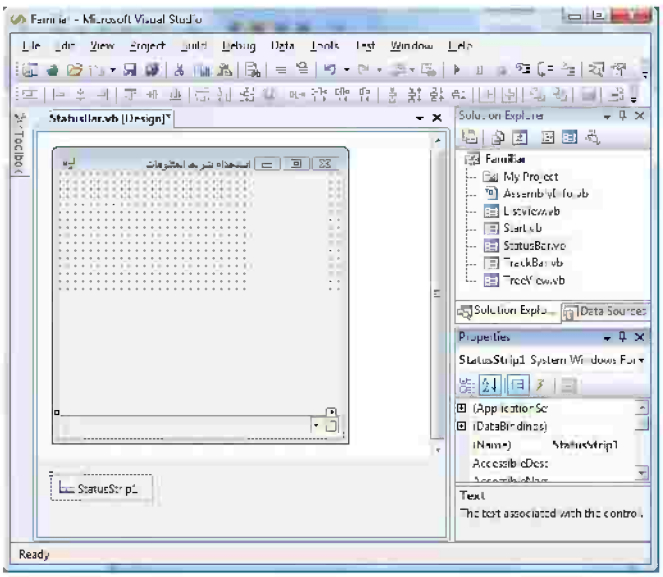


شكل ١١-١ نموذج لشريط المعلومات في برنامج Microsoft Word

في الماضي (قبل Visual Basic 4 على سبيل التحديد) كان المبرمج يقوم بتفصيل شريط معلومات من مكونات أخرى مثل اللوحة Panel، والعنوان Label. ولكن مع ظهـــور أداة شريط المعلومات صار الأمر أسهل كثيراً. كما سترى في الشرح التالى.

إنشاء شريط معلومات Status Strip

لإضافة شريط معلومات إلى نموذج قم باختيار الرمز من شريط الأدوات، ثم ارسمه في أي مكان على النموذج. سيأخذ موضعه الطبيعي تلقائياً في أسفل النموذج وبحجم افتراضي يتمثل فى عرض النموذج المستخدم ويحتوى على خانة واحدة Panel (انظر شكل ١١-٢).



شكل ١١-٢ شريط المعلومات عند إضافته للنموذج

الحجم الافتراضي الذي تأخذه اللوحة بعرض النافذة ويظل كذلك طول فتـــرة تشـــغيل البرنامج ولا يمكنك التحكم فيه أو تغييره، أما ارتفاع الشريط فيمكن تحديده بالخاصـــية Height أو تكبير الارتفاع قليلاً أثناء التصميم.

بعد إضافة شريط المعلومات، يجب أن تقوم بإعداده ليناسب المعلومات المراد عرضها من خلاله. وبخلاف حجم الشريط، هناك خاصيتان يتحكمان في شكل الشريط وموضعه. الخاصية الأولى هي Dock وتحدد الجانب الذي سيرسو عليه الشريط في النافذة، ويأخذ القيم الموضحة بجدول ١١-١ التالى:

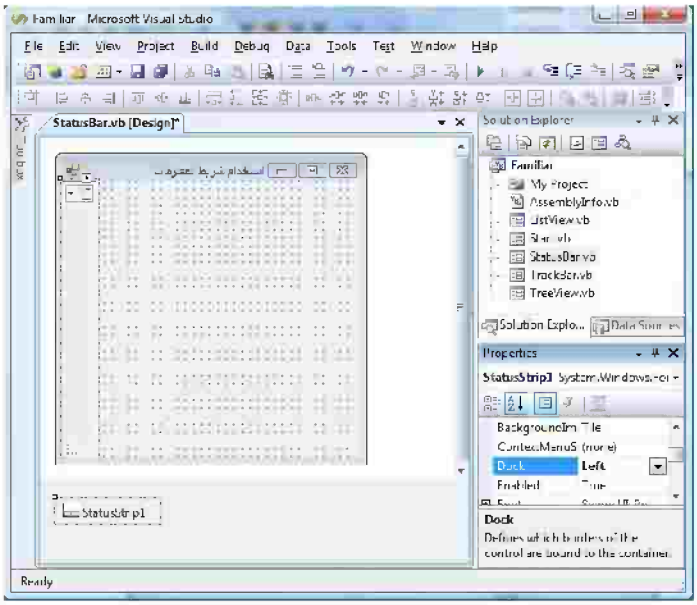
جدول ١١-١ قيم الخاصية Dock داخل شريط المعلومات

القيمة	مكان شريط المعلومات
Top	أعلى النافذة
Bottom	أسفل النافذة
Left	يسار النافذة
Right	يمين النافذة
Fill	بكامل النافذة
None	يمكن وضع الشريط فى أى مكان بالنموذج

فى الشكل ١١-٢ السابق ظهر شريط المعلومات أسفل النافذة وذلك لاحتواء الخاصية Dock على القيمة Bottom. وإذا قمت بتغيير هذه القيمة إلى Left، يظهر الشريط بالجانب الأيسر من النافذة (انظر شكل ١١-٣).

عموماً .. إن لم يكن لديك سبب قوي لوضع شريط المعلومات فى أى من الجوانب الأخرى للنافذة، فيستحسن الإبقاء على الشكل القياسي وهو وضع الشريط أسفل النافذة.

غالباً ما يحدث عند تغيير الخاصية Dock أن يملأ الشريط مساحة النموذج بالكامل، عليك حينئذ تغيير الخاصية Height أو Width تبعاً لوضع شريط المعلومات لتصغير الشريط.



شكل ١١-٣ شريط المعلومات بالجانب الأيسر من النافذة

التحكم في خانات شريط المعلومات

يتم تمثيل الوظيفة الحقيقية لشريط المعلومات من خلال الخانات التى يحتويها هذا الشريط. قم بتخصيص القيمة **True** للخاصية **Show Panels** بالمجموعة **Behavior** بمربع الخصائص المصاحب لشريط المعلومات كى يتم عرض خانات الشريط، حيث يظهر عدد الخانات الافتراضى وهو خانة واحدة فقط. وكل خانة **Panel** من خانات الشريط عبارة عن كائن مستقل له خصائصه، ويجمع هذه الخانات (إذا احتوى الشريط على أكثر من خانة أى أصبح شريط متعدد الخانات) مجموعة **Collection** تسمى **Items**. بعد إضافتك لشريط المعلومات يمكنك البدء في إضافة الخانات وتحديد خواصها. لأداء ذلك، تابع معنا الخطوات الآتية:

١. من مربع الخصائص المصاحب لشريط المعلومات، انقر الزر ... المجاور للخاصية **Items**، تظهر نافذة **Items Collection Editor** خالية لا تحتوى على أية عناصر.

٢. يمكنك إضافة أربعة عناصر مختلفة إلى شريط الحالة، وهى العنوان StatusLabel (وهو العنصر الافتراضى فى أشرطة الحالة) وشريط التقـــدم ProgressBar وزر القائمة المنسدلة DropDownButton وزر الانسدال SplitButton.

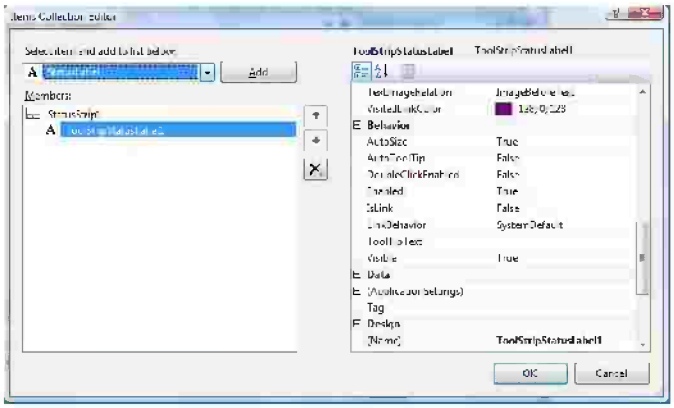
٣. اختر النوع المناسب للعنصر من القائمة المنسدلة Select Item and add to list below ثم انقـــر زر Add، يـــتم إضـــافة خانـــة جديـــدة باســـم ToolStripSplitLabel1 بالعمود الأيسر من النافذة بينما تظهر خصائص هذه الخانة بالعمود الأيمن (انظر شكل ١١-٤).

٤. قم بتعيين خصائص الخانة حسبما يروق لك وطبقاً للجدول ١١-٢ التالى والشرح التابع.

٥. قم بتكرار الخطوتين السابقتين لإضافة المزيد من الخانات إلى شريط المعلومات.

٦. إذا وجدت أنك لست فى حاجة إلى إحدى الخانات بعد إنشائها، قم باختيار الخانة من العمود الأيسر ثم انقر زر الحذف ✕ لحذفها.

٧. انقر زر Ok لإغلاق النافذة ولاحظ ظهور الخانات التى قمت بتعيينها داخل شريط المعلومات.



شكل ١١-٤ تعيين خانات شريط المعلومات

جدول ١١-٢ خصائص خانات شريط المعلومات

الخاصية	الاستخدام
Text	تعين النص الذى يظهر داخل الخانة
TextDirection	تعيين اتجاه النص، أفقى أم رأسى
TextAlign	تعيين محاذاة النص داخل الخانة بالاختيار من بين تسع قيم
AutoSize	التحجيم التلقائى للخانة داخل شريط المعلومات
BorderStyle	تعيين نمط إطار الخانة
Image	تعيين صورة تظهر داخل الخانة
DisplayStyle	تعيين نمط الخانة بإظهار النص فقط أو النص والصورة أو الصورة فقط
ToolTipText	تعيين نص التلميح الذى يظهر داخل مربع صغير عند مرور مؤشر الفأرة فوق الخانة
Width	تعيين عرض الخانة
Name	تعيين اسم الخانة

وفيما يلى نوضح بعض الخصائص الهامة الموجودة بالجدول السابق.

الخاصية *DisplayStyle*

تحدد هذه الخاصية محتويات الخانة، وتحتوى على أربع قيم، الأولى **None** وتعنى عدم ظهور أى بيانات بالخانة والثانية **Text** وتعنى ظهور النص فقط داخل الخانة والثالثة **Image** وتعنى ظهور الصورة فقط داخل الخانة والرابعة والأخيرة **imageAndText** وتعنى ظهور النص والصورة معاً داخل الخانة.

الخاصية *AutoSize*

تقوم هذه الخاصية بتحديد الحجم التلقائى للخانة داخل شريط المعلومات وتحتوى على إحدى قيمتين، القيمة **True** وتعنى التغير التلقائى لحجم الخانة تبعاً لمحتوياتها، والقيمة

False وتعنى الالتزام بالأبعاد التى يتم تحديدها.

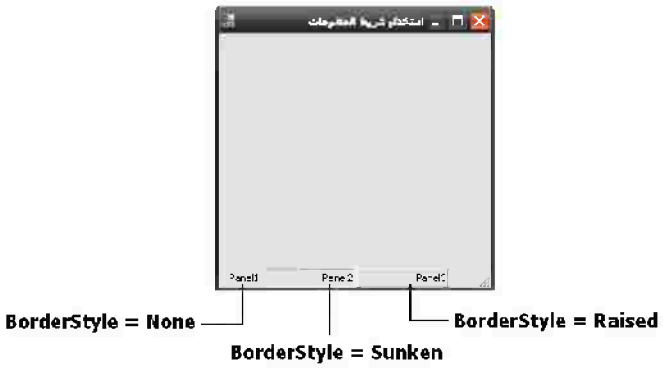
الخاصية *Spring*

وتحتوى هذه الخاصية على قيمتين، إما **True** إذا أردت امتداد الخانة لتملأ المساحة الخالية أو **False** إذا أردت عدم امتداد الخانة بغض النظر عن وجود مساحة خالية داخل الشريط. لاحظ أن قيمة هذه الخاصية والخاصية السابقة تؤثر على عرض الخانات، وليس عرض الشريط الذي سيمتد بعرض النافذة دائماً.

الخاصية *BorderStyle*

تقوم هذه الخاصية بتعيين نمط إطار الخانة من بين ما يقرب من عشر قيم مثل القيمة **Flat** المستخدمة فى إظهار الخانة مسطحة على شريط المعلومات (أى بلا إطار)، والقيمة **Sunken** المستخدمة فى إظهار الخانة محفورة داخل شريط المعلومات، والقيمة **Raised** المستخدمة فى إظهار الخانة بارزة على شريط المعلومات.

يوضح شكل ١١-٥ بعض قيم الخاصية **BorderStyle**.



شكل ١١-٥ تأثير القيم المختلفة للخاصية BorderStyle

التعامل مع خانات شريط المعلومات من خلال الكود

تعرفنا على كيفية إضافة الخانات إلى شريط المعلومات أثناء التصميم من خلال محرر الخانات. يتيح لك شريط المعلومات كذلك إضافة وإزالة الخانات أثناء التشغيل من خلال الوظائف **Add() , Remove()** الخاصة بالمجموعة **Items** التي ذكرناها سابقاً.

المثال التالي يقوم بإضافة خانة إلى شريط الأدوات، ثم يزيل الخانة الأولى:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
    Dim Newidx As Integer
    StatusStrip1.Items.Add("New")
    Newidx = StatusStrip1.Items.Count - 1
    With StatusStrip1.Items(Newidx)
      .DisplayStyle = ToolStripItemDisplayStyle.Text
      .ToolTipText = Format(Now(), "Long Date")
    End With
    StatusStrip1.Items.Remove(ToolStripStatusLabel1)

  End Sub
```

لاحظ في المثال السابق استخدامنا للخاصية **Count** وهي خاصية للمجموعات عمومـــاً، وهي تحتوي على عدد العناصر في المجموعة. قم بتشغيل التطبيق مرةً أخرى، تلاحظ إضافة خانة جديدة وحذف الخانة الأولى (انظر شكل ١١-٦).

شكل ١١-٦ إضافة الخانات وحذفها من خلال الكود.

يحتوى العنصر الأول داخل المجموعة **Items** على الدليل **0**، لذا قمنا فى الكود السابق بطرح العدد **1** من قيمة الخاصية **Count** للحصول على ترتيب العنصر الجديد داخل اللوحة.

الكتابة في شريط المعلومات أثناء عمل البرنامج

الوظيفة التي أنشأنا من أجلها شريط المعلومات هي إظهار معلومات للمستخدم، يتم ذلك أثناء تشغيل البرنامج من خلال تحديد قيمة الخاصية **Text** بالنص المراد إظهاره في شريط المعلومات. مثلاً لو أردنا في برنامج لمعالجة النصوص إن نعرض الصفحة الحالية وعدد الصفحات نستخدم

```
StatusStrip1.Items(1).Text = "Page 1 Of 10"
```

إظهار معدل إنجاز الأعمال بمقياس التقدم Progress Bar

عندما يحتوي برنامجك على وظائف تأخذ زمناً طويلاً نوعاً ما، ينبغي أن نخبر المستخدم بمقدار ما تم إنجازه من العملية وما تبقى منها، والأداة المناسبة لذلك هي مقياس التقدم **Progress Bar**. فمثلاً عند نسخ ملف في **Windows**، يظهر مستكشف **Windows** مقياساً يوضح عملية نسخ الملف ومقدار ما تم منها. هذا الأمر ذو أهمية مزدوجة للمستخدم:

الأهمية الأولى: أن يطمئن المستخدم أن البرنامج يعمل وليس في حالة عطل **Hang**.

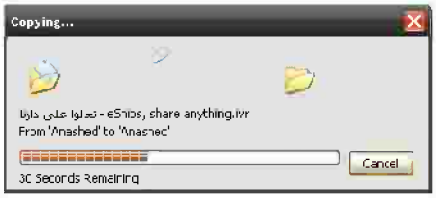
الأهمية الثانية: أن يعرف المستخدم ما تم وما تبقى من العملية، ليقرر (إن أمكن) إلغاؤها أو الاستمرار فيها.

إعداد مقياس التقدم

لا يختلف مقياس التقدم عن بقية الأدوات، كل ما عليك اختيار الأداة **ProgressBar** من شريط الأدوات ورسمها على النافذة أو النموذج.

في هذه الأداة بالذات تحمل الخاصية **Width** أهمية خاصة، حيث أن نسبة عرض الشريط إلى طوله تؤثر بشدة على مظهره القياسي. على أية حال تنصحك **Microsoft** أن يزيد عرض الشريط بمقدار ١٢ضعف عن طوله **Height** . هناك خاصية أخرى ذات أهمية كبرى وهي **Value** وهي تحدد المقدار الذي يتم الملء إليه (مثل الزئبق في الترمومتر حتى درجة الحرارة).

أيضاً الخاصيتين Minimum وMaximum لهما أهمية كبيرة، حيث يحددان الحدود التي تأخذها قيمة الخاصية Value. (لاحظ تشابه هذه الخصائص مع شريط التمرير Scroll Bar).



شكل ١١-٧ المظهر القياسي لمقياس التقدم كما يظهر في مربع نسخ الملفات

تعديل مقياس التقدم بالكود

يعتمد إظهار مدى اكتمال الوظائف على تغيير قيمة الخاصية Value. ويتم تغيير قيمة هذه الخاصية أثناء إجراء الوظيفة لتدل على مقدار ما تم منها. المثال التالي يوضح ذلك:

```
For i =1 To 100
        قراءة ملف قاعدة البيانات ...'
        ProgressBar1.Value = i
Next i
```

استخدام أداة التعقب TrackBar

من الأدوات الشائعة أيضاً في Windows أداة التعقب TrackBar التى كانت تعرف فى الإصدارات السابقة من Visual Basic بالشريط المنزلق Sliding Bar أو اختصاراً "المنزلق" Slider. تشبه هذه الأداة شريط التمرير إلى حدٍ بعيد في وظيفتها وإن اختلفت في شكلها.

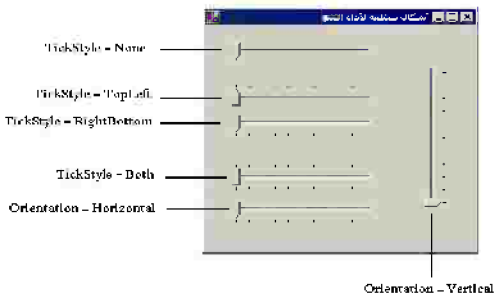
إعداد أداة التعقب

كما فعلنا مع الأدوات السابقة انقر أداة التعقب TrackBar من شريط الأدوات ثم ارسم الأداة على النموذج، تضاف الأداة بنفس الطريقة المعتادة. وبعد ذلك قم بتحديد قيم الخصائص التي تهمنا كما فى جدول ١١-٣.

جدول ١١-٣ خصائص أداة التعقب TrackBar

الخاصية	الاستخدام
Orientation	تعيين اتجاه أداة التعقب. فمن الممكن أن تكون الأداة أفقية (Horizontal) أو رأسية (Vertical)
SmallChange	تعيين مقدار تحرك رأس الأداة عند استخدام مفاتيح الأسهم الموجودة داخل لوحة المفاتيح
LargeChange	تعيين مقدار تحرك رأس الأداة عند استخدام الفأرة أو المفاتيح Page Up و Page Down
TickFrequency	تعيين المسافة بين كل علامة والعلامة التالية لها على التدريج
TickStyle	تعيين شكل التدريج. وتأخذ القيمة TopLeft لإظهار العلامات أعلى الأداة أو القيمة BottomRight لإظهار العلامات أسفل الأداة أو القيمة Both لإظهار العلامات أعلى وأسفل الأداة وأخيراً القيمة None التى تعنى عدم إظهار العلامات على الإطلاق
Minimum	تعيين أقل قيمة لموضع المؤشر على الأداة
Maximum	تعيين أكبر قيمة لموضع المؤشر على الأداة
Value	تعيين قيمة الأداة ومن ثمّ تعيين مكان المؤشر

يوضح الشكل ١١-٨ التالى تأثير القيم المختلفة لهذه الخصائص.



شكل ١١-٨ تأثير قيم الخصائص الهامة في المنزلق

كما قلنا يشبه هذا الشريط شريط التمرير **Scroll Bar** الذي شرحناه من قبل وتتوفر له الخصائص **LargeChange, SmallChange, Maximum, Minimum** وتستخدم كما في شريط التمرير تماماً.

مثال تطبيقي

سنقوم بإنشاء مثال بسيط على أداة التتبع يبين كيفية التعامل معها. تابع معنا الخطوات الآتية:

١. قم بإنشاء مشروع جديد باسم مناسب وليكن **TrackBar** ثم قم بإضافة أداة تتبع من شريط الأدوات إلى النموذج الموجود داخل المشروع.

٢. قم بتخصيص القيمة **Yes** للخاصية **RightToLeft** المصاحبة للنموذج وقم بتعيين عنوان مناسب للنموذج من خلال الخاصية **Caption**.

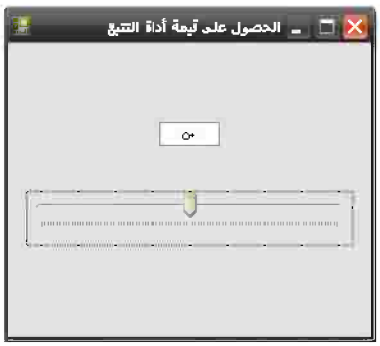
٣. حدد الخصائص **Minimum**، **Maximum** للأداة بـ **0**، **100** على الترتيب.

٤. قم أيضاً بإضافة مربع نص من شريط الأدوات. وتأكد من عدم وجود قيمة بالخاصية **Text** المصاحبة لمربع النص وإلا قم بحذفها.

٥. انقر أداة التتبع نقراً مزدوجاً ثم قم بإدخال الكود التالى إلى الإجراء الحدثي **Scroll** الخاص بأداة التتبع:

```
Private Sub TrackBar1_Scroll(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TrackBar1.Scroll
    TextBox1.Text = TrackBar1.Value
End Sub
```

٦. قم بتشغيل البرنامج بضغط **F5**، ثم استخدم الفأرة في تحريك الشريط يميناً أو يساراً لتشاهد تغيير قيمة **Value** ظاهرة في مربع النص (انظر شكل ١١-٩).



شكل ١١-٩ تتغير محتويات مربع النص بتغير مكان مؤشر أداة التتبع

استخدام أداة عرض القوائم Listview

من الأدوات التي شاع استخدامها مع Windows أداة عرض القوائم Listview، وأداة العرض الشجري Treeview واللتين تراهما في شكل ١١-١٠على نافذة Windows Explorer .وسنشرح أولاً أداة ListView ثم نشرح أداة TreeView.

شكل ١١-١٠عارض القوائم والعارض الشجري على نافذة مستكشف Windows

تشبه أداة عرض القوائم ListView مربع السرد Listbox مع بعض الاختلافات التي سنوضحها بعد قليل.

باستخدام عارض القوائم يمكنك عرض العناصر برموز صغيرة أو كبيرة أو كقائمة (تحتوي على رموز صغيرة) أو كجدول يحتوى على العديد من التفاصيل. هذه الخيارات هي التي تجدها في قائمة "عرض" View فى مستكشف Windows وهى (Large Icons, Small Icons, List, Details). ونوع طريقة العرض يتم تحديده بالخاصية View (أثناء التصميم أو تشغيل البرنامج).

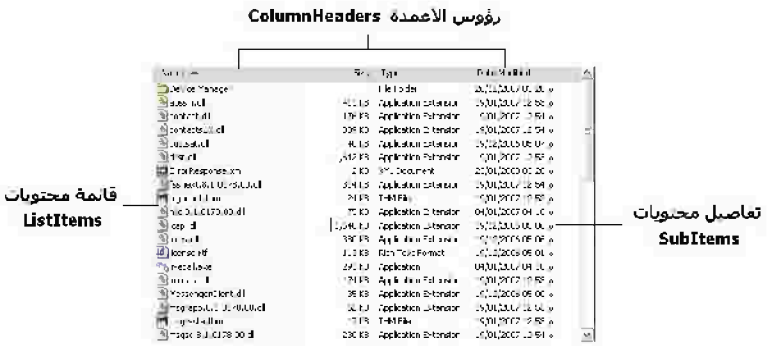
وبما أن هناك حجمان للرموز (صغير وكبير) فإن عارض القوائم يحتاج إلى قائمتي صور ImageList (كان شريط الأدوات يحتاج لواحدة فقط)، تحتوى قائمة منهما على الرموز الكبيرة والأخرى على الرموز الصغيرة. يتم تحديد ذلك بالخاصيتين SmallImageList, LargeImageList أثناء التصميم أو من خلال الكود كما في السطرين التاليين:

```
ListView1.LargeImageList = ImageList1
ListView1.SmallImageList = ImageList2
```

وإن لم تقم بتحديد قائمتي الصور أو إحداهما فإن الرموز التابعة لها لن تظهر.

يتم التعامل مع عارض القوائم من خلال عدة مجموعات Groups، المجموعة الأساسية هي Items وهي تشبه List في مربع السرد إلا أنها هنا لا تحتوي فقط على نص وإنما نص وإشارة للرموز المصاحبة في قوائم الصور الملحقة. أيضاً في حالة العرض بصورة التفاصيل نجد تعدد في البيانات المصاحبة لكل عنصر، تسمى هذه البيانات بالعناصر الفرعية SubItems، ويتم تخزينها في مجموعة باسم Subitems كل عنصر من عناصرها عبارة عن مصفوفة حرفية تحتوي على العناصر الفرعية. أيضاً نلاحظ وجود عناوين تعلو أعمدة البيانات الفرعية هذه العناوين تخزن في مجموعة تسمى Columnheaders.

هذه المجموعات المختلفة تظهر عناصرها مبينة بالشكل ١١-١١.



شكل ١١-١١شكل توضيحي للمجموعات المختلفة في عارض القوائم

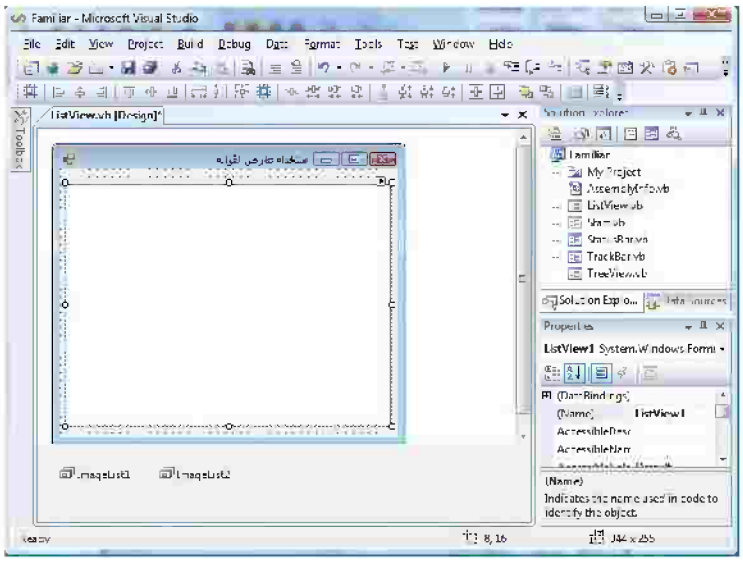
مثال تطبيقي

بدلاً من الإطالة في الشرح النظري لهذه الأداة سنقوم بإنشاء مثال تطبيقي بسيط نتعـــرض فيه للخطوط العامة في التعامل مع عارض القوائم. المثال سيكون عن قائمة بسيطة لمنتجات خاصة بالحاسب، يتم عرض اسمها ورمز لها (وثمنها فيما بعد). المشروع موجـــود علـــى القرص المدمج المرفق بالكتاب ويمكنك متابعته عند الحاجة لذلك.

تابع الخطوات التالية:

١. قم بإنشاء مشروع جديد باسم مناسب وليكن **ListView** وقم بتغيير اسم النافذة إلى **ListView** والعنوان إلى "استخدام عارض القوائم".

٢. أضف إلى النموذج أداتين من النوع قائمة صور **ImageList** وأداة عارض القوائم **Listview** وكبر حجمها لتبدو النافذة كما في الشكل ١١-١٢ التالى.



شكل ١١-١٢ نافذة البرنامج بعد إضافة العناصر الثلاثة لها.

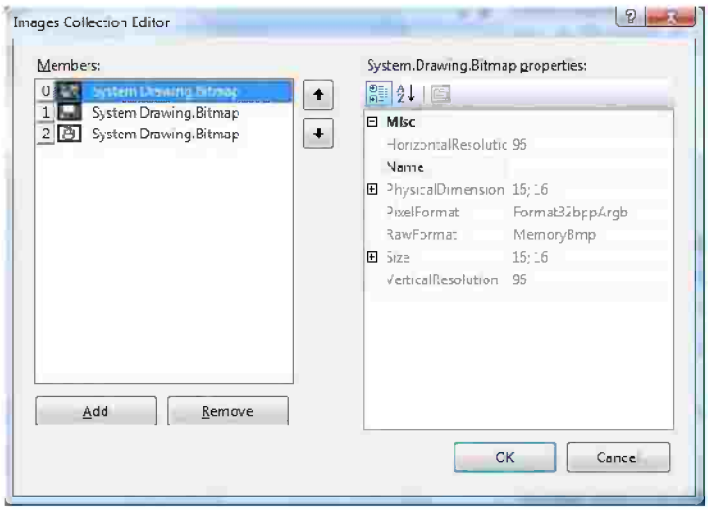
٣. الخطوة التالية أن نقوم بملء قائمتي الصور بالرموز التي نريدها حيث تحتوى القائمة الأولى **ImageList1** على الصور الكبيرة بينما تحتوى الثانية **ImageList2** على الصور الصغيرة. لتنفيذ ذلك اختر القائمة الأولى **Imagelist1** ثم انقر زر الفأرة الأيمن لإظهار القائمة الموضعية ومنها اختر **Properties** لفتح مربع الخصائص المصاحب للأداة.

٤. من مربع الخصائص، انقر الزر المجاور للخاصية **Images**، تظهر نافذة **Image Collection Editor** التى يتم من خلالها تعيين الصور المطلوبة.

٥. انقر زر **Add** ثلاث مرات لإضافة ثلاث صور إلى القائمة من المربع الحوارى الناتج (انظر شكل ١١-١٣).

٦. انقر زر **Ok** لإغلاق نافذة محرر الصور.

٧. كرر الخطوات الأربع السابقة لإضافة الرموز لقائمة الصور الثانية.



شكل ١١-١٣ إضافة ثلاث صور إلى قائمة الصور الأولى

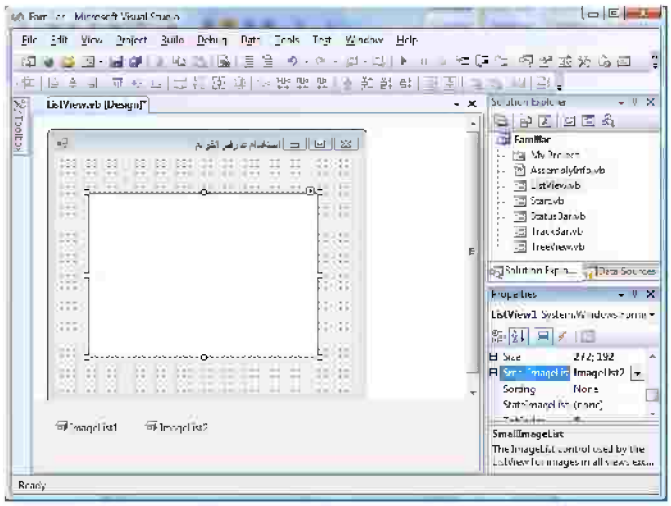
الربط بين عارض القوائم وقائمة الصور

أعددنا بذلك الرموز الكبيرة والصغيرة التي سنستخدمها. الخطوة التالية أن نقوم بالربط بين عارض القوائم وبين هذه الرموز ليتعامل معها. يتم ذلك من خلال ضبط الخاصيتين **LargeImageList** و**SmallImageList** والتي تتم إما من خلال الكود أو أثناء التصميم من خلال مربع الخصائص التابعة لعارض القوائم، لأداء ذلك، تابع معنا الخطوات الآتية:

١. قم بتنشيط عارض القوائم الموجود على النموذج.

٢. قم بتخصيص القيمة **ImageList1** للخاصية **LargeImageList**.

٣. قم بتخصيص القيمة **ImageList2** للخاصية **SmallImageList**.



شكل ١١-١٤تعيين قوائم الصور للخصائص LargeImageList و SmallImageList

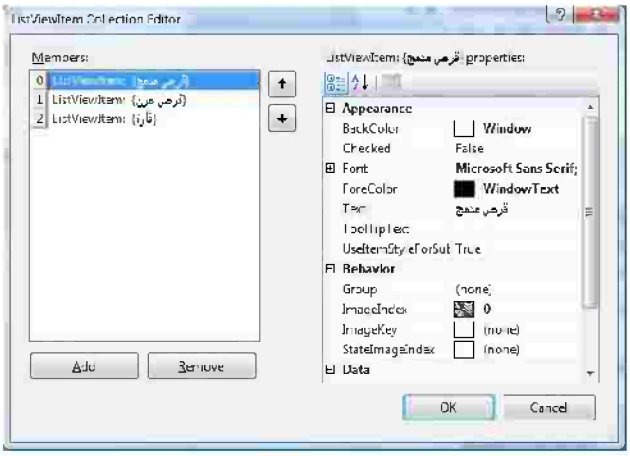
لقد أتممنا الآن كافة الإعدادات، يبقى أن نضيف العناصر إلى عارض القوائم. لأداء ذلك، تابع معنا الخطوات الآتية:

١. تأكد من تنشيط أداة عارض القوائم ثم انقر الزر المجاور للخاصية Items، تظهر نافذة ListViewItem Collection Editor.

٢. انقر زر Add، يتم إضافة عنصر جديد بالعمود الأيسر من النافذة.

٣. اختر الصورة التى ترغب فى إظهارها من الخاصية ImageIndex بالعمود الأيمن من النافذة وكذلك قم بتعيين نص الصورة داخل الخاصية Text.

٤. قم بتكرار الخطوتين السابقتين لإضافة بقية العناصر (انظر شكل ١١-١٥).



شكل ١١-١٥ إضافة العناصر إلى عارض القوائم

٥. قم بتشغيل التطبيق، تلاحظ ظهور العناصر التى قمت بتحديدها داخل عارض القوائم (انظر شكل ١١-١٦).

شكل ١١-١٦ ظهور العناصر داخل عارض القوائم

٦. يمكنك بدلاً من ذلك تعيين العناصر من خلال الكود. لأداء ذلك، قم بإدخال الكود التالى داخل الإجراء الحدثي Form_Load للنموذج:

```
ListView1.Items.Add("فأرة", 0)
ListView1.Items.Add("قرص مرن", 1)
```

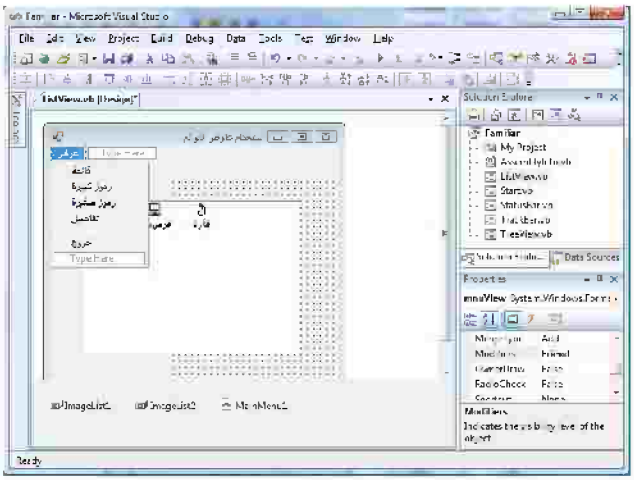
```
ListView1.Items.Add("قرص مدمج", 2)
```

هذه الأسطر تقوم بإضافة ثلاثة عناصر إلى العارض مع تحديد نص كل منها وترتيب الصورة المصاحبة داخل قائمة الصور.

وحتى الآن يمكن اعتبار البرنامج ناجحاً حيث أننا أضفنا قائمة الصور إلى عارض القوائم ، ولكننا نريد الاستفادة من الطرق المختلفة للعرض التي تتيحها هذه الأداة، لتغيير طريقة العرض سنضيف قائمة تسمح للمستخدم بذلك. بالطبع لن نعيد كيفية شرح عمل القائمة فقد قمنا بشرحها من قبل. قم بإضافة أداة قائمة إلى النموذج ثم قم بتعيين عناصرها كما فى الجدول ١١-٤ التالى (انظر شكل ١١-١٧).

جدول ١١-٤ عناصر قائمة النموذج

القائمة	الاسم
عرض	mnuView
قائمة	mnuList
رموز كبيرة	mnuLarge
رموز صغيرة	mnuSmall
تفاصيل	mnuDetails
-	mnuSeparator
خروج	mnuExit



شكل ١١-١٧ إضافة القائمة إلى النموذج

بعد إضافة هذه القوائم، قم بإضافة الكود التالي لكل عنصر من عناصر القائمة على حده:

```
Private Sub mnuList_Click(ByVal Sender As System.Object,
ByVal e As System.EventArgs) Handles mnuList.Click
    ListView1.View = View.List
End Sub

Private Sub mnuLarge_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuLarge.Click
    ListView1.View = View.LargeIcon
End Sub

Private Sub mnuSmall_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuSmall.Click
    ListView1.View = View.SmallIcon
End Sub

Private Sub mnuDetails_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuDetails.Click
    ListView1.View = View.Details
End Sub

Private Sub mnuExit_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuExit.Click
```

```
    End
End Sub
```

يمكنك الآن الضغط على F5 وتغيير طريقة العرض باستخدام القائمة، لإظهار الرموز على شكل قائمة مرة ورموز كبيرة مرة أخرى ورموز صغيرة مرة ثالثة. يظهر فى شكل ١١-١٨الرموز فى حالتين، حالة إظهار الرموز كقائمة وحالة إظهارها كرموز صغيرة.

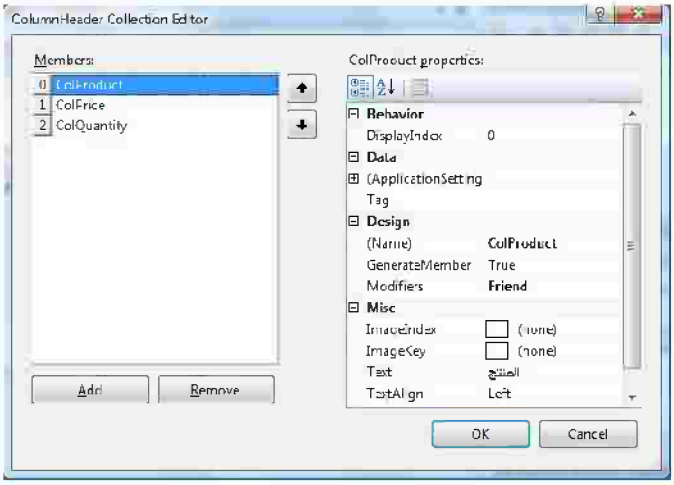
شكل ١١-١٨ استخدام قائمة الأوامر في عرض العناصر بعدة طرق في عارض القوائم

والآن ننتقل إلى طريقة العرض الأخيرة وهي العرض التفصيلي أو الجدولي، وهذا يحتاج إلى تحديد بيانات فرعية لكل عنصر داخل الخاصية SubItems. لنتمكن من العرض بهذه الطريقة، سنقوم بإضافة ثلاثة أعمدة لعارض القوائم. لأداء ذلك، تابع معنا الخطوات الآتية:

١. تأكد من تنشيط عارض القوائم داخل النموذج وتأكد كذلك من ظهور مربع الخصائص.
٢. انقر الزر المجاور للخاصية Columns، تظهر نافذة ColumnHeader Collection Editor التى يمكنك من خلالها تعيين رؤوس الأعمدة التى تظهر داخل عارض القوائم.
٣. انقر زر Add، يتم إضافة عمود جديد (انظر شكل ١١-١٩).
٤. قم بتغيير اسم العمود الجديد إلى اسم معبر وليكن ColProduct وذلك من الخاصية Name بالقسم الأيمن من النافذة. قم بتغيير عنوان العمود داخل الخاصية Text وليكن Product.

٥. قم بتكرار الخطوتين السابقتين لإضافة عمودين أحدهما باسم **ColPrice** وعنوان **Price** والآخر باسم **ColQuantity** وعنوان **Qty**.

٦. انقر زر **Ok** لإغلاق النافذة.



شكل ١١-١٩ تعيين رؤوس الأعمدة من خلال نافذة **ColumnHeader Collection Editor**.

قم بتشغيل التطبيق، ثم افتح قائمة "عرض" واختر "تفاصيل" من القائمة المنسدلة، يتم عرض عناصر القائمة فى صورة أعمدة، إلا أن العمودين الجديدين لا يحتويان على أية بيانات وهذا ما سنقوم بإضافته فى الخطوات التالية (انظر شكل ١١-٢٠).

شكل ١١-٢٠ عرض عناصر القائمة فى صورة أعمدة

لإضافة بيانات العمودين الثانى والثالث، تابع معنا الخطوات الآتية:

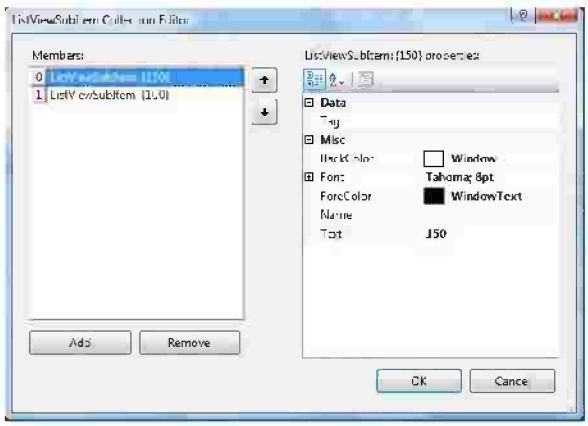
١. تأكد من تنشيط عارض القوائم داخل النموذج وتأكد كذلك من ظهور مربع الخصائص.

٢. انقر الزر المجاور للخاصية **Items**، تظهر نافذة **ListViewItem Collection Editor** التى تحتوى على عناصر القائمة والتى قمنا بإعدادها من قبل (راجع شكل ١١-١٥).

٣. انقر العنصر الأول من العمود الأيسر ثم انقر الزر المجاور للخاصية **SubItems** بالعمود الأيمن من النافذة، تظهر نافذة أخرى بعنوان **ListViewSubItem Collection Editor** (انظر شكل ١١-٢١).

٤. انقر زر **Add** لإضافة عمود جديد وقم بتغيير عنوانه إلى **150** وهو سعر القرص المدمج.

٥. انقر زر **Add** لإضافة عمود آخر وقم بتغيير عنوانه إلى **100** وهو كمية القرص المدمج الموجودة بالمخزن ثم انقر زر **Ok** لإغلاق النافذة الفرعية.



شكل ١١-٢١ إضافة قيم العناصر الفرعية

٦. قم بتكرار الخطوات ٣ و٤ و٥ مع العنصر الثانى والثالث داخل القائمة.

٧. انقر زر **Ok** لإغلاق النافذة.

قم الآن بتشغيل التطبيق ثم افتح قائمة "عرض" واختر "تفاصيل" من القائمة المنسدلة، يتم عرض عناصر القائمة فى صورة أعمدة بكامل بياناتها (انظر شكل ١١-٢٢).

شكل ١١-٢٢ عرض عناصر القائمة فى صورة أعمدة تحتوى على بيانات

يمكنك إنشاء الأعمدة والعناصر الفرعية باستخدام الكود كما يلى:

```
ListView1.Columns.Add("Price", 60, HorizontalAlignment.Center)
ListView1.Columns.Add("Qty", 60, HorizontalAlignment.Center)
ListView1.Items(0).SubItems.Add("110")
ListView1.Items(0).SubItems.Add("100")
ListView1.Items(1).SubItems.Add("2")
ListView1.Items(1).SubItems.Add("200")
ListView1.Items(2).SubItems.Add("10")
ListView1.Items(2).SubItems.Add("210")
```

كما ترى بدأنا بتحديد عناوين الأعمدة، ثم أضفنا بيانات العنصرين الثانى والثالث.

إلى هذا الحد كان كل همنا في "إخراج" أو إظهار بيانات وصور على الشاشة، هذه الصور تظهر في الحقيقة لهدف آخر، وهو تلقي و "إدخال" الأوامر من المستخدم، وكمثال على استخدام عناصر عارض الصور في الإدخال قم بكتابة الكود التالي في الإجراء الحدثي Click:

```
Private Sub ListView1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ListView1.Click
    MessageBox.Show("You Clicked Me")
End Sub
```

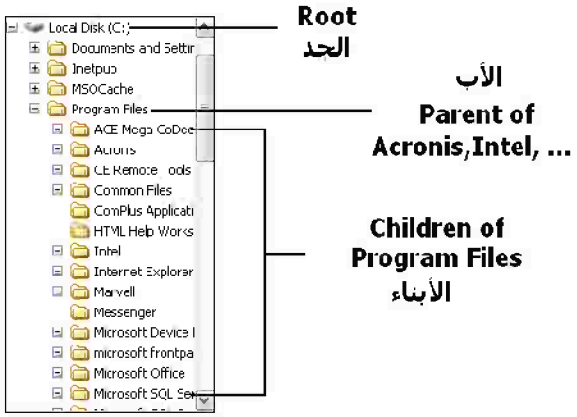
والمثال السابق عبارة عن مثال بسيط عن عارض الصور، وكما ترى أن أسلوبنا من بداية الكتاب يسير وفق " الأبسط هو الأفضل". والهدف من المثال هو عرض طريقة التشغيل بأوضح أسلوب ودون الدخول في أي تفاصيل فرعية، والتركيز على ما نريد تعلمه من المثال. وبالطبع هدفنا ليس استقصاء الأداة أو خصائصها ووظائفها فهذا أمر يطول شرحه خصوصاً مع العدد الكبير لهذه الخصائص.

إذا أردت التعرف على بقية الخصائص التى تخص الأداة، افتح مربع الخصائص وتعرف على الخصائص التى تشتمل عليها فى ضوء الشرح الذى تقدم عن الأدوات الأخرى.

وكما رأيت أن التركيز على كتابة كود لا يتجاوز سطر أو سطرين في معظم حالاته هو أفضل الوسائل لإيضاح الوظائف حيث أن الكثير منا يصيبه السأم إن لم يكن الإعراض التام عن الأمثلة التي يتجاوز طول كتابتها الصفحة الواحدة.

استخدام أداة العرض الشجري Tree View

تقوم أداة العرض الشجري **TreeView** بسرد عناصر مع رموز تماما كعارض القوائم إلا أنها تتميز بأنها تعرض العناصر بشكل هرمى، يتفرع إلى عدة مستويات وليس في مستوى واحد كما يفعل عارض القوائم. هذا الشكل الهرمى يسمى شجرة **Tree** لأنه يشبه الشجرة من حيث وجود جذر وفروع وفروع أخرى متفرعة عن هذه الفروع ... وهكذا. وهو يشبه أيضاً شجرة العائلة فكل عضو في الشجرة له أب **Parent** وهو النقطة التي تفرع منها، أيضا لبعض الأعضاء أبناء **Children** كما أن الأعضاء في نفس الفرع (ذات الأب الواحد) يمكنها الإشارة إلى بعضها من خلال الخاصيتين **NextNode** و**PreviousNode**. وهناك أيضاً جذر أو جد للشجرة يسمى **Root**. لتوضيح ذلك، انظر الشكل ١١-٢٣ الذى يحتوى على نموذج لأداة عرض شجري في برنامج **Windows Explorer**.



شكل ١١-٢٣ مثال لاستخدام أداة العرض الشجري في Windows Explorer

تقوم الأداة TreeView داخليا بحفظ جميع نقاطها، بغض النظر عن أماكنها وتفريعاتها في مجموعة تسمى Nodes يمكن الإضافة إليها والحذف منها تماماً كما فعلنا مع أداة عـــرض القوائم ListView. ولكن كما سترى يحتاج إضافة عنصر لأحد الفروع (تحت أحـــد النقاط) إلى تحديد هذه النقطة والتي تسمى Parent وهذا ما يتم بالفعل، فمثلاً لإضـــافة العنصر "Mohamed" مثلا إلى الوالد "Waleed" في أداة عرض شجري لعائلة، نستخدم السطر الآتي:

```
e.Node.Nodes.Add("Mohamed")
```

حيث يوجد العنصر الحالى (Waleed) داخل e.Node كما سنرى فى الكود بعد قليل.

لن نطيل في الشرح النظري لهذه الأداة، فهي تعتبر من أعقد الأدوات فى Visual Basic وشرحها النظري الطويل قد يسبب بعض الملل، لذا سنحاول فهمها واستخدامها عن طريق المثال التالي.

مثال تطبيقي

نوضح فيما يلى مثال تطبيقى لاستخدام أداة العرض الشجرى. المثال موجود على القرص المرفق بالكتاب ويمكنك تجربته فى حالة الضرورة.

تابع الخطوات التالية:

١. قم بإنشاء مشروع جديد باسم مناسب وليكن **TreeView** ثم قم بتغيير اسم النموذج إلى **TreeView** والعنوان إلى "استخدام أداة العرض الشجرى".

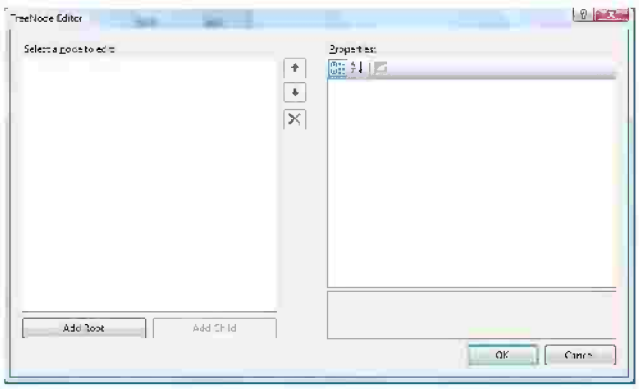
٢. قم بإضافة أداة عرض شجرى من شريط الأدوات إلى النموذج بنقر الأداة نقراً مزدوجاً.

٣. قم أيضاً بإضافة أداة قائمة صور **ImageList** إلى النموذج الجديد.

٤. تأكد من تنشيط أداة الصور ثم انقر الخاصية **Images** ثم قم بتعيين مجموعة من الصور كما تعلمت من قبل.

٥. قم بتنشيط أداة العرض الشجرى ثم قم بربط الأداة بقائمة الصور. لأداء ذلك، انقر الخاصية **ImageList** ثم قم باختيار القائمة التى قمت بإعدادها فى الخطوة السابقة.

٦. تأكد من تنشيط أداة العرض الشجرى ثم انقر الخاصية **Nodes** من مربع الخصائص، تظهر النافذة **TreeNode Editor** التى يمكنك من خلالها تعيين عناصر الشجرة (انظر شكل ١١-٢٤).



شكل ١١-٢٤ تعيين عناصر الشجرة من خلال نافذة **TreeNode Editor**.

٧. انقر زر **Add Root**، يتم إضافة عنصر أبوى جديد باسم **Node1** مع ظهور الصورة الأولى داخل قائمة الصور بجانب العنصر.

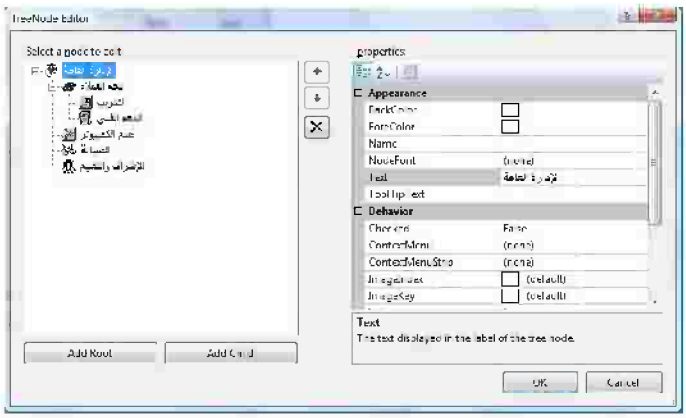
٨. قم بتغيير عنوان العنصر من مربع النص **Label**.

٩. يمكنك التحكم فى الصورة المصاحبة للعنصر من خلال مربعى السرد Image وSelectedImage حيث يتم تحديد الصورة التى تظهر فى حالة عدم اختيار العنصر داخل مربع السرد Image بينما يتم تحديد الصورة التى تظهر فى حالة عدم اختيار العنصر داخل مربع السرد SelectedImage.

١٠. لإضافة العناصر الفرعية إلى العنصر السابق، انقر زر Add Child ثم قم بتعديل عنوان العنصر كما سبق.

١١. لحذف أى عنصر، انقر زر Delete.

١٢. قم بتكرار الخطوات من ٦ إلى ٩ لإدخال جميع العناصر الأبوية والوليدة (انظر شكل ١١-٢٥).



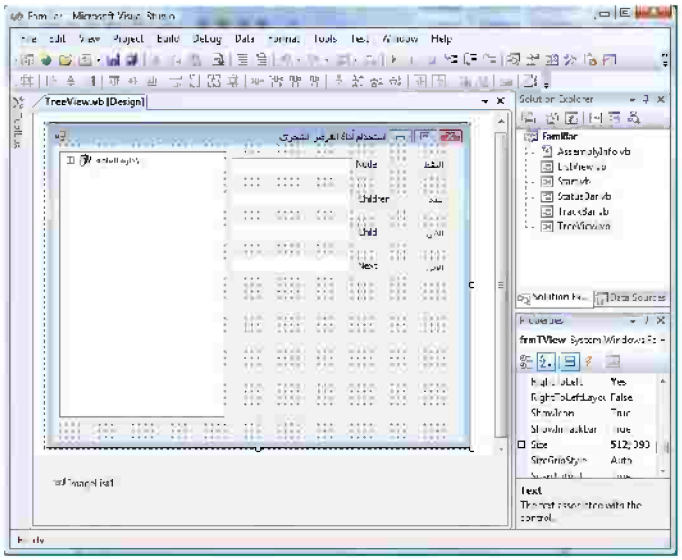
شكل ١١-٢٥ تعيين عناصر أداة العرض الشجرى.

١٣. انقر زر Ok لإغلاق نافذة TreeNode Editor.

يمكنك الآن تشغيل البرنامج ورؤية الشجرة داخل العارض (انظر شكل ١١-٢٦).

شكل ١١-٢٦ شجرة أو هيكل إحدى الشركات داخل العارض الشجري

لتوضيح بقية الخصائص المرتبطة بهذه الأداة سنحتاج إلى إضافة عدد من مربعات النصوص والعناوين كما ترى في شكل ١١-٢٧. قم بإضافة مربعات النصوص والعناوين إلى النموذج، مع تسمية مربعات النصوص TxtNode وTxtChildren وtxtChild وtxtNext على الترتيب ثم قم بإضافة الكود التالى إلى الحدث AfterSelect الخاص بأداة العرض الشجرى:



شكل ١١-٢٧ النموذج بعد إضافة مربعات النصوص

```
Private Sub TView_AfterSelect(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.TreeViewEventArgs)
Handles TView.AfterSelect
        txtNode.Text = e.Node.Text
        If e.Node.GetNodeCount(True) = 0 Then
            txtChildren.Text = "  لا يوجد "
        Else
            txtChildren.Text = e.Node.GetNodeCount(True)
        End If
        If e.Node.FirstNode Is Nothing Then
            txtChild.Text = "لا يوجد "

            txtNext.Text = "لا يوجد "
        Else
            txtChild.Text = e.Node.FirstNode.Text
            txtNext.Text = e.Node.FirstNode.NextNode.Text
        End If
    End Sub
```

لاحظ استخدامنا هنا لعدة خصائص جديدة منها FirstNode وNextNode وكـــذلك الوظيفة GetNodeCount التى تقوم بإرجاع عدد العناصر الموجودة أســـفل العنصـــر

الحالى. ومن الجدير بالذكر أن الخاصية **FirstNode** تعطي الابن الأول فقط، للوصول إلى الأبناء التالية إن وجدت نستخدم الخاصية **NextNode** وهي خاصية للابـــن و ليســـت للأب، وهي تعطي الأخ التالي الذي له نفس الأب. وقد اســـتخدمناها للابـــن الأول في **Node.FirstNode.NextNode**.

من أهم الأمور هنا أنك تجد عبارات **If** في وسط الكود في عدة مواضع، هذه العبـــارات ضرورية، وستحصل على خطأ إن لم تستخدمها، السبب في ذلك أن بعض الخصائص غير معرفة لبعض النقاط، مثلا **FirstNode** غير معرفة إذا كانت النقطة لـــيس لهـــا أبنـــاء **NodeCount = 0** كذلك الخاصية **NextNode** غير معرفة لآخر فرد في نفس الطبقـــة (التي لها نفس الأب). يجب دائما إجراء الاختبارات عند التعامل مع هذه الخصائص حتى لا يحدث خطأ في البرنامج.

بالإضافة إلى الخاصية **NextNode** هناك أيضاً الخاصية **PreviousNode** والتي تشير إلى الفرد أو الأخ السابق في نفس الطبقة.

يوضح الشكل ١١-٢٨ النموذج أثناء التشغيل، يمكنك إضافة المزيد من النقاط ومتابعـــة القيم التي يظهرها البرنامج.

شكل ١١-٢٨ شكل البرنامج أثناء عمله و يعرض عدد من خصائص كل نقطة

في الحقيقة هذه الأداة إمكانياتها كبيرة، ولكى تستطيع إتقان التعامل معها ينبغي أن تتمرن كثيراً وتقرأ ملفات المساعدة الخاصة بـ Visual Basic وتشاهد الأمثلة الملحقة باللغة. ولكن المثال السابق يكفي لكي تضع هذه الأداة في برنامجك وتستخدمها لعرض المعلومات.