

الفصل الرابع عشر إنشاء أدوات التحكم المخصصة

جميع أدوات التحكم التي استخدمناها في الفصول السابقة من الكتاب كانت من الأدوات المعرفة أساساً داخل اللغة. سنتعرف في هذا الفصل على كيفية إنشاء أدوات تحكم مخصصة وكيفية استخدام هذه الأدوات داخل النماذج بالإضافة إلى الطريقة البديلة لمصفوفات أدوات التحكم الموجودة داخل إصدارات **Visual Basic**.

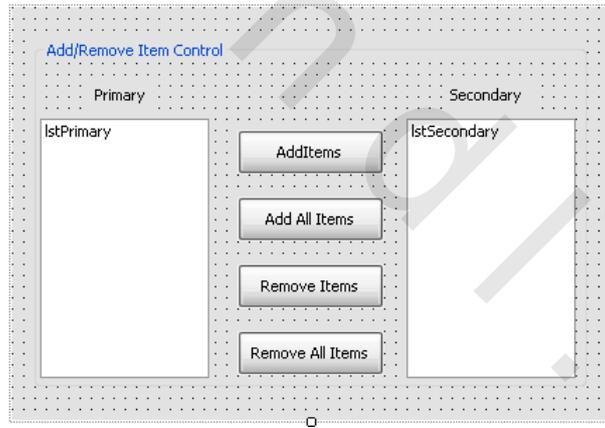
بانتهاء هذا الفصل سنتعرف على:

- ◆ أدوات التحكم المعرفة من قبل المستخدم
- ◆ استخدام أدوات التحكم المخصصة
- ◆ مصفوفات أدوات التحكم

أدوات التحكم المعرفة من قِبَل المستخدم

بعيداً عن أدوات التحكم المعرفة مسبقاً داخل المسميات الموجودة بالمسمى الرئيسي System، يمكنك إنشاء الأدوات الخاصة بك من خلال مصمم النوافذ Windows Forms Designer.

لتوضيح كيفية إنشاء أداة جديدة بمثال عملي، سنقوم فيما يلي بإنشاء أداة تحكم جديدة باسم AddRemove والتي تتكون من مربعي سرد، الأول يحتوي على القائمة الرئيسية بينما يحتوي الثاني على القائمة الثانوية. تقوم هذه الأداة بإضافة العناصر من القائمة الرئيسية إلى القائمة الثانوية أو حذفها من خلال أربعة أزرار تحكم، الأول لإضافة العناصر النشطة داخل القائمة الرئيسية إلى القائمة الثانوية والثاني لإضافة جميع العناصر، أما الثالث فيقوم بحذف العناصر النشطة من القائمة الثانوية بينما يقوم الرابع والأخير بحذف جميع العناصر الموجودة بهذه القائمة (انظر شكل ١٤-١).



شكل ١٤-١ الأداة الجديدة داخل نافذة مصمم النماذج.

لإنشاء الأداة الجديدة، تابع معنا الخطوات الآتية:

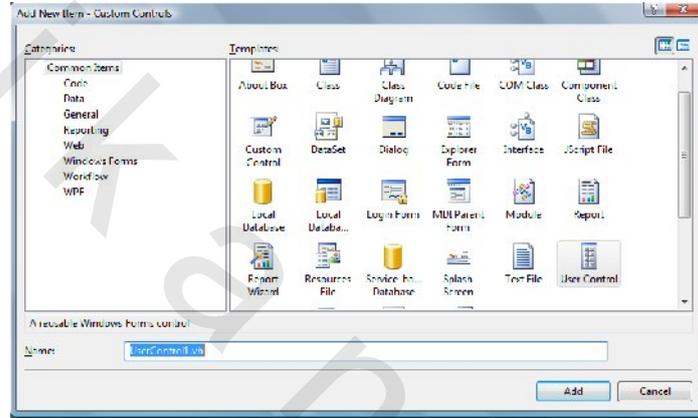
١. تأكد أنك داخل بيئة التطوير Visual Studio 2008 ثم افتح قائمة Project من شريط القوائم واختر Add User Control، يظهر المربع الحوار Add

New Item الذى يمكنك من خلاله إضافة العديد من العناصر الجديدة (انظر شكل ١٤-٢).

٢. تأكد من اختيار الرمز User Control بالمربع Templates الموجود يمين المربع

الحوارى ثم قم بتعيين اسم مميز للأداة الجديدة داخل مربع النص Name.

٣. انقر زر Open، يتم فتح نافذة لتصميم الأداة الجديدة.



شكل ١٤-٢ المربع الحوارى Add New Item.

٤. من مربع الأدوات قم بإضافة مربع مجموعة ومربعي سرد وأربعة أزرار تحكم

وعنوان لكل مربع سرد (راجع شكل ١٤-١) وبالإستعانة بجدول ١٤-١ التالى.

جدول ١٤-١ عناصر أداة التحكم المخصصة

| الاسم | العنصر |
|---------------|---------------------------------------|
| GroupBox1 | مربع المجموعة Add/Remove Item Control |
| IstPrimary | مربع السرد الأساسى |
| IstSecondary | مربع السرد الثانوى |
| Label1,Label2 | أداة عنوان |
| btnAddItems | زر الأمر Add Items |
| btnAddAll | زر الأمر Add All Items |

| العنصر | الاسم |
|---------------------------|----------------|
| زر الأمر Remove Items | btnRemoveItems |
| زر الأمر Remove All Items | btnRemoveAll |

٥. قم بتخصيص القيمة MultiExtended للخاصية SelectionMode لكل من مربعي السرد حتى تتمكن من اختيار أكثر من عنصر.

٦. قم بعد ذلك بكتابة كود أحداث احتواء حدث نقر الأزرار الأربعة، فمثلاً تحتوى دالة احتواء حدث نقر الزر btnAddItems الذى يقوم بإضافة العناصر النشطة فقط من القائمة الرئيسية إلى القائمة الثانوية على الكود التالى:

```
Private Sub btnAddItems_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAddItems.Click
Dim i, j As Integer
For i = 0 To lstPrimary.SelectedItems.Count - 1
    lstSecondary.Items.Add(lstPrimary.SelectedItems(i))
Next i
For j = lstPrimary.SelectedIndices.Count - 1 To 0 Step -1
    lstPrimary.Items.Remove(lstPrimary.SelectedItems(j))
Next j
UseControl()
End Sub
```

حيث تقوم الدوارة for الأولى بنسخ العناصر المختارة من القائمة الرئيسية إلى القائمة الثانوية بينما تقوم الدوارة for الثانية بحذف هذه العناصر من القائمة الرئيسية. وبعد ذلك يتم استدعاء الوظيفة UseControl() التى سنتعرض لها بعد قليل والتي تستخدم لتمكين الأزرار أو تعطيلها تبعاً لمحتويات القائمتين.

٧. قم بعد ذلك بإنشاء الوظيفة UseControl() بإدخال الكود التالى داخل التصنيف الرئيسى للأداة الجديدة:

```
Public Sub UseControl()
    If (lstPrimary.Items.Count > 0) Then
        Me.btnAddAll.Enabled = True
        Me.btnAddItems.Enabled = True
    Else
```

```
Me.btnAddAll.Enabled = False
Me.btnAddItems.Enabled = False
End If
If (IstSecondary.Items.Count > 0) Then
    Me.btnRemoveAll.Enabled = True
    Me.btnRemoveItems.Enabled = True
Else
    Me.btnRemoveAll.Enabled = False
    Me.btnRemoveItems.Enabled = False
End If
End Sub
```

حيث تقوم الوظيفة كما ترى بتمكين الأزرار أو تعطيلها طبقاً لعدد العناصر الموجودة داخل كل قائمة وكذلك عدد العناصر النشطة في كل منها.

٨. يمكنك بنفس الطريقة كتابة كود الأزرار الثلاثة الأخرى كما يلي:

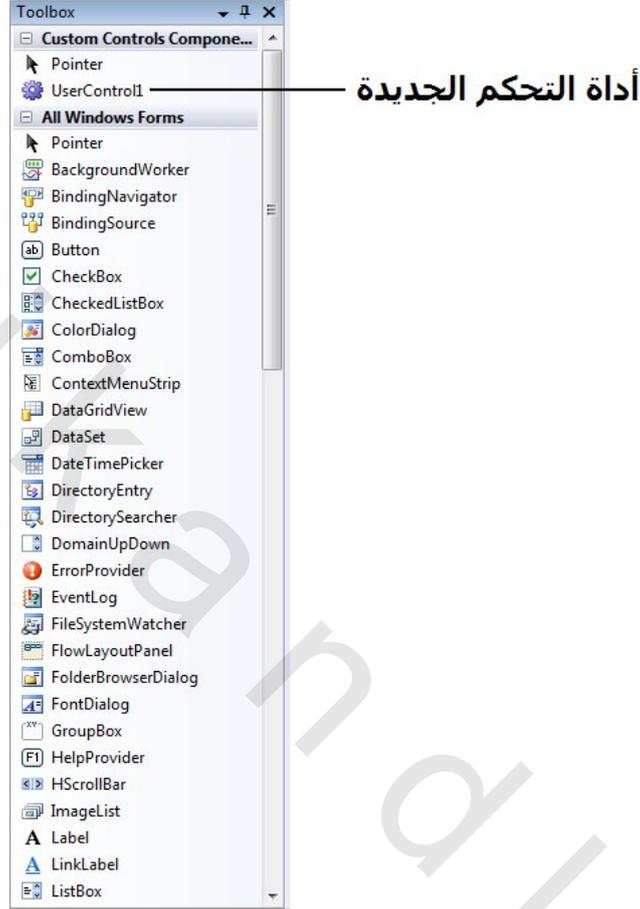
```
Private Sub btnAddAll_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAddAll.Click
    Dim i, j As Integer
    For i = 0 To IstPrimary.Items.Count - 1
        IstSecondary.Items.Add(IstPrimary.Items(i))
    Next i
    For j = IstPrimary.Items.Count - 1 To 0 Step -1
        IstPrimary.Items.Remove(IstPrimary.Items(j))
    Next j
    UseControl()
End Sub
```

```
Private Sub btnRemoveItems_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnRemoveItems.Click
    Dim i, j As Integer
    For i = 0 To IstSecondary.SelectedItems.Count - 1
        IstPrimary.Items.Add(IstSecondary.SelectedItems(i))
    Next i
    For j = IstSecondary.SelectedItems.Count - 1 To 0 Step -1
        IstSecondary.Items.Remove(IstSecondary.SelectedItems(j))
    Next j
    UseControl()
```

End Sub

```
Private Sub btnRemoveAll_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnRemoveAll.Click
    Dim i, j As Integer
    For i = 0 To lstSecondary.Items.Count - 1
        lstPrimary.Items.Add(lstSecondary.Items(i))
    Next i
    For j = lstSecondary.Items.Count - 1 To 0 Step -1
        lstSecondary.Items.Remove(lstSecondary.Items(j))
    Next j
    UseControl()
End Sub
```

٩. بمجرد الانتهاء من إدخال الكود، قم بترجمة الأداة الجديدة كى يقوم مصمم النماذج تلقائياً بإضافتها إلى مجموعة أدوات التحكم الموجودة بمربع الأدوات بحيث يمكنك استخدامها فى أى وقت داخل نماذجك (انظر شكل ١٤-٣). لأداء ذلك، افتح قائمة **Build** من شريط القوائم ثم اختر **Build Custom Controls** من القائمة المنسدلة.

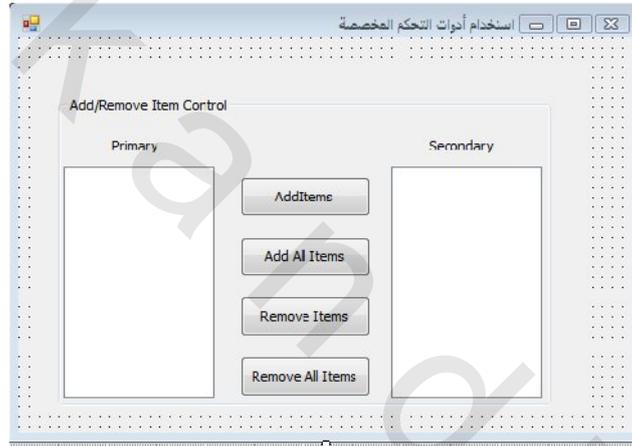


شكل ١٤-٣ إضافة الأداة لمربع الأدوات

استخدام أدوات التحكم المخصصة

بعد أن قمت بإنشاء أداة التحكم المخصصة في الفقرة السابقة، يمكنك استخدام هذه الأداة في أي نموذج داخل المشروع الحالي، كما يمكنك إحضارها إلى أي مشروع آخر. سنتعرف فيما يلي على كيفية استخدام الأداة داخل نموذج جديد وكيفية التحكم في خصائصها من داخل هذا النموذج. لأداء ذلك، تابع معنا الخطوات الآتية:

١. حتى الآن لم نستخدم النموذج Form1 الذي يتم إنشاؤه تلقائياً بمجرد إنشاء المشروع. قم بتنشيط هذا النموذج من داخل نافذة المستكشف.
٢. قم بإضافة الأداة الجديدة UserControl1 من مربع الأدوات إلى النموذج (انظر شكل ٤-١٤).
٣. كي تتمكن من اختبار عمل الأداة سنقوم بإضافة بعض العناصر إلى مربع السرد الرئيسي من خلال الكود أثناء تحميل النموذج. قم بإدخال الكود التالي داخل حدث تحميل النموذج Form1_Load:



شكل ٤-١٤ إضافة أداة التحكم إلى النموذج

```
Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
Me.UserControl1.IstPrimary.Items.Add("Ali")
Me.UserControl1.IstPrimary.Items.Add("Mohamed")
Me.UserControl1.IstPrimary.Items.Add("Waleed")
Me.UserControl1.IstPrimary.Items.Add("Ahmed")
Me.UserControl1.IstPrimary.Items.Add("Ramy")
Me.UserControl1.IstPrimary.Items.Add("Magdy")
Me.UserControl1.IstPrimary.Items.Add("Samir")
End Sub
```

وكما ترى يتم استخدام تقنية التسمية المنقوطة للوصول إلى مربع السرد IstPrimary وإضافة العناصر إليه من خلال الوظيفة Add().

الفصل الرابع عشر : إنشاء أدوات التحكم المخصصة

٤. والآن قم بتشغيل المشروع، تحصل على أداة التحكم داخل النموذج وبها العناصر التي قمنا بتحديدتها داخل الكود (انظر شكل ١٤-٥).
٥. لتجربة عمل الأداة، قم بتحديد عنصر أو أكثر داخل مربع السرد الرئيسي ثم انقر زر **Add Items** يتم حذف العناصر من مربع السرد الرئيسي وإضافتها إلى مربع السرد الثانوي.



- شكل ١٤-٥ تظهر أداة التحكم داخل النموذج وبها العناصر التي قمنا بتحديدتها داخل الكود
٦. قم باختيار هذه العناصر مرة أخرى ثم انقر زر **Remove Items**، يتم إعادتها مرة أخرى إلى مربع السرد الأساسي.
 ٧. انقر زر **Add All Items**، يتم نقل جميع عناصر مربع السرد الرئيسي إلى مربع السرد الثانوي، كما يتم تعطيل زر **Add Items** و **Add All Items** لعدم وجود عناصر أساساً داخل مربع السرد الرئيسي (انظر شكل ١٤-٦).



شكل ١٤-٦ نقل جميع العناصر من المربع الرئيسي إلى المربع الثانوي

٨. اختر عنصراً أو أكثر داخل مربع السرد الثانوي ثم انقر زر **Remove Items** يتم نقل العناصر المختارة إلى مربع السرد الرئيسي (انظر شكل ١٤-٧).
٩. بنفس الطريقة انقر زر **Remove All Items**، يتم نقل جميع عناصر مربع السرد الثانوي إلى مربع السرد الرئيسي مرةً أخرى، كما يتم تعطيل زرى **Remove Items** و **Remove All Items** لعدم وجود عناصر أساساً داخل مربع السرد الثانوي.



شكل ١٤-٧ نقل بعض العناصر إلى مربع السرد الرئيسي

يمكنك تعيين عناصر مربع السرد أثناء تصميم الأداة نفسها عن طريق تخصيص هذه العناصر للخاصية Items كما يمكنك استخدام مصدر للبيانات سواءً أثناء التصميم من خلال الخاصية DataSource أو أثناء التشغيل من خلال الكود.



مصفوفات أدوات التحكم

إذا كنت عزيزي القارئ من مستخدمي الإصدارات السابقة من Visual Basic (وتحديداً الإصدارات التي تسبق Visual Basic.NET)، فلعلك تعلم جيداً مفهوم مصفوفات أدوات التحكم. وهو ببساطة شديدة وضع مجموعة من أدوات التحكم المتشابهة (أى من نفس النوع) داخل مصفوفة واحدة بحيث يمكنك إجراء العمليات والوظائف وتنفيذ الأحداث على هذه المصفوفة مجتمعة. وكان ذلك يتم من خلال استخدام نفس الاسم لجميع عناصر المصفوفة مع تخصيص دليل Index يميز كل عنصر من هذه العناصر عن العناصر الأخرى. فمثلاً يمكنك إنشاء مصفوفة تحتوي على ثلاثة مربعات نصوص باسم TextBox على أن يكون الأول باسم TextBox(1) والثاني باسم TextBox(2) والثالث باسم TextBox(3) حيث يتم تعيين الاسم TextBox من خلال الخاصية Name والدليل المستخدم من خلال الخاصية Index.

وبدءاً من Visual Basic.Net لم يعد هناك تدعيم لخاصية تجميع الكائنات المتشابهة داخل مصفوفات، وذلك لأن الهدف من استخدام هذه المصفوفات هو مشاركة الأحداث وهو ما أصبح متاحاً الآن داخل Visual Basic 2005 ، ليس فقط مع أدوات التحكم المتشابهة ذات نفس النوع وإنما مع الأنواع المختلفة. فمثلاً ربما تقوم بإضافة زرى أمر Button باسم Button1 وButton2 ومربع اختيار CheckBox باسم CheckBox1 إلى أحد النماذج ثم تقوم بتعيين معالج حدث نقر هذه العناصر مجتمعة كما يلي:

```
Private Sub MixedControls_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click, Button2.Click, CheckBox1.Click
```

يتم هنا كتابة كود الأدوات مجمعة

End Sub

وكما ذكرنا أنك في **Visual Basic 6.0** مثلاً تستطيع استخدام الأداة من خلال دليلها المتمثل في الخاصية **Index**. وعلى الرغم من أن أدوات التحكم الموجودة في **Visual Basic 2005** لا تحتوي على الخاصية **Index**، إلا أنك تستطيع أداء نفس الوظيفة من خلال أي من الخصائص الأخرى مثل الخاصية **TabIndex** أو الخاصية **Tag**. فعلى سبيل المثال يمكنك تعيين الخاصية **TabIndex** لمجموعة من أدوات التحكم ثم استخدام هذه الخاصية داخل عبارة **Select Case** كما يلي:

```
Private Sub MixedControls_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click, Button2.Click, CheckBox1.Click
    Select Case sender.TabIndex
        Case 0
            MessageBox.Show("Button 1")
        Case 1
            MessageBox.Show("Button 2")
        Case 2
            MessageBox.Show("CheckBox 1")
    End Select
End Sub
```

