

الفصل السابع استخدام أدوات السرد List Controls

تعد طريقة تمثيل البيانات داخل البرنامج عامل أساسي في تقبل المستخدم للبرنامج وتفهمه لخطواته وطريقة عمله، لذا يجب أن نختار الأداة المناسبة لتمثيل كل جزء من أجزاء البرنامج. سنقوم في هذا الفصل بالتعرف على مجموعة الأدوات المستخدمة لحصر مجموعة من العناصر.

بانتهاه هذا الفصل ستتعرف على:

- ◆ عرض قوائم البيانات داخل المربعات الحوارية.
- ◆ ترتيب البيانات داخل شجرة البيانات.
- ◆ اختيار أكثر من عنصر داخل القائمة.
- ◆ الربط بين القوائم المختلفة داخل المربع الحوارى.

عند العمل مع برامج النوافذ، تلاحظ أنه كثيراً ما يتم تمثيل البيانات في صورة قوائم مثل المجلدات والملفات وأسماء الخطوط والأشكال. ولأن الطريقة المثلى لتمثيل البيانات في صورة قائمة تعتمد أساساً على نوعية هذه البيانات وعلى ما إذا كنت ستختار عنصراً واحداً فقط أم أكثر من عنصر من هذه القائمة، لذا هناك أربعة أنواع من أدوات السرد، لكل منها الخصائص والأشكال التي تميزه عن باقي الأنواع، وهي مربع السرد والتحرير Combo Box ومربع السرد List Box وشجرة البيانات Tree وأخيراً أداة القائمة List Control. وفي هذا الفصل سنقوم بالتعرف على كيفية إضافة هذه الأنواع والفرق بينها وكيفية عملها داخل البرنامج مع أدوات التحكم الأخرى. قم بإنشاء مشروع حوارى جديد باسم Lists بنفس الطريقة التي ذكرناها في الفصول السابقة لاستخدامه في شرح محتويات هذا الفصل.

استخدام مربع السرد والتحرير

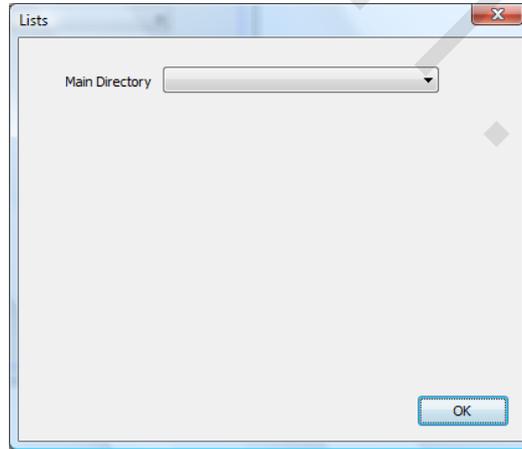
تحتوى هذه الأداة أساساً على ثلاث أدوات هي مربع نص ومربع سرد وزر أمر، لذا أطلق عليها اسم مربع السرد والتحرير Combo Box والذي يقوم بعرض قائمة من الخيارات بحيث يمكنك اختيار عنصر واحد فقط منها، حيث تتميز عن باقي أدوات السرد في ظهور العنصر المختار دائماً أعلى المربع.

يوجد ثلاثة أنواع من مربعات السرد والتحرير كما في جدول ٧-١، حيث يتم التحكم في هذه الأنواع من خلال الخاصية Type بمربع الخصائص المصاحب لمربع السرد والتحرير.

جدول ٧-١ أنواع مربع السرد والتحرير

المواصفات	النوع
يحتوى على مربع نص ومربع قائمة	Simple
يحتوى على مربع نص وزر ومربع قائمة	Dropdown
يحتوى على أداة عنوان وزر ومربع قائمة	Drop List

- لإضافة مربع السرد والتحرير إلى المربع الحوارى **Lists**، تابع معنا الخطوات الآتية:
١. قم بفتح قالب المربع الحوارى **IDD_LISTS_DIALOG** داخل نافذة محرر الموارد ثم قم بحذف النص الموجود بالمربع الحوارى.
 ٢. قم أيضاً بحذف زر **Cancel** وانقل زر **Ok** من مكانه إلى الركن الأيمن السفلى من المربع الحوارى.
 ٣. اختر رمز أداة العنوان من مربع الأدوات ثم قم بإضافة عنوان إلى الركن الأيسر العلوى من المربع الحوارى. قم بتغيير عنوان الأداة إلى **Main Directory**.
 ٤. اختر رمز مربع السرد والتحرير **ComboBox** من مربع الأدوات ثم قم بإضافة المربع يمين العنوان **Main Directory**. قم بزيادة طول مربع السرد والتحرير إلى يمين المربع الحوارى.
 ٥. قم بإدخال اسم مربع السرد والتحرير إلى الخاصية **ID** وليكن **IDC_MAIN_DIR**.
 ٦. اختر **Drop List** من الخاصية **Type** بمربع الخصائص. يجب أن يظهر المربع الحوارى الآن كما فى شكل ٧-١.



شكل ٧-١ إضافة مربع السرد والتحرير إلى المربع الحوارى

بمجرد إضافة مربع السرد والتحرير إلى المربع الحوارى، تلاحظ تخصيص القيمة **True** للخاصية **Sort** داخل مربع الخصائص تلقائياً، وهذا يعنى أنه سيتم ترتيب عناصر الأداة ترتيباً أبجدياً. فإذا لم يرق لك ذلك، قم بتخصيص القيمة **False** للخاصية **Sort** من مربع الخصائص.

قد لا يكون حجم القائمة المنسدلة كافٍ لعرض البيانات كاملة. لتغيير حجم مربع السرد والتحرير، انقر السهم الموجود يمين المربع، يظهر إطار يبين حجم القائمة المنسدلة. قم بإعادة تحجيم الأداة باستخدام أزرار التحجيم.



كما تعودنا في الفصول السابقة، للعمل مع أى أداة تحكم يجب ربطها بتغيير يحتوى على قيمة الأداة. لربط مربع السرد والتحرير الذى أضفناه منذ قليل بتغيير، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.
٢. انقر التصنيف **CListsDlg** بزر الفأرة الأيمن ثم اختر **Add > Add Variable** من القائمة الموضوعية، يظهر معالج إضافة متغير جديد.
٣. قم بتنشيط مربع الاختيار **Control variable** لأننا نرغب فى إنشاء متغير مرتبط بإحدى أدوات التحكم الموجودة بالمربع الحوارى.
٤. اختر أداة مربع السرد **IDC_MAIN_DIR** من مربع السرد **Control ID**.
٥. تأكد من اختيار **Control** من مربع السرد **Category** لأننا نحتاج إلى التعامل مع الخصائص الأخرى المصاحبة لهذه الأداة.
٦. تأكد من اختيار نوع البيانات **CComboBox** بمربع السرد والتحرير **Variable Type**.
٧. اكتب اسم المتغير فى خانة **Variable name** وليكن **m_cbMainDir** واختر درجة المتغير من مربع السرد **Access** وهو **Public** فى هذه الحالة.
٨. انقر زر **Finish** لإغلاق نافذة المعالج.

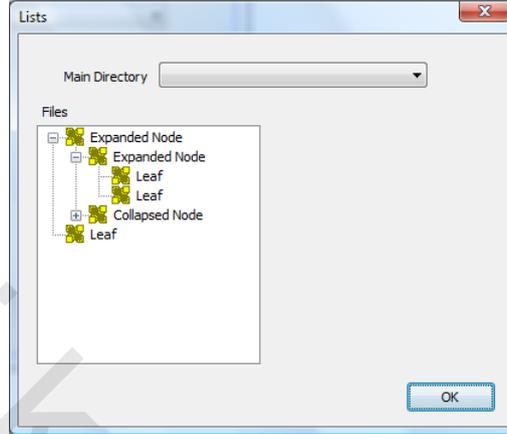
تحتوى لغة **Visual C++** على مربع سرد وتحرير محسّن يسمى **Extended ComboBox** يختلف عن مربع السرد والتحرير المعتاد في إمكانية إضافة الصور. سنتعرض بالتفصيل لهذه الأداة في مكان آخر من الكتاب.



استخدام شجرة البيانات

تتميز شجرة البيانات بعرض العناصر في صورة هرمية تبدأ من اليسار متجهةً إلى اليمين. يوجد العنصر الرئيسى للشجرة في أقصى اليسار ويسمى **Root node**. كما يوجد العنصر الأخير الذى لا يحتوى على عناصر فرعية في أقصى اليمين ويسمى **Leaf node**، بينما تسمى العناصر الموجودة بين العنصر الأول والعنصر الأخير **Branch nodes**. يمكنك افتراضياً اختيار عنصر واحد فقط من الشجرة، فإذا أردت اختيار أكثر من عنصر في نفس الوقت، قم بكتابة الكود اللازم لذلك كما سنرى في الجزء الثانى من هذا الفصل. لإضافة شجرة بيانات إلى المربع الحوارى **Lists**، تابع معنا الخطوات الآتية:

١. قم بفتح قالب المربع الحوارى **IDC_LISTS_DIALOG** داخل نافذة محرر الموارد.
٢. اختر رمز أداة العنوان من مربع الأدوات ثم قم بإضافة عنوان للشجرة أسفل العنوان **Main Directory** وقم بتغيير العنوان إلى **Files** (انظر شكل ٧-٢).
٣. اختر رمز شجرة البيانات **Tree Control** من مربع الأدوات ثم قم بإضافة شجرة يسار المربع الحوارى وأسفل **Files**.
٤. قم بإدخال اسم الشجرة إلى الخاصية **ID** بمربع الخصائص وليكن **IDC_FILES_TREE**.
٥. قم بتخصيص القيمة **True** للخصائص **Has lines** و **Has buttons** و **Lines at root** و **Scroll** داخل مربع الخصائص. يجب أن يظهر المربع الحوارى الآن كما في شكل ٧-٢.



شكل ٧-٢ إضافة شجرة البيانات إلى المربع الحوارى

لربط شجرة البيانات بمتغير، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.
٢. انقر التصنيف **CListsDlg** بزر الفأرة الأيمن ثم اختر **Add Variable**.
٣. من القائمة الموضوعية، يظهر معالج إضافة متغير جديد.
٤. قم بتنشيط مربع الاختيار **Control variable** لأننا نرغب فى إنشاء متغير مرتبط بإحدى أدوات التحكم الموجودة بالمربع الحوارى.
٥. اختر أداة شجرة البيانات **IDC_FILES_TREE** من مربع السرد **Control ID**.
٦. تأكد من اختيار **Control** من مربع السرد **Category** لأننا نحتاج إلى التعامل مع الخصائص الأخرى المصاحبة لهذه الأداة.
٧. تأكد من اختيار نوع البيانات **CTreeCtrl** بمربع السرد والتحرير **Variable Type**.
٨. اكتب اسم المتغير فى خانة **Variable name** وليكن **m_treeFiles** واختر درجة المتغير من مربع السرد **Access** وهو **Public** فى هذه الحالة.
٩. انقر زر **Finish** لإغلاق نافذة المعالج.

استخدام مربع السرد

يشتمل مربع السرد List Box في أبسط صورته على قائمة من العناصر، إلا أنها؛ بعكس شجرة البيانات أو مربع السرد والتحرير، تتيح للمستخدم اختيار أكثر من عنصر داخل القائمة. يوجد أربعة أنواع من الخيارات المتاحة عند العمل مع مربع السرد كما بالجدول ٧-٢ التالي والتي يمكنك اختيار أي منها من الخاصية Selection بمربع الخصائص.

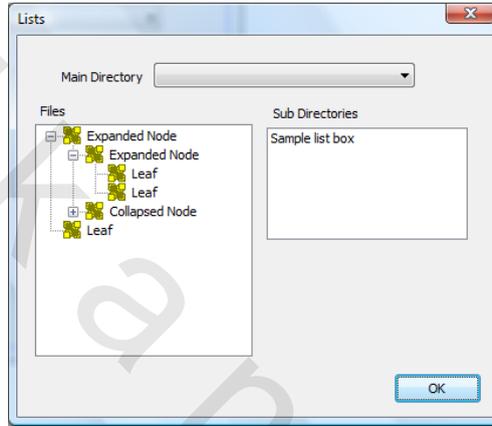
جدول ٧-٢ أنواع مربع السرد

النوع	المواصفات
Single	يتيح اختيار عنصر واحد فقط
Multiple	يتيح اختيار أكثر من عنصر باستخدام الفأرة ومفتاح Ctrl أو Shift
Extended	يتيح اختيار أكثر من عنصر باستخدام الفأرة فقط
None	لا يتيح اختيار أي عنصر

كما في مربع السرد والتحرير، بمجرد إضافة مربع سرد إلى المربع الحوارى، يتم تخصيص القيمة True للخاصية Sort تلقائياً، وهذا يعنى ترتيب عناصر مربع السرد ترتيباً أبجدياً. فإذا لم يرق لك ذلك، يمكنك تخصيص القيمة True لهذه الخاصية. لإضافة مربع سرد إلى المربع الحوارى، تابع معنا الخطوات الآتية:

١. قم بفتح قالب المربع الحوارى IDD_LISTS_DIALOG داخل نافذة محرر الموارد.
٢. اختر رمز أداة العنوان من مربع الأدوات ثم قم بإضافة عنوان لمربع السرد بمحاذاة العنوان Files و قم بتغيير العنوان إلى Sub Directories.
٣. اختر رمز مربع السرد List Box من مربع الأدوات ثم قم بإضافة مربع سرد أسفل العنوان Sub Directories (انظر شكل ٧-٣).
٤. قم بإدخال اسم مربع السرد إلى الخاصية ID بمربع الخصائص وليكن IDC_SUB_DIR.

٥. قم بتخصيص القيمة **Extended** للخاصية **Selection** بمربع الخصائص.
٦. قم بتخصيص القيمة **False** للخاصية **No Integral Height** حتى يتم حساب ارتفاع مربع السرد تبعاً للعناصر الموجودة فيه. يجب أن يظهر المربع الحوارى الآن كما في شكل ٧-٣.



شكل ٧-٣ إضافة مربع السرد إلى المربع الحوارى

لربط مربع السرد بمتغير، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.
٢. انقر التصنيف **CListsDlg** بزر الفأرة الأيمن ثم اختر **Add Variable**.
٣. قم بتنشيط مربع الاختيار **Control variable** لأننا نرغب في إنشاء متغير مرتبط بإحدى أدوات التحكم الموجودة بالمربع الحوارى.
٤. اختر أداة شجرة البيانات **IDC_SUB_DIRS** من مربع السرد **Control ID**.
٥. تأكد من اختيار **Control** من مربع السرد **Category** لأننا نحتاج إلى التعامل مع الخصائص الأخرى المصاحبة لهذه الأداة.

٦.

٧. تأكد من اختيار نوع البيانات **CListBox** بمربع السرد والتحرير **Variable** .
Type .

٨. اكتب اسم المتغير في خانة **Variable name** وليكن **m_lbSubDir** واختبر
 درجة المتغير من مربع السرد **Access** وهو **Public** في هذه الحالة.

٩. انقر زر **Finish** لإغلاق نافذة المعالج.

استخدام أداة القائمة

تعتبر أداة القائمة أكثر الأنواع الأربعة تعقيداً، كما تستخدم بكثرة داخل نوافذ العرض أكثر منها في المربعات الحوارية، كما أن لها القدرة على إظهار الصور إلى جانب النص ولعل مستكشف **Windows** وجهاز الكمبيوتر خير مثال لأداة القائمة. يوجد أربعة أنماط عرض لأداة القائمة والموضحة بمجدول ٥-٣ والتي يمكنك اختيار أي منها من الخاصية **View** بمربع الخصائص.

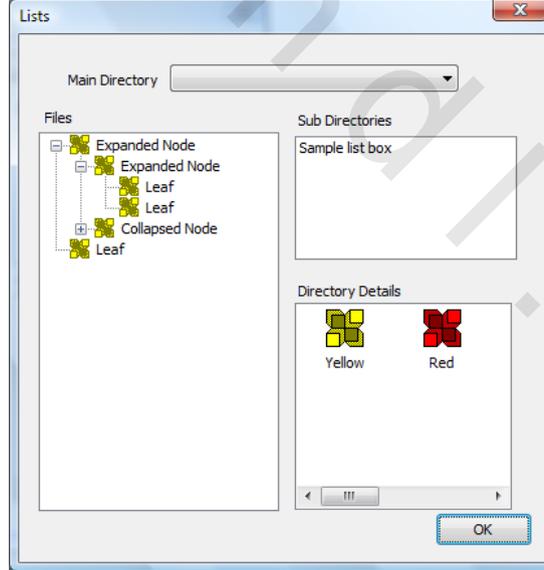
جدول ٥-٣ أنواع أداة القائمة

النوع	المواصفات
Icon	تقوم بإظهار أيقونات كبيرة ويكون النص أسفل الأيقونة ويكون الترتيب بالعرض ثم لأسفل
Small Icon	تقوم بإظهار أيقونات صغيرة ويكون النص يمين الأيقونة ويكون الترتيب بالعرض ثم لأسفل
List	مثل النوع السابق إلا أن الترتيب يكون لأسفل ثم بالعرض
Report	تظهر معلومات العناصر داخل أعمدة لها رؤوس

عندما تفقد أداة القائمة التركيز، يتم إلغاء العنصر المختار. فإذا أردت أن يكون العنصر المختار مرئياً دائماً، قم بتخصيص القيمة **True** للخاصية **Show Selection always** داخل مربع الخصائص.



١. لإضافة أداة القائمة إلى المربع الحوارى، تابع معنا الخطوات الآتية:
قم بفتح قالب المربع الحوارى **IDD_LISTS_DIALOG** داخل نافذة محرر الموارد.
٢. اختر رمز أداة العنوان من مربع الأدوات ثم قم بإضافة عنوان لأداة القائمة أسفل مربع السرد وقم بتغيير العنوان إلى **Directory Details**.
٣. اختر رمز أداة القائمة **List Control** من مربع الأدوات ثم قم بإضافة أداة قائمة أسفل العنوان **Directory Details** (انظر شكل ٧-٤).
٤. قم بإدخال اسم أداة القائمة إلى الخاصية **ID** بمربع الخصائص وليكن **.IDC_DIR_DET**.
٥. قم بتخصيص القيمة **Report** إلى الخاصية **View** وذلك باختيارها من مربع السرد والتحرير المجاور. يجب أن يظهر المربع الحوارى الآن كما فى شكل ٧-٤.



شكل ٧-٤ إضافة أداة القائمة إلى المربع الحوارى

لربط أداة القائمة بتغيير، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.
٢. انقر التصنيف **CListsDlg** بزر الفأرة الأيمن ثم اختر **Add Variable** من القائمة الموضوعية، يظهر معالج إضافة متغير جديد.
٣. قم بتنشيط مربع الاختيار **Control variable** لأننا نرغب في إنشاء متغير مرتبط بإحدى أدوات التحكم الموجودة بالمربع الحوارى.
٤. اختر أداة القائمة **IDC_DIR_DET** من مربع السرد **Control ID**.
٥. تأكد من اختيار **Control** من مربع السرد **Category** لأننا نحتاج إلى التعامل مع الخصائص الأخرى المصاحبة لهذه الأداة.
٦. تأكد من اختيار نوع البيانات **CListCtrl** بمربع السرد والتحرير **Variable** **Type**.
٧. اكتب اسم المتغير في خانة **Variable name** وليكن **m_lbidirDetails** واختر درجة المتغير من مربع السرد **Access** وهو **Public** في هذه الحالة.
٨. انقر زر **Finish** لإغلاق نافذة المعالج.

إضافة العناصر لأدوات السرد

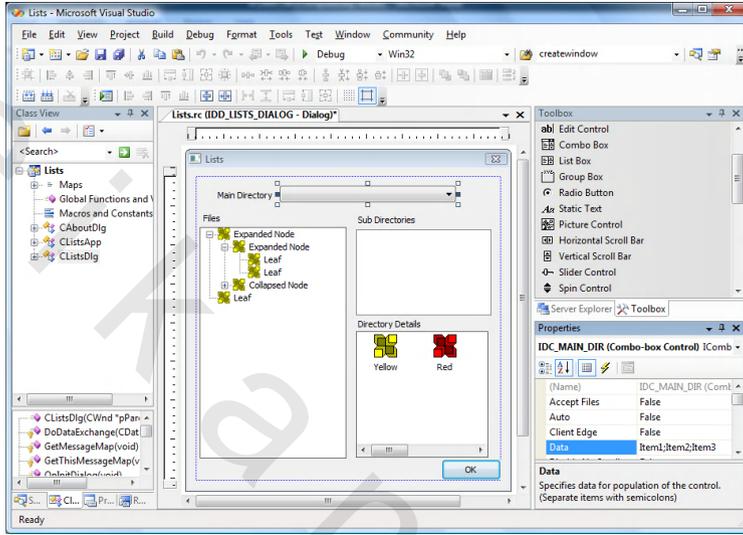
تعتبر عملية إدخال العناصر إلى أدوات السرد أولى خطوات برمجة تلك الأدوات. ورغم التقارب الشديد بين الطرق المتبعة لإدخال العناصر إلى أدوات السرد الأربعة، إلا أن لكل أداة بعض المتطلبات الخاصة التى تختلف فيها عن باقى الأدوات وهو ما سنتعرف عليه من خلال الجزء المتبقى من هذا الفصل.

ملء مربع السرد والتحرير بالبيانات

يعد مربع السرد والتحرير هو أداة السرد الوحيدة التى يمكن إدخال عناصرها أثناء التصميم من خلال مربع الخصائص وذلك كما يلى (انظر شكل ٧-٥):

١. قم بتنشيط مربع السرد والتحرير الموجود بالمربع الحوارى.
٢. استخدم شريط التمرير للوصول إلى الخاصية **Data** داخل مربع الخصائص.

٣. قم بكتابة عناصر مربع السرد واحداً تلو الآخر باستخدام الفاصلة المنقوطة ";" لفصل هذه العناصر.



شكل ٧-٥ ملء مربع السرد والتحرير باستخدام مربع الخصائص

وهذه الطريقة نادرة الاستخدام، حيث يتم غالباً إدخال العناصر أثناء تنفيذ البرنامج من خلال الدالة `OnInitDialog()` التي يتم استدعاؤها بمجرد فتح مربع الحوارى وقبل ظهور مربع السرد والتحرير.

يطلق على التصنيف الذى يحتوى على وظائف مربع السرد والتحرير داخل مكتبة **MFC** اسم `CComboBox` ويحتوى على مجموعة من الدوال التي تخصص بإضافة وحذف العناصر والموضحة بجدول ٧-٤.

جدول ٧-٤ الدوال المستخدمة لإضافة العناصر لمربع السرد والتحرير

الاستخدام	الدالة
إضافة عنصر إلى نهاية المربع أو في ترتيبه الصحيح	<code>AddString()</code>
حذف عنصر من المربع	<code>DeleteString()</code>
إضافة عنصر لمكان معين داخل المربع	<code>InsertString()</code>

الاستخدام	الدالة
حذف جميع عناصر المربع	ResetContent()
إضافة أسماء الملفات كعناصر عادية داخل المربع	Dir()

في المثال الذي بين أيدينا، تقوم الدالة `OnInitDialog()` بملء مربع السرد والتحرير ببعض المجلدات الرئيسية. قم بإضافة دالة جديدة إلى التصنيف `CListsDlg` باسم `PopulateCombo()` وهي من النوع `void` وذلك باستخدام معالج إضافة الدالة ثم قم بتنشيط التبويب `Class View` وانقر الدالة `OnInitDialog()` الموجودة أسفل التصنيف `CListDlg` نقراً مزدوجاً وقم بتعديل كود هذه الدالة وإنشاء كود الدالة `PopulateCombo()` كما يلي (اختر التصنيف `CListDlg` من الجزء العلوى من التبويب ثم انقر الدالة `OnInitDialog()` من الجزء السفلى نقراً مزدوجاً):

```

1.  BOOL CListsDlg::OnInitDialog()
2.  {
3.      CDialog::OnInitDialog();

4.      // Add "About..." menu item to system menu.
5.      // IDM_ABOUTBOX must be in the system command
        range.
6.      ASSERT((IDM_ABOUTBOX & 0xFFF0) ==
                IDM_ABOUTBOX);
7.      ASSERT(IDM_ABOUTBOX < 0xF000);
8.      CMenu* pSysMenu = GetSystemMenu(FALSE);
9.      if (pSysMenu != NULL)
10.     {
11.         CString strAboutMenu;
12.         strAboutMenu.LoadString(IDS_ABOUTBOX);
13.         if (!strAboutMenu.IsEmpty())
14.         {
15.             pSysMenu->AppendMenu(MF_SEPARATOR);
16.             pSysMenu->AppendMenu(MF_STRING,
                IDM_ABOUTBOX, strAboutMenu);
17.         }
18.     }

```

```

19. // Set the icon for this dialog. The framework does this
    automatically
20. // when the application's main window is not a dialog
21. SetIcon(m_hIcon, TRUE); // Set big icon
22. SetIcon(m_hIcon, FALSE); // Set small icon
23. // TODO: Add extra initialization here
24. PopulateCombo();
25. return TRUE; // return TRUE unless you set the focus to
    a control
26. }

27. void CListsDlg::PopulateCombo(void)
28. {
29.     TCHAR szBuffer[MAX_PATH];
30.     GetWindowsDirectory(szBuffer,MAX_PATH);
31.     m_cbMainDir.AddString (szBuffer);

32.     szBuffer[2] = 0;
33.     m_cbMainDir.AddString(szBuffer);

34.     GetSystemDirectory(szBuffer,MAX_PATH);
35.     m_cbMainDir.AddString (szBuffer);

36.     GetCurrentDirectory(MAX_PATH,szBuffer);
37.     m_cbMainDir.AddString(szBuffer);
38. }

```

وعن هذا الكود، نوضح ما يلي:

- الدوال `GetSystemDirectory` و `GetWindowsDirectory()` و `GetCurrentDirectory` الموجودة بالسطور ٣٠، ٣٤، ٣٦ هي دوال عامة تقوم بملء المصفوفة `szBuffer` باسم المسار المناسب. كما يقوم المتغير `MAX_PATH` بتعيين أكبر طول للمسار والذي غالبا ما يكون ٢٦٠.
- بعد الحصول على المسار داخل مصفوفة `szBuffer`، يتم تمرير المصفوفة إلى الدالة `AddString()` الخاصة بالمتغير `m_cbMainDir` والذي يقوم بإضافة المسار إلى

مربع السرد والتحرير. ولأن قيمة الخاصية **Sort** هي **True**، يتم ترتيب العناصر المضافة لمربع السرد والتحرير ترتيباً أبجدياً.

يجب أن تقوم باختيار القيم المرجعة من الدوال **AddString()** و **InsertString()** والتي تقوم بإرجاع الرسائل **CB_ERR** أو **CB_ERRSPACE** عند حدوث خطأ.



الاستجابة لأحداث مربع السرد والتحرير

الغرض من استخدام مربع السرد والتحرير هو السماح للمستخدم باختيار عنصر واحد من العناصر الموجودة فيه، وحينئذٍ يجب أن يتعرف البرنامج على هذا الاختيار باستقبال الحدث **CBN_SELCHANGE** من مربع السرد والتحرير إلى المربع الحوارى كلما تم تغيير العنصر المختار.

قم بإضافة دالة الحدث **CBN_SELCHANGE** لمربع السرد والتحرير **IDC_MAIN_DIR** من خلال مربع الخصائص باسم **OnCbnSelChangeMainDir()** ثم قم بإضافة متغير عضو من النوع **CString** باسم **m_strMainDir** وقم بإضافة الكود التالى للدالة:

```
1. void CListsDlg::OnCbnSelChangeMainDir()
2. {
3.     // TODO: Add your control notification handler code here
4.     int nIndex = m_cbMainDir.GetCurSel();
5.     if(nIndex != CB_ERR)
6.     {
7.         m_cbMainDir.GetLBText(nIndex,m_strMainDir);
8.         PopulateTree();
9.     }
10.    PopulateListBox();
11. }
```

وعن هذا الكود نوضح ما يلي:

- فى السطر رقم ٤ تم استدعاء الدالة **GetCurSel()** التى تقوم بإرجاع ترتيب العنصر المختار إلى المتغير **nIndex**.

- في السطر رقم ٥، إذا لم يتم اختيار أي من عناصر مربع السرد والتحرير، يتم إرجاع القيمة **CB_ERR** وهي قيمة خاصة متعارف عليها.
- في السطر رقم ٧، إذا كانت قيمة **nIndex** صحيحة، يتم تمريرها إلى الدالة **GetLBText()** بحيث يتم تخزين النص المختار في المتغير **m_strMainDir**.
- في السطر رقم ٨ تم استدعاء الدالة **PopulateTree** التي ستقوم بملاء شجرة البيانات والتي سنقوم بإنشائها في البند التالي.
- في السطر رقم ١٠ تم استدعاء الدالة **PopulateListBox()** التي ستقوم بملاء مربع السرد وستعرض لها أيضاً بعد قليل.

ملء شجرة البيانات

عند إضافة العناصر إلى شجرة البيانات، يجب تخزين معلومات عن تركيب الشجرة حتى يتم وضع كل عنصر في المكان المناسب. يحتوي التصنيف **CTreeCtrl** على وظائف شجرة البيانات كما يحتوي على مجموعة من الدوال التي تختص بإضافة وحذف العناصر والموضحة بجدول ٧-٥ التالي.

جدول ٧-٥ الدوال المستخدمة من قبل شجرة البيانات

الاستخدام	الدالة
إضافة عنصر إلى شجرة البيانات	InsertItem()
حذف عنصر من شجرة البيانات	DeleteItem()
حذف جميع عناصر شجرة البيانات	DeleteAllItems()

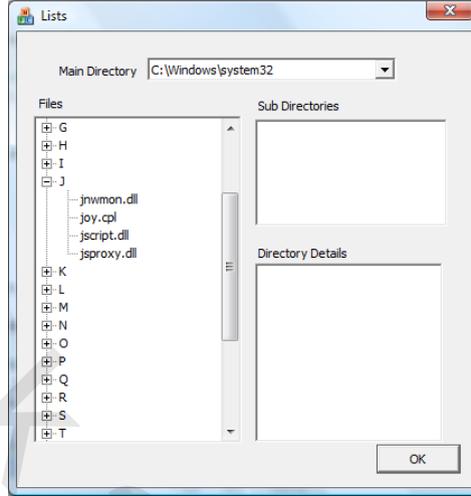
في المثال الذي بين أيدينا، تقوم شجرة البيانات بعرض ملفات المجلد المختار بمربع السرد والتحرير في ترتيب أبجدي. قم أولاً بإنشاء دالة عضو جديدة باسم **PopulateTree()** من النوع **void** والتي تقوم بإنشاء عنصر أبوى لكل حرف من الحروف الأبجدية اللاتينية وعددها ٢٦ عنصر ثم تقوم بإنشاء العنصر رقم ٢٧ والذي سيحتوي على الملفات التي لا تبدأ بحرف من الحروف الأبجدية. وفي سبيل ذلك تستخدم الدالة بعض دوال **Windows**

اللازمة لمعرفة محتويات المجلد ووضع كل عنصر في مكانه المناسب. إذا كنت قد انتهيت الآن من إنشاء الدالة `PopulateTree()`، قم بإدخال الكود التالي إلى الدالة:

```
1. void CListsDlg::PopulateTree(void)
2. {
3.     m_treeFiles.DeleteAllItems();
4.     HTREEITEM hLetter[27];
5.     for(int nChar = 'A' ; nChar <= 'Z' ; nChar++)
6.         hLetter[nChar - 'A'] = m_treeFiles.InsertItem
           ((TCHAR*)&nChar);
7.     hLetter[26] = m_treeFiles.InsertItem("Other");
8.     HANDLE hFind;
9.     WIN32_FIND_DATA dataFind;
10.    BOOL bMoreFiles = TRUE;
11.    CString strFile;
12.    hFind = FindFirstFile(m_strMainDir + "\\*.*", &dataFind);
13.    while(hFind != INVALID_HANDLE_VALUE && bMoreFiles
           == TRUE)
14.    {
15.        if(dataFind.dwFileAttributes ==
           FILE_ATTRIBUTE_ARCHIVE)
16.        {
17.            int nChar = dataFind.cFileName[0];
18.            if(islower(nChar))
19.                nChar -= 32;
20.            if(isalpha(nChar))
21.                nChar -= 'A';
22.            else
23.                nChar = 26;
24.            m_treeFiles.InsertItem(dataFind.cFileName,
           hLetter[nChar]);
25.        }
26.        bMoreFiles = FindNextFile(hFind,&dataFind);
27.    }
28.    FindClose(hFind);
29. }
30. }
```

وعن هذا الكود نوضح ما يلي:

- فى السطر رقم ٣ تم استدعاء الدالة `DeleteAllItems()` لحذف جميع عناصر الشجرة وذلك لأننا نقوم بإعادة ملء الشجرة كلما تم اختيار مجلد جديد من مربع السرد والتحرير.
 - فى السطر رقم ٤ تم تعريف مصفوفة `hLetter[]` والتي ستحتوى على الحروف الأبجدية وهى من النوع `HTRIEITEM`.
 - فى السطور ٦ و ٧ يتم استدعاء الدالة `InsertItem()` داخل الدوارة `For` لإضافة الحروف الأبجدية وكلمة `Other` كعناصر أساسية فى الشجرة وذلك لأننا لم ندخل معاملات أخرى تحدد مكان هذه العناصر كما سنرى بعد قليل.
 - فى السطر رقم ١٢ يتم استدعاء الدالة `FindFirstFile()` وذلك بتمرير معامل يحتوى على جميع الملفات الموجودة بالمجلد.
 - فى السطر رقم ١٥ يتم اختبار صفات الملف للتأكد من أنه ملف وليس مجلد.
 - فى السطر رقم ١٧ يتم تخزين الحرف الأول من اسم الملف وتحويله فى السطر ١٩ إلى حرف كبير فى حالة الضرورة.
 - فى السطر ٢٠ يتم اختبار الحرف الأول من اسم الملف. فإذا كان الملف يبدأ بحرف أبجدي، قام البرنامج بوضعه فى المكان المناسب داخل المصفوفة `hLetter[]` وإلا قام بوضعه فى آخر المصفوفة.
 - فى السطر ٢٤، يتم وضع الملف فى المكان الصحيح داخل الشجرة باستخدام دالة `InsertItem()`.
- والآن قم ببناء البرنامج وتنفيذه. اختر مجلد من مربع السرد والتحرير `Main Directory`، تلاحظ ظهور شجرة بالملفات الموجودة فى هذا المجلد داخل شجرة البيانات `Files` (انظر شكل ٧-٦).



شكل ٧-٦ عرض الملفات داخل شجرة البيانات

ملء مربع السرد

تشبه طريقة ملء مربع السرد إلى حد كبير الطريقة المستخدمة في ملء مربع السرد والتحرير وذلك لأن عناصر مربع السرد والتحرير هي بالفعل عناصر مربع السرد، إلا أن الاختلاف يكمن في طريقة العرض وإمكانية اختيار أكثر من عنصر في حالة مربع السرد. يحتوى التصنيف **CListBox** على وظائف مربع السرد والتي تتمثل في مجموعة من الدوال التي تختص بإضافة وحذف العناصر وهي مشابهة تماماً لتلك الموجودة بجدول ٧-٤ السابق. في المثال الذى بين أيدينا، نريد لمربع السرد أن يحتوى على المجلدات الفرعية الموجودة تحت المجلد الرئيسى بمربع السرد والتحرير. قم بإنشاء دالة جديدة باسم **PopulateListBox()** من النوع **void** ثم قم بإضافة الكود التالى إلى الدالة.

1. **void CListsDlg::PopulateListBox(void)**
2. **{**
3. **m_lbSubDir.ResetContent();**
4. **HANDLE hFind;**
5. **WIN32_FIND_DATA dataFind;**
6. **BOOL bMoreFiles = TRUE;**
7. **hFind = FindFirstFile(m_strMainDir + "*.*", &dataFind);**
8. **while(hFind != INVALID_HANDLE_VALUE && bMoreFiles**

```

9.      {
10.     if(dataFind.dwFileAttributes ==
11.         FILE_ATTRIBUTE_DIRECTORY)
12.     {
13.         if(strcmp(dataFind.cFileName, ".")
14.            if(strcmp(dataFind.cFileName, "..")
15.                m_lSubDir.AddString(dataFind.cFileName);
16.            }
17.        }
18.    FindClose(hFind);
19. }

```

وعن هذا الكود نوضح ما يلي:

- في السطر رقم ٣ تم استدعاء الدالة **ResetContent()** لإزالة محتويات مربع السرد وذلك لأننا نقوم بملء مربع السرد كلما تم تغيير اسم المجلد من مربع السرد والتحرير.
- في السطر رقم ١٠، يتم اختبار العنصر للتأكد من أنه مجلد فرعي وليس ملف.
- في السطور ١٢ و ١٣ يتم تجاهل المجلدات الفرعية "." و "..".
- في السطر رقم ١٤ يتم إضافة اسم المجلد الفرعي إلى مربع السرد باستخدام الدالة **.AddString()**

يجب أن تقوم باختيار القيم المرجعة من الدوال **AddString()** و **InsertString()** والتي تقوم بإرجاع الرسائل **CB_ERR** أو **CB_ERRSPACE** عند حدوث خطأ.



الاستجابة لأحداث مربع السرد

عندما يقوم المستخدم بأداء حدث من الأحداث داخل مربع السرد، كأن يقوم باختيار عنصر أو أكثر من العناصر الموجودة بالمربع، يقوم **Windows** بإرسال رسالة إلى المربع الحوارى تسمى **LBN_SELCHANGE**.

في المثال الذي بين أيدينا، يتيح لك مربع السرد اختيار أكثر من عنصر، وعلى هذا يتم تخزين العناصر المختارة داخل متغير جديد وليكن `m_strList` وهو من النوع `CStringList`. قم بإضافة المتغير الجديد إلى تصنيف المربع الحوارى باستخدام المعالج `Add Member Variable Wizard` ثم قم بإنشاء دالة الحدث `LBN_SELCHANGE` لمربع السرد `IDC_SUB_DIR` من خلال مربع الخصائص واترك اسمها الافتراضى (`OnLbnSelchangeSubDir()`) كما هو. تقوم هذه الدالة بالمهام الآتية:

١. تحدد عدد العناصر المختارة.
٢. تتعرف على نص كل عنصر ثم تقوم بإضافته إلى المتغير `m_strList`.
٣. تقوم باستدعاء دالة جديدة باسم (`PopulateListControl()`) التى تقود المتغير `m_strList` لإضافة التفاصيل لأداة القائمة كما سنرى بعد قليل.

والآن قم بإضافة الكود التالى إلى الدالة (`OnLbnSelchangeSubDirs()`) كما يلى:

```
1. void CListsDlg::OnLbnSelchangeSubDir()
2. {
3.     // TODO: Add your control notification handler code here
4.     int nSelCount = m_lbSubDir.GetSelCount();
5.     m_strList.RemoveAll();
6.     if(nSelCount)
7.     {
8.         CString str;
9.         LPINT pltems = new int[nSelCount];
10.        m_lbSubDir.GetSelItems(nSelCount,pltems);
11.        for(int i= 0 ; i < nSelCount; i++)
12.        {
13.            m_lbSubDir.GetText(pltems[i],str);
14.            m_strList.AddTail(str);
15.        }
16.        delete [] pltems;
17.    }
18.    PopulateListControl();
19. }
```

وعن هذا الكود، نوضح ما يلي:

- في السطر رقم ٤ يتم استدعاء الدالة **GetSelCount()** التي تقوم بإرجاع عدد العناصر المختارة وتستخدم في حالة الاختيار المتعدد فقط.
- في السطر رقم ٥ يتم استدعاء الدالة **RemoveAll()** لإزالة جميع العناصر الموجودة بالمتغير **m_strList** وذلك لإعادة بناء العناصر المختارة من جديد.
- في السطر رقم ٩ يتم حجز مصفوفة من الأرقام الصحيحة لتحتوي على أرقام العناصر المختارة بحيث يكون حجم المصفوفة هو عدد العناصر المختارة **nSelCount**.
- في السطر رقم ١٠ تقوم الدالة **GetSelItems()** بإرجاع قائمة بالعناصر المختارة وتخزينها داخل المصفوفة.
- في السطر رقم ١١ يتم استخدام الدوارة **For** بعدد العناصر المختارة وفي كل مرة يتم استدعاء الدالة **GetText()** لتخزين النص المختار في المتغير **str**.
- في السطر رقم ١٤ يتم إضافة محتويات المتغير **str** إلى نهاية المتغير **m_strList** الذي سيحتوي على جميع العناصر المختارة.
- في السطر رقم ١٦ يتم حذف المصفوفة **pltems** لأننا لم نعد في حاجة إليها.
- في السطر رقم ١٨ يتم استدعاء الدالة **PopulateListControl()** المستخدمة في ملء أداة القائمة والتي سنتعرف عليها بعد قليل.

ملء أداة القائمة

تختلف طريقة ملء أداة القائمة نوعاً ما عن تلك المستخدمة مع باقى أنواع السرد التي ذكرناها في هذا الفصل. كما أنها تختلف باختلاف نوع أداة القائمة نفسها، حيث يوجد كما ذكرنا أربعة أنواع من أداة القائمة وهي الرمز **Icon** والرمز الصغير **Small Icon** والقائمة **List** والتقرير **Report** والتي قمنا بشرحها قبل ذلك. وبعد التقرير أكثر هذه الأنواع استخداماً وفيه يتم إظهار البيانات في أعمدة رأسية. فمثلاً طريقة عرض التقرير في

مستكشف Windows تقوم بعرض أعمدة للاسم **Name** والحجم **Size** والنوع **Type** وآخر تاريخ للتعديل **Modified** وعلى هذا فكل صف من الصفوف يعبر عن بيانات ملف معين.

يحتوى التصنيف **CListCtrl** على وظائف أداة القائمة ممثلةً في مجموعة من الدوال التى تختص بإضافة وحذف العناصر والموضحة بجدول ٧-٦ التالى.

جدول ٧-٦ الدوال المستخدمة من قبل أداة القائمة

الاستخدام	الدالة
إضافة عمود جديد في مكان معين داخل القائمة	InserColumn()
حذف عمود من القائمة	DeleteColumn()
إضافة عنصر جديد للقائمة	InsertItem()
حذف عنصر من القائمة	DeleteItem()
حذف جميع عناصر القائمة	DeleteAllItems()
إضافة نص لعنصر فرعى في القائمة	SetItemText()

في المثال الذى بين أيدينا، نستخدم أداة سرد من نوع التقرير تحتوى على ثلاثة أعمدة، الأول اسم المجلد **Directory** والثانى عدد الملفات الموجودة بالمجلد **Files** والأخير حجم هذه الملفات **Size KB**. الخطوة الأولى هى إنشاء الأعمدة الثلاثة لأنها لن تتغير أثناء سير البرنامج وذلك باستخدام الدالة **OnInitDialog()**. قم بإضافة الكود التالى للدالة فى السطور من ٧ إلى ٩.

```

1.  BOOL CListDlg::OnInitDialog()
2.  {
3.  .....
4.
5.      // TODO: Add extra initialization here
6.      PopulateCombo();
7.      m_lbDirDetails.InsertColumn(0,"Directory",
                                LVCFMT_LEFT,70);
8.      m_lbDirDetails.InsertColumn(1,"Files",LVCFMT_RIGHT,

```

```

50);
9.     m_lbDirDetails.InsertColumn(2,"Size
                                   KB",LVCFMT_RIGHT, 60);
10.    return TRUE; // return TRUE unless you set the focus to
        a control
11.    }

```

لاحظ إضافة الأعمدة الثلاثة في السطور من ٧ إلى ٩ باستخدام الدالة InsertColumn() والتي تحتوى على أربعة معاملات: الأول ترتيب العمود والثاني عنوان العمود والثالث محاذاة النص والرابع عرض العمود الذى يمكنك تغييره بالفأرة أثناء تنفيذ البرنامج.

الخطوة التالية هى إدخال عناصر أداة القائمة وذلك من خلال الدالة PopulateListControl(). قم بإنشاء الدالة باستخدام المعالج ثم قم بإدخال الكود التالى لهذه الدالة:

```

1.    void CListsDlg::PopulateListControl(void)
2.    {
3.        m_lbDirDetails.DeleteAllItems();
4.        POSITION pos;
5.        for(pos = m_strList.GetHeadPosition();pos != NULL;)
6.        {
7.            int nItem;
8.            HANDLE hFind;
9.            WIN32_FIND_DATA dataFind;
10.           BOOL bMoreFiles = TRUE;
11.           CString str;
12.           CString strFind;
13.           str = m_strList.GetAt(pos);
14.           nItem = m_lbDirDetails.InsertItem(0,str);
15.           strFind = m_strMainDir + "\\\" + str + "\\*.***";
16.           hFind = FindFirstFile(strFind,&dataFind);
17.           int nFileCount = 0;
18.           double nFileSize = 0;
19.           while(hFind != INVALID_HANDLE_VALUE &&
                                   bMoreFiles == TRUE)
20.           {
21.               if(dataFind.dwFileAttributes ==

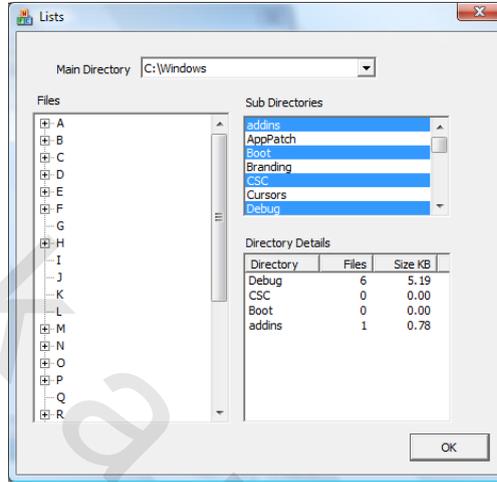
```

```
FILE_ATTRIBUTE_ARCHIVE)
22.     {
23.         nFileCount++;
24.         nFileSize += (dataFind.nFileSizeHigh *
                        MAXDWORD) + dataFind.nFileSizeLow;
25.     }
26.     bMoreFiles = FindNextFile(hFind,&dataFind);
27. }
28. FindClose(hFind);
29. str.Format("%1d",nFileCount);
30. m_lbDirDetails.SetItemText(nItem,1,str);
31. str.Format("%-1.2f",nFileSize /1024.0);
32. m_lbDirDetails.SetItemText(nItem,2,str);
33. m_strList.GetNext(pos);
34. }
35. }
```

وعن هذا الكود نوضح ما يلي:

- في السطر رقم ٣ يتم حذف جميع العناصر الموجودة بأداة القائمة عدا الأعمدة الرئيسية باستخدام الدالة `DeleteAllItems()`.
- في السطر رقم ٥ يتم استخدام الدوارة `for` بعدد المجلدات المختارة بمربع السرد والمخزنة داخل المتغير `m_strList`.
- في السطر رقم ١٤ يتم إضافة اسم المجلد بالعمود الأول `Directory` في أداة القائمة باستخدام الدالة `InsertItem()`.
- في السطور ٢٣ و ٢٤ يتم زيادة عدد الملفات الموجودة بالمجلد `Directory` والحجم الكلي لهذه الملفات باستخدام الدالة `SetItemText()`.
- في السطر رقم ٣٠ يتم إضافة عدد ملفات المجلد في العمود الثاني `Files` باستخدام الدالة `SetItemText()`.
- في السطر رقم ٣٢ يتم إضافة حجم ملفات المجلد بالعمود الثالث `Size KB` باستخدام الدالة `SetItemText()`.

وفي النهاية، قم ببناء البرنامج وتنفيذه ثم قم باختبار الاختيار المتعدد من مربع السرد، لتحصل على نتائج مشابهة لتلك الموجودة بشكل ٧-٧ التالي.



شكل ٧-٧ اختيار أكثر من مجلد وعرض بياناً في القائمة

