الفصل الثالث
استخدام بيئة التطوير المتكاملة

تحتوى مجموعة **Visual Studio 2005** على بيئة تطوير متكاملة مشتركة عالية الجودة بها العديد من الأدوات التى تساعدك على تطوير تطبيقاتك بسهولة تامة بالإضافة إلى إمكانية توفيق بيئة التطوير لتتوافق مع زوقك الشخصى وخبرتك المسبقة فى التعامل مع بيئات التطوير المختلفة.

بانتهاء هذا الفصل ستتعرف على:

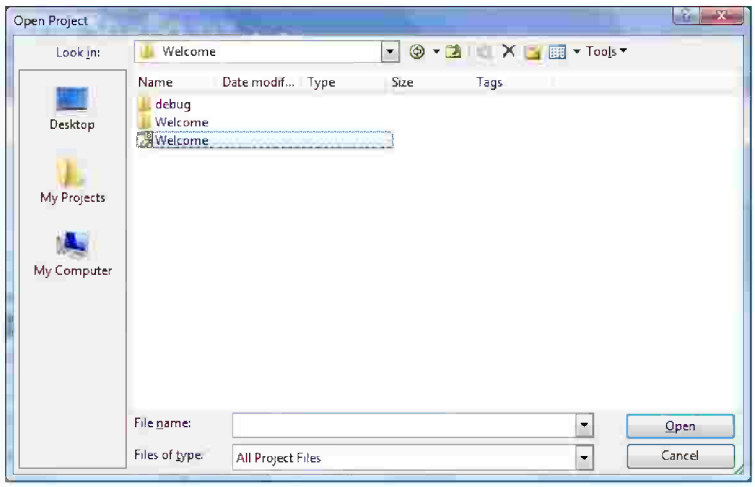
- فتح مشروع موجود.
- نافذة عرض التصنيفات **Class View**.
- نافذة عرض الموارد **Resource View**.
- نافذة مستكشف الحل **Solution Explorer**.
- السمات الهامة داخل بيئة تطوير **Visual Studio 2005**.
- توفيق بيئة تطوير **Visual Studio 2005**.
- الخروج من بيئة تطوير **Visual Studio 2005**.

مع النظرة الأولى لبيئة التطوير المتكاملة **Visual Studio 2005**، يخيل إليك صعوبتها، فهناك القوائم والأزرار التى يمكنك نقرها لأداء المهام التى تقوم بها الخيارات الموجودة بالقوائم اختصاراً للوقت والجهد. كما يقودك العديد من هذه الخيارات وتلك الأزرار إلى عدد من مربعات حوارية أخرى بها العديد من الخيارات الإضافية. وسنقوم أثناء عرضنا للكتاب بالتعرف على محتويات هذه البيئة بالتفصيل، لذا لا تتعجل فى معرفة كل شيءٍ عن كل أداة من الأدوات أو طريقة من الطرق. كل ما أريده منك أن تأخذ المعلومة الجديدة وتضعها فى حقيبة أفكارك حتى يأتى تفصيلها بعد ذلك.

فتح مشروع موجود

لفتح أحد مشروعاتك مرةً أخرى، تأكد أنك داخل بيئة تطوير **Visual Studio 2005** وإلا قم بفتحها من قائمة **Start** كما تعلمت فى الفصل السابق ثم اتبع أيٍ من الطريقتين الآتيتين:

- إذا كان المشروع الذى ترغب فى فتحه ضمن آخر أربعة مشروعات قمت بفتحها قم باختيار المشروع مباشرة من قائمة المشروعات الأخيرة الموجودة بصفحة البدء **Start Page** أو افتح قائمة **File** ثم قم باختيار المشروع من المجموعة الفرعية **Recent Projects**.
- افتح قائمة **File** ثم اختر **Open** من القائمة المنسدلة، تظهر قائمة فرعية أخرى. اختر منها **Project/Solution**، يظهر المربع الحوارى **Open Project** (انظر شكل ٣-١). قم بتحديد اسم المشروع ثم انقر زر **Open**، يبدأ تحميل المشروع داخل بيئة تطوير**Visual Studio 2005**.



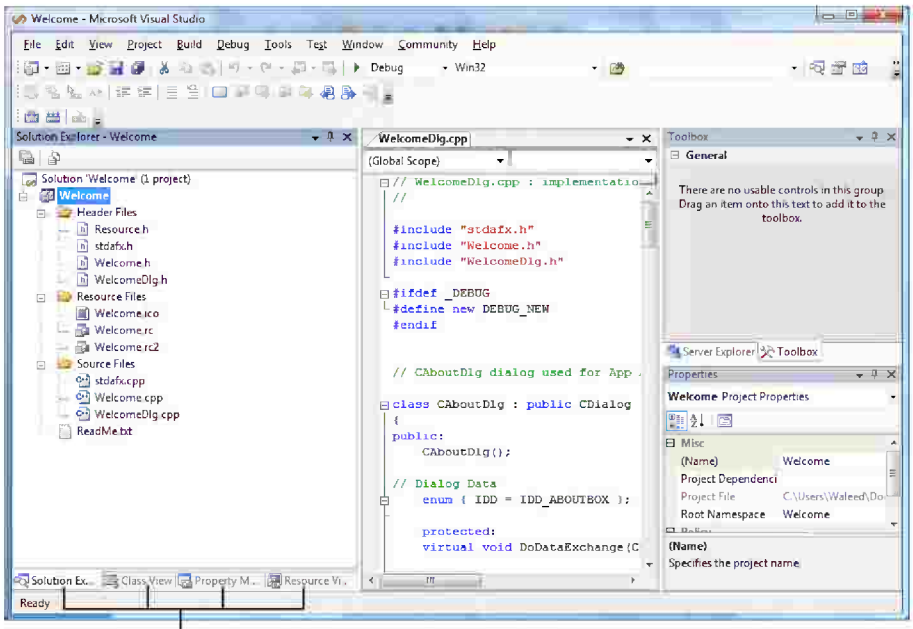
شكل ٣-١ فتح المشروع Welcome من خلال المربع الحوارى Open Project

شرح نوافذ المشروع

عند العمل فى بيئة التطوير المتكاملة، تظهر لك العديد من النوافذ ولعل أهمها على الإطلاق النوافذ الأربعة التالية التى تظهر فى صورة تبويبات متجاورة:

- نافذة عرض التصنيفات Class View والتى تقوم بعرض التصنيفات Classes الموجودة بالمشروع
- نافذة عرض الموارد Resource View والتى تقوم بعرض الموارد Resources الموجودة بالمشروع
- نافذة مستكشف الحل Solution Explorer والتى تقوم بعرض جميع الملفات التى يتكون منها المشروع
- نافذة مدير الخصائص Property Manager والتى يتم من خلالها التحكم فى إعدادات المشروع المعرفة داخل صفحات الخصائص

لعرض نافذة من هذه النوافذ، نشط التبويب بنقره بزر الفأرة الأيسر حيث تظهر عناصر التبويب مسلسلة فى شكل شجرة من العناصر التى يمكنك توسيعها أو تقليصها بنقر الزر + أو - على الترتيب والموجود يسار المجموعة مباشرة (انظر شكل ٣-٢).

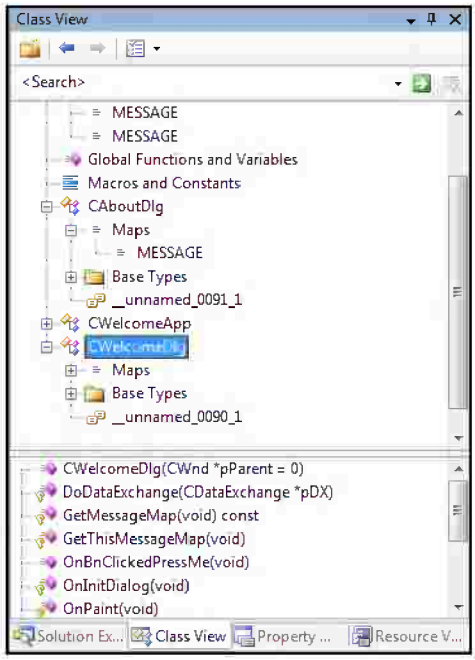


تظهر النوافذ الأربعة في صورة تبويبات متجاورة

شكل ٣-٢ النوافذ الرئيسية للمشروع

نافذة عرض التصنيفات Class View

نشط التبويب **Class View** (تبويب عرض التصنيفات)، تظهر جميع التصنيفات المستخدمة فى المشروع والتى قام معالج التطبيقات **Application Wizard** بإنشائها عند بدء إنشاء المشروع وتظهر هذه التصنيفات على شكل شجرة يمكن إظهار جميع فروعهـا وأغصانها أو إخفاء بعضها كما يروق لك وذلك من خلال زرى التوسيع والتقليص (انظر شكل ٣-٣).



شكل ٣-٣ محتويات نافذة عرض التصنيفات

بالنظر إلى شجرة التصنيفات، تلاحظ أن كل عنصر من عناصر الشجرة يمثل تصنيف من التصنيفات. فإذا قمت باختيار التصنيف، تلاحظ ظهور عناصر فرعية بالجزء السفلى من نافذة التبويب يعبر كل منها عن دالة عضو **Member function** فى التصنيف، أو متغير عضو **Member variable** فى التصنيف. فمثلاً يحتوى التصنيف **CWelcomeDlg** بالجزء العلوى على الدالة **OnBnClickedPressMe()** بالجزء السفلى (راجع شكل ٣-٣) وهى الدالة التى أنشأناها فى المثال **Welcome** فى الفصل السابق. لاحظ ظهور رموز صغيرة يسار كل عنصر من العناصر يدل على نوعية هذا العنصر وذلك كما بالجدول ٣-١ التالى.

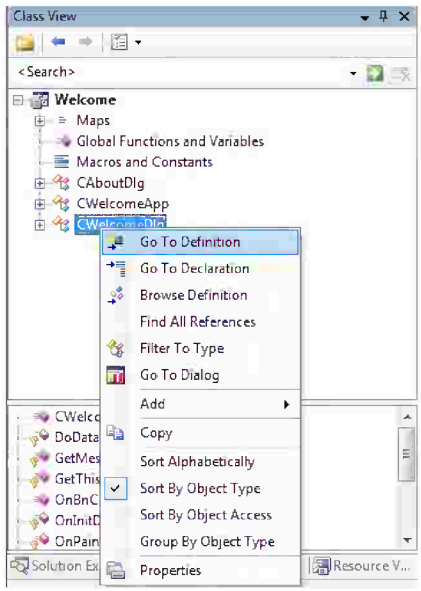
جدول ٣-١ مدلول الرموز الموجودة يسار عناصر التبويب **Class View**

الرمز	المدلول
	تصنيف **Class**

الرمز	المدلول
	دالة عامة Public function
	دالة خاصة Private function
	دالة محمية Protected function
	متغير خاص Private variable
	متغير محمى Protected variable
	متغير عام Public variable

لأن معظم برمجة Visual C++ تعتمد أساساً على التصنيفات، فمن المتوقع أن تتعامل بكثرة مع هذه النافذة دون غيرها من النوافذ الأخرى. ويمكنك التعامل مع عناصر هذه النافذة بالنقر المزدوج للعنصر بزر الفأرة الأيسر إذا ما أردت الذهاب مباشرةً إلى تعريف العنصر داخل التصنيف، أو استعمال القوائم الموضعية Context menu التى تظهر بجوار مؤشر الفأرة عند نقر العنصر بزر الفأرة الأيمن. ومن المعروف عن القوائم الموضعية أن محتوياتها تختلف من عنصر إلى آخر تبعاً لنوع هذا العنصر والعمليات التى يكثر تنفيذها عليه.

انقر التصنيف CWelcomeDlg بزر الفأرة الأيمن، تظهر القائمة الموضعية الخاصة بالتصنيف كما فى شكل ٣-٤ والتى تسمح لك بإجراء العديد من العمليات على هذا التصنيف والتى يمكن توضيحها من خلال جدول ٣-٢ التالى.



شكل ٣-٤ القائمة الموضعية الخاصة بالتصنيف

جدول ٣-٢ استخدام عناصر القائمة الموضعية للتصنيف

الخيار	الاستخدام
Go To Definition	الذهاب مباشرةً إلى ملف المصدر الذى يحتوى على تعريف التصنيف وهو مكافئ تماماً لنقر اسم التصنيف نقراً مزدوجاً
Go To Declaration	الذهاب إلى سطر الإعلان عن التصنيف. فإذا لم يتم الإعلان عن التصنيف وتم تعريفه مباشرةً يتم الذهاب إلى تعريف التصنيف، أى نحصل على نفس نتيجة الخيار السابق
Browse Definition	فتح مستعرض الكائنات على التصنيف المختار
Find All References	البحث عن جميع الرموز والمراجع التى تتوافق مع اسم

الخيار	الاستخدام
	التصنيف المختار حيث تشتمل هذه الرموز على أسماء التصنيفات والدوال والمتغيرات وغيرها، حيث يظهر الاسم ورقم السطر الذى يوجد به العنصر داخل الكود. ففى حالة التصنيف CWelcomeDlg، يتم إيجاد تعريف التصنيف CWelcomeDlg بالإضافة إلى دالة إنشاء التصنيف CWelcomeDlg
Filter To Type	يتم عرض نوع التصنيف المختار فقط داخل نافذة التبويب وإخفاء بقية الأنواع. لعرض جميع العناصر مرةً أخرى، اختر Clear search من القائمة المنسدلة الموجودة بالجزء العلوى من نافذة التبويب
Go To Dialog	إظهار المربع الحوارى فى طريقة عرض التصميم
Add➢Add Function	فتح معالج إضافة دالة جديدة إلى التصنيف المختار
Add➢Add Variable	فتح معالج إضافة متغير جديد إلى التصنيف المختار
Copy	نسخ اسم التصنيف إلى الحافظة حتى تتمكن من لصقه فى أى مكان آخر
Sort by Type	عرض العناصر الفرعية للتصنيف داخل نافذة عرض التصنيفات تبعاً للنوع بحيث يتم عرض الدوال أولاً ثم المتغيرات بعد ذلك
Sort by Access	عرض العناصر الفرعية للتصنيف داخل نافذة عرض التصنيفات تبعاً لمستوى الوصول بحيث يتم عرض العناصر العامة Public أولاً ثم العناصر الخاصة Private وأخيراً العناصر المحمية Protected

الخيار	الاستخدام
Group by Type	تجميع جميع الدوال الأعضاء بالتصنيف داخل مجلد مستقل وكذلك المتغيرات داخل مجلد مستقل آخر
Properties	عرض نافذة خصائص التصنيف التى تحتوى على العديد من المعلومات

وفيما يلى نقوم بإضافة متغير عضو فى التصنيف يحتوى على عدد مرات نقر الزر Press Me ثم إضافة دالة عضو فى التصنيف تقوم بإظهار عدد مرات نقر هذا الزر.

إضافة متغير عضو جديد

لإضافة متغير جديد إلى التصنيف CWelcomeDlg، تابع معنا الخطوات الآتية:

١. انقر التصنيف CWelcomeDlg بزر الفأرة الأيمن ثم اختر Add➢ Add Variable من القائمة الموضعية، يظهر معالج إضافة متغير جديد (انظر شكل ٣-٥).

شكل ٣-٥ إضافة متغير جديد للتصنيف

٢. اختر نوع المتغير من مربع السرد Variable Type وهو int فى هذه الحالة ثم اكتب اسم المتغير فى خانة Variable name وليكن m_nClickCount واختر درجة المتغير من مربع السرد Access وهو Public فى هذه الحالة.

٣. انقر زر Finish لإغلاق نافذة المعالج.

٤. من التبويب Class View، لاحظ ظهور المتغير الجديد داخل التصنيف CWelcomDlg بالجزء السفلى من التبويب. انقر المتغير الجديد نقراً مزدوجاً، تظهر نافذة المحرر وبها نقطة الإدراج عند سطر الكود الذى تم إضافته لتعريف المتغير الجديد.

٥. لأن المتغير الجديد سيحتوى على عدد مرات نقر الزر Press Me، سنقوم أولاً بوضع صفر كقيمة ابتدائية للمتغير وذلك داخل دالة الإنشاء الخاصة بالتصنيف CWelcomDlg.من الجزء السفلى بتبويب عرض التصنيفات Class View، انقر دالة الإنشاء CWelcomeDlg() نقراً مزدوجاً، تلاحظ ظهور ملف المصدر الذى يحتوى على الدالة داخل نافذة المحرر. قم بإضافة الكود كما فى السطر رقم ٦ كما يلى:

```
1.    CWelcomeDlg::CWelcomeDlg(CWnd* pParent /*=NULL*/)
2.    : CDialog(CWelcomeDlg::IDD, pParent)
3.    , m_nClickCount(0)
4.    {
5.        m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
6.        m_nClickCount = 0 ;
7.    }
```

وبذلك ستبدأ عملية العد من الصفر، وكلما قمت بنقر الزر Press Me يزداد العداد بمقدار ١ وذلك من خلال الدالة OnBnClickedPressMe().

٦. انقر الدالة OnBnClickedPressMe() نقراً مزدوجاً، تلاحظ ظهور الدالة داخل نافذة المحرر. قم بإضافة الكود كما فى السطر رقم ٥ كما يلى:

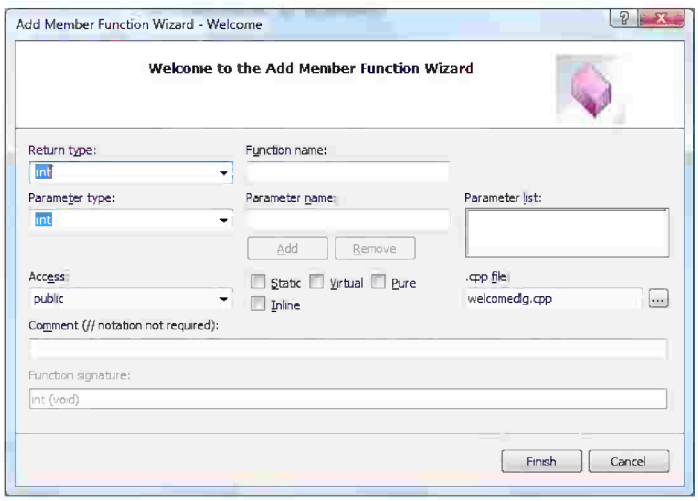
```
1.    void CWelcomeDlg::OnBnClickedPressMe()
2.    {
3.        // TODO: Add your control notification handler code here
```

```
4.        AfxMessageBox("This is our first application");
5.        m_nClickCount++;
6.    }
```

إضافة دالة عضو جديدة

لإضافة دالة عضو جديدة إلى التصنيف **CWelcomeDlg**، تابع معنا الخطوات الآتية:

١. انقر التصنيف **CWelcomeDlg** بزر الفأرة الأيمن ثم اختر **Add** ⟩ **Add Function** من القائمة الموضعية، يظهر معالج إضافة دالة جديدة (انظر شكل ٣-٦).



شكل ٣-٦ إضافة دالة جديدة للتصنيف

٢. اختر نوع الدالة (أى نوع المتغير الذى ستعيده الدالة) من مربع السرد **Return Type** وهو **bool** فى هذه الحالة وذلك لأن الدالة ستعيد إحدى القيمتين، صحيح **True** أو خطأ **False**.

٣. اكتب اسم الدالة فى مربع النص **Function Name** وليكن **ClickCountMessage**.

٤. إذا احتاجت الدالة إلى معاملات، قم باختيار نوع المعامل من مربع السرد **Parameter Type** ثم قم بتعيين اسم المعامل داخل مربع النص **Parameter**

Name وأخيراً انقر زر Add لإضافة المعامل إلى قائمة معاملات الدالة بمربع السرد Parameter List.

٥. اختر مستوى وصول الدالة من مربع السرد Access وهو Public فى هذه الحالة.

٦. انقر زر Finish لإغلاق نافذة المعالج وإضافة الدالة الجديدة إلى التصنيف.

٧. من التبويب Class View، لاحظ ظهور الدالة الجديدة داخل التصنيف CWelcomDlg بالجزء السفلى من نافذة التبويب. انقر الدالة الجديدة نقراً مزدوجاً، تظهر نافذة المحرر وبها نقطة الإدراج عند بداية الدالة الجديدة.

٨. قم بإضافة الكود التالى إلى الدالة:

```
1.    bool CWelcomeDlg::ClickCountMessage(void)
2.    {
3.      if (m_nClickCount == 0)
4.         {
5.            AfxMessageBox("You haven't clicked
              me",MB_ICONSTOP);
6.             return false;
7.         }
8.      if (m_nClickCount > 1)
9.         {
10.           CString str;
11.           str.Format("Press Me was pressed %d
                                    times",m_nClickCount);
12.           AfxMessageBox(str);
13.        }
14.     else
15.        {
16.           AfxMessageBox("Press Me was clicked once");
17.        }
18.   return true;
19.   }
```

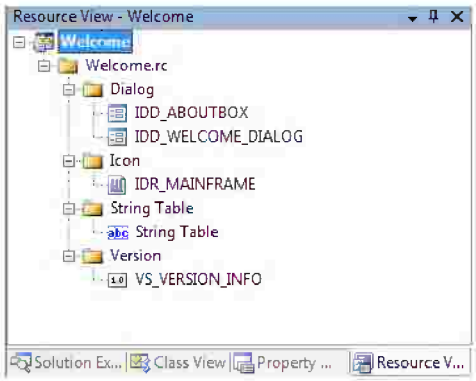
وعن هذا الكود نوضح ما يلى:

- تقوم الدالة باختبار قيمة المتغير **m_nClickCount** واستخدام هذه القيمة لإظهار الرسالة المناسبة.
- فى السطر رقم ٥ تحتوى الدالة **AfxMessageBox()** على معاملين جديدين، الأول **Don't leave** وهو عنوان مربع الرسالة والثانى **MB_ICONSTOP** والذى يتسبب فى ظهور علامة **Stop** داخل مربع الرسالة.
- فى السطر رقم ٦، تقوم الدالة بإرجاع القيمة **false** إذا كانت قيمة المتغير صفر، أى لم يتم نقر الزر **Press Me** نهائياً.
- فى السطر رقم ١٠تم تعريف المتغير **str** من النوع **CString** وهو تصنيف من تصنيفات مكتبة **MFC** يحتوى على العديد من مهام البرمجة الشهيرة الخاصة بمعالجة النصوص.
- فى السطر رقم ١١، قمنا باستخدام الدالة **Format()** والتى تشبه إلى حد كبير الدالة **Printf()** الموجودة بلغة **C**.
- فى السطر رقم ١٨، تقوم الدالة بإرجاع القيمة **true** إذا كانت قيمة المتغير أكبر من الصفر.

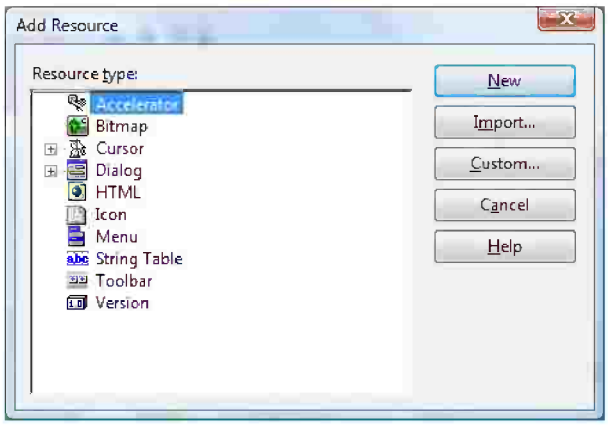
نافذة عرض الموارد Resource View

كما ذكرنا فى الفصل السابق، تقوم نافذة عرض الموارد **Resource View** بسرد جميع الموارد **Resources** الموجودة داخل المشروع، وهى العناصر المرئية الموجودة بالمشروع مثل المربعات الحوارية والقوائم والرموز. لعرض قائمة بالموارد الموجودة بالمشروع، انقر التبويب **Resource View** من نافذة بيئة التطوير ثم انقر زر التوسيع المجاور للمجلد **Welcome.rc** وكذلك المجلدات الفرعية لمشاهدة جميع موارد المشروع (انظر شكل ٣-٧). انقر المجلد **Welcome.rc** بزر الفأرة الأيمن ثم اختر **Add** ثم **Resource** من القائمة الموضعية، يظهر المربع الحوارى **Add Resource** والذى

يحتوى على جميع أنواع الموارد التى يمكن استخدامها داخل مشروعات Visual C++ والتى يمكن بيانها كما يلى (انظر شكل ٣-٨):



شكل ٣-٧ نافذة عرض الموارد



شكل ٣-٨ إضافة مورد جديد للمشروع

- *المعجل* ***Accelerator*** وهو قائمة من المترادفات بين لوحة المفاتيح والأوامر المستخدمة بالبرامج وهو ما يطلق عليه الاختصار **Shortcut** وهو أسهل طريقة لتنفيذ الأمر بدلاً من استخدام القوائم.

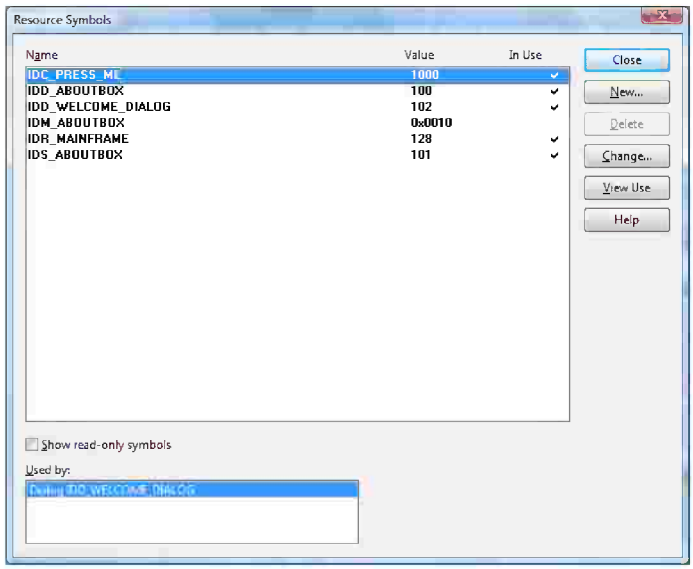
- *الصورة النقطية* **Bitmap** وهى عبارة عن صورة متعددة الأحجام تتكون من مجموعة من النقاط ولكل نقطة لون مميز عن النقطة الأخرى. ويستخدم هذا المورد لتمثيل الصور فى شرائط القوائم والصور الموجودة بالشاشة الافتتاحية للبرامج.
- *المؤشر* **Cursor** ويشبه إلى حد كبير المورد السابق فى كونه صورة ويستخدم أساساً لتمثيل صور مؤشر الفأرة.
- *الرمز* **Icon** ويستخدم أساساً لتمثيل رموز التطبيقات والمشروعات التى تقوم بتطويرها. فمشروع **Welcome** الذى بين أيدينا يحتوى على رمز يسمى **IDR_MAINFRAME**. انقر هذا الرمز نقراً مزدوجاً من المجموعة **Icon** داخل نافذة عرض الموارد، تلاحظ ظهور الرمز داخل نافذة المحرر.
- *المربع الحوارى* **Dialog** *عبارة* عن قالب لنافذة يمكنها احتواء باقى أنواع الموارد مثل القوائم والأزرار وهو ما سنتعرض له بالتفصيل فيما بعد إن شاء الله.
- *القائمة* **Menu** وتستخدم لتمثيل القوائم الإضافية للمشروع مثل تلك الموجودة أعلى النافذة الرئيسية أو القوائم الموضعية.
- *جدول سلسلة البيانات* **String Table** ويحتوى على جميع معرفات النصوص المستخدمة فى التطبيق، حيث يتم تعريف كل عنصر بمعرف فريد يستخدم للرجوع إلى النص داخل الكود (انظر شكل ٣-٩). ولعل السبب فى وضع النصوص داخل جدول وليس مباشرةً فى الملفات الأساسية هو إمكانية ترجمة هذه النصوص إلى لغة أخرى فى جدول آخر، وعند بدء التطبيق يتم استخدام الجدول المناسب.

شكل ٣-٩ جدول سلسلة البيانات

- *شريط الأدوات* **Toolbar** ويحتوى على مجموعة من الأزرار، كل زر يقوم بتمثيل أمر معين من الأوامر الموجودة بالقوائم. كما يمكن وضع صور على هـذه الأزرار وإضافة تلميحات تظهر بمجرد مرور مؤشر الفأرة على الزر.
- *الإصدار* **Version** ويحتوى على بيانات عن التطبيق مثل اسم الشـركة المنتجـة واسم التطبيق ورقم الإصدار.

يمكنك فى أى وقت إضافة موارد جديدة للمشروع، حيث يخصص لكـل مـورد جديد اسم فريد يعبر عن هذا المورد. فعند إضافة مربع حوارى جديد مثلاً، يخصـص لـه الاسم **IDD_DIALOG1**، إلا أنه ينبغى تغيير هذا الاسم ليعبر عن المهام المنوطة بـالمورد الجديد حتى يمكنك الرجوع إليه بسهولة داخل الكود، وهذا ما فعلناه فى المثال الذى بـين أيدينا حينما قمنا بتغيير اسم الزر من **IDC_BUTTON1** إلى **IDC_PRESS_ME**.

لعرض أسماء جميع الموارد الموجودة بالمشروع، انقر المجلد **Welcome.rs** بزر الفأرة الأيمن ثم اختر **Resource Symbols** من القائمـة الموضـعية، يظهـر المربـع الحـوارى **Resource Symbols** والذى يمكنك من خلاله البحث عن محتويـات مـورد معـين باستخدام اسمه عن طريق اختياره من القائمة ثم نقر زر **View Use** (انظر شكل ٣-١٠).



شكل ٣-١٠ عرض الموارد الموجودة بالمشروع

لتعديل أيٍ من الموارد الموجودة بالمشروع، انقر اسم المورد نقراً مزدوجاً من نافذة عرض الموارد Resource View، يظهر المورد داخل نافذة المحرر ومن ثمَّ يمكنك تعديله بسهولة.

لتكملة المثال الذى نحن بصدد الحديث عنه فى هذا الفصل، تابع معنا الخطوات الآتية:

١. انقر المربع الحوارى IDD_WELCOM_DIALOG نقراً مزدوجاً، يظهر قالب المربع الحوارى داخل نافذة المحرر.

٢. انقر زر Ok نقراً مزدوجاً، يتم إنشاء هيكل دالة لحدث نقر الزر باسم OnBnClickedOk() داخل نافذة الكود.

٣. قم بتعديل كود الدالة OnBnClickedOk() كما يلى:

```
void CWelcomeDlg::OnBnClickedOk()
{
        // TODO: Add your control notification handler code here
        if (ClickCountMessage() == TRUE )
        {
```

```
            OnOK();
        }
}
```

٤. من شريط التبويبات العلوى، انتقل إلى نافذة قالب المربع الحوارى IDD_WELCOM_DIALOG مرةً أخرى.

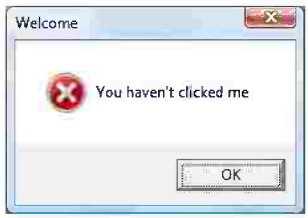
٥. انقر زر Cancel نقراً مزدوجاً، يتم إنشاء هيكل دالة لحدث نقر الزر باسم OnBnClickedCancel() داخل نافذة الكود.

٦. قم بتعديل كود الدالة OnBnClickedCancel() كما يلى:

```
void CWelcomeDlg::OnBnClickedCancel()
{
        // TODO: Add your control notification handler code here
        if (ClickCountMessage() == TRUE )
         {
           OnCancel();
         }
}
```

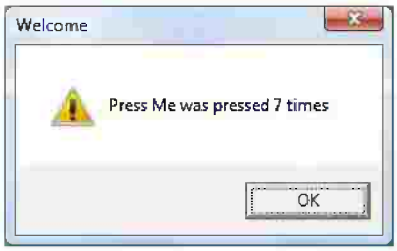
وبهذا نكون قد أضفنا دالتين، الأولى تغطى الدالة الافتراضية CDialog::OnOk() والثانية تغطى الدالة CDialog::OnCancel()، حيث تتسبب كلٌ من الدالتين فى إغلاق المربع الحوارى حينما ترجع الدالة ClickCountMessage() القيمة TRUE.

لتجربة التعديلات الجديدة على المشروع بنفسك، قم ببناء المشروع وتشغيله. انقر زر Ok أو زر Cancel دون النقر على زر Press Me، تظهر لك الرسالة الموضحة بشكل ٣-١١وسيستمر البرنامج فى العمل.



شكل ٣-١١ الرسالة التى تظهر عند عدم نقر الزر Press Me

انقر زر **Press Me** عدة مرات ثم انقر زر **Ok** أو زر **Cancel**، تظهر رسالة تخبرك بعدد المرات التي قمت فيها بنقر زر **Press Me** (انظر شكل ٣-١٢). انقر زر **Ok** لإغـــلاق الرسالة والتطبيق معاً.

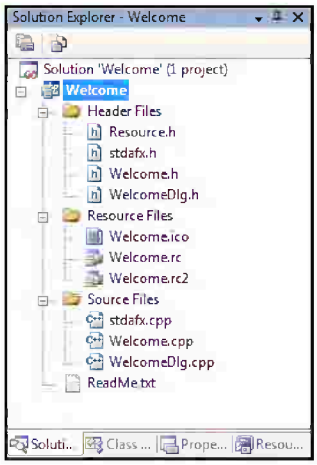


شكل ٣-١٢الرسالة التي تظهر عند نقر الزر **Press Me** عدة مرات

نافذة مستكشف الحل Solution Explorer

ذكرنا فيما سبق مصطلحين، أحدهما المشروع **Project** والآخر الحل **Solution**، فمـــا الفرق بين الاثنين ؟. يحتوى المشروع حقيقةً على جميع الملفات بينما يحتوى الحـــل علـــى مشروع واحد أو أكثر وهو بذلك يشبه إلى حد كبير مســـاحة العمـــل **Workspace** الموجودة فى الإصدارات القديمة من **Visual C++**.

تقوم نافذة مستكشف الحل بسرد الملفات المستخدمة فى بناء المشروع (أو المشروعات إذا احتوى الحل على أكثر من مشروع). لعرض قائمة بالملفات الموجودة بالمشـــروع، انقـــر التبويب **Solution Explorer** من نافذة عمل المشروع لتنشيطه (انظر شكل ٣-١٣).



شكل ٣-١٣ نافذة مستكشف الحل

بالنظر إلى النافذة، تلاحظ ترتيب الملفات داخل مجلدات منفصلة تبعاً لنوع الملف. هذه المجلدات لا تعبر بالطبع عن أماكن الملفات على القرص الصلب، كما أن لكل ملف من هذه الملفات امتداده الخاص الذى يعبر عن نوعه. فمثلاً الملفات ذات الامتداد rc تقوم بتخزين معلومات عن الموارد، لذا عند فتحها تظهر نافذة عرض الموارد.

يقوم جدول ٣-٣ التالى بعرض أنواع الملفات الشهيرة المستخدمة داخل المشروعات.

جدول ٣-٣ أنواع الملفات المستخدمة داخل المشروعات

النوع	امتداد الملف	الاستخدام
ملف الحل Solution	.sln	يعبر عن الحل Solution الذى يحتوى على مشروع أو أكثر
ملف مشروع Project File	.vcproj	تخزين تفاصيل مشروع
ملف عناوين Header File	.h	يحتوى على الكود اللازم لتمثيل التصنيفات
ملف مصدر	.cpp	يحتوى على كود تمثيل المشروع

النوع	امتداد الملف	الاستخدام
Source File		
ملف مورد Resource File	.rc	تخزين تفاصيل الموارد مثل المربعات الحوارية أو القوائم
ملف مورد Resource File	.rc2	يستخدم لتضمين الموارد داخل أكثر من مشروع
ملف صورة نقطية Bitmap File	.bmb	يستخدم لتخزين الصور النقطية
ملف رمز Icon File	.ico	يستخدم لتخزين الرموز

إذا قمت بفتح مجلد المشروع داخل مستكشف Windows، تلاحظ وجود أنواع أخرى غير الأنواع السابقة مثل الملفات المؤقتة Temporary Files المستخدمة من قِبَل نظام التشغيل، بالإضافة إلى بعض الملفات الأخرى التى تعتمد على نوع المشروع ومحتوياته.

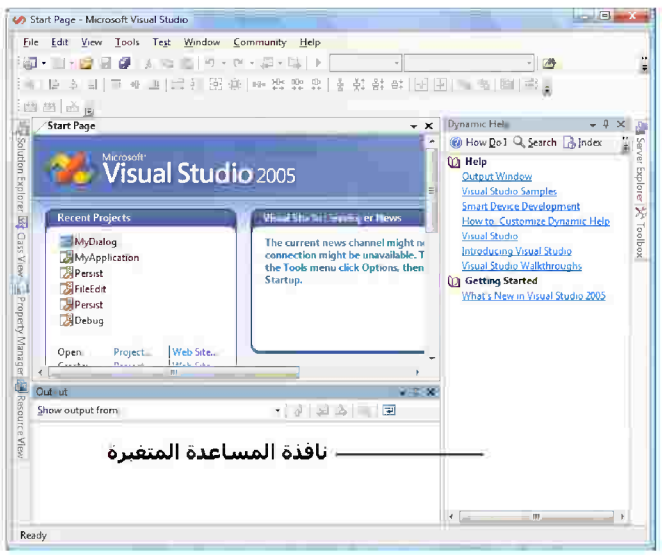
يمكنك من نافذة مستكشف الحلSolution Explorer استخدام القوائم الموضعية كما سبق لأداء العديد من المهام على ملفات معينة دون الأخرى، أو إضافة ملفات من مشروعات أخرى إلى المشروع الحالى.

السمات الهامة داخل بيئة تطوير Visual Studio 2005

كما قلنا فإن بيئة التطوير Visual Studio 2005 مبنية أساساً على عدد من السمات التى تساعدك على تطوير تطبيقاتك بسهولة تامة، حيث يمكن إجمال أهم هذه السمات فيما يلى:

- *إظهار المعلومات الصحيحة فى الوقت المناسب*. فحينما تقوم على سبيل المثال بفتح بيئة التطوير دون أن تفتح أيٍ من التطبيقات الموجودة على حاسبك، فإن بيئة التطوير تقوم بإظهار معلومات عن كيفية إنشاء التطبيقات الجديدة كما تقوم

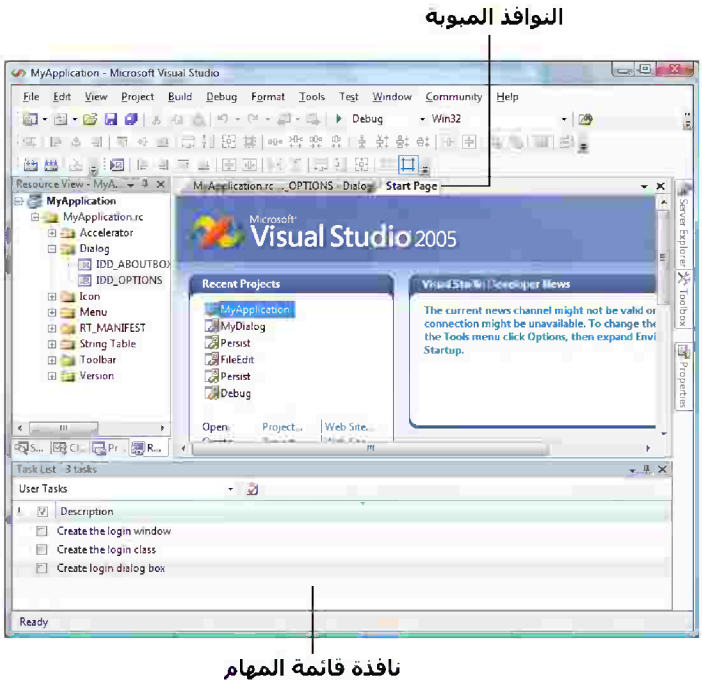
بإمدادك بمعلومات عن القوالب والمعالجات المتاحة التى يمكنك استخدامها فى إنشاء تطبيقاتك. وبمجرد البدء فى إنشاء التطبيق، يتم إظهار المعلومات المناسبة التى تعتمد على موقعك داخل التطبيق وعلى الأداة التى تقوم باستخدامها. كل هذه المعلومات تظهر داخل نافذة المساعدة المتغيرة **Dynamic Help Window** (انظر شكل ٣-١٤).



شكل ٣-١٤ الحصول على المعلومات من خلال نافذة المساعدة المتغيرة

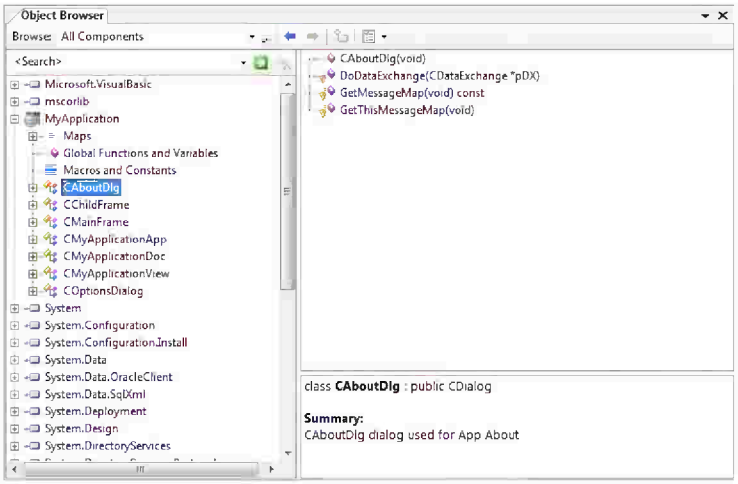
- *محرر صفحات الويب المرئى* وهو محرر يعمل بتقنية "ما تراه هو ما تحصل عليه" أو **WYSIWYG** حيث يمكنك من خلال هذا المحرر استخدام الواجهة الرسومية لإنشاء صفحات الويب دون الحاجة إلى معرفة كود **HTML** أو كود المستند البرمجى **Script**. وحقيقةً فإن هذا المحرر يحتوى على العديد من السمات التى تمكنك من إنشاء صفحات الويب بسهولة مثل الإكمال التلقائى للعبارات واختبار صيغة **XML** أثناء التشغيل والتحكم فى وضع العناصر على الصفحة.
- *نافذة قوائم المهام* ***Task Lists*** التى يمكنك من خلالها إضافة المهام التى ترغب فى تنفيذها داخل الكود. فأثناء كتابتك للكود قد تواجهك بعض المهام التى ترغب فى

تنفيذها فى أوقاتٍ أخرى. وهذه القوائم تعمل فى الواقع كمعلومات مفيدة لمطور التطبيق والأعضاء الجدد الذين يستخدمون هذا التطبيق حيث تحتوى على معلومات مفيدة عن حالة الكود الموجود داخل التطبيق. فإذا أردت التعرف على الكود المصاحب لأحد التلميحات الموجودة فى قائمة المهام، انقر هذا التلميح نقراً مزدوجاً (انظر شكل ٣-١٥).



شكل ٣-١٥ استخدام قوائم المهام لتعيين تلميحات لأجزاء معينة من الكود

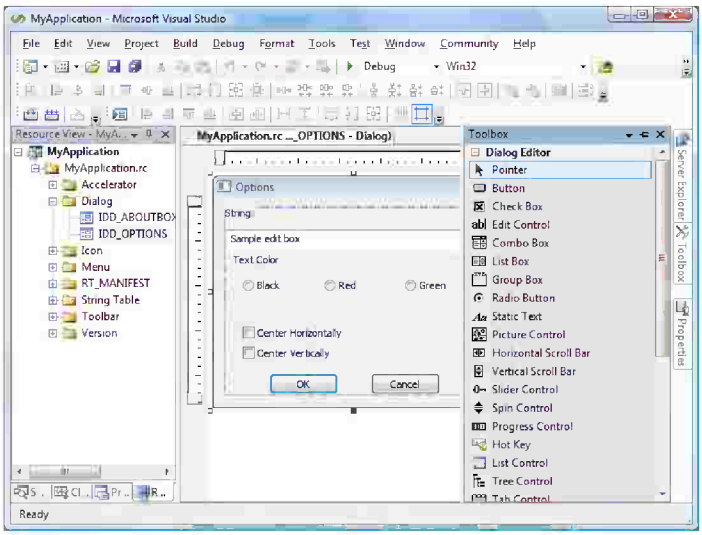
- *مستعرض الكائنات* ***Object Browser*** ويقوم بإظهار خريطة بجميع الكائنات الموجودة بالنظام. كما يمكنك من خلاله البحث عن الكائنات وترتيبها أو حتى تجميعها (انظر شكل ٣-١٦).



شكل ٣-١٦ يقوم مستعرض الكائنات بإظهار معلومات عن جميع الكائنات الموجودة بالتطبيق بغض النظر عن اللغة المستخدمة لإنشائها

- *النوافذ المبوَّبة* **Tabbed Windows** والتى تتيح للنوافذ المفتوحة الارتصاص فوق بعضها البعض عن طريق شريط من التبويبات يوجد أعلى النوافذ أو أسفلها. يمكنك استخدام هذا الشريط لتنشيط النافذة التى ترغب فى إظهارها (راجع شكل ٣-١٦).
- *سمة الإخفاء التلقائى للنوافذ* **Auto-hide** والتى يمكنك من خلالها الاحتفاظ بأكبر مساحة للشاشة. وهذه السمة مفيدة جداً خاصةً مع زيادة كمية النوافذ والسمات الموجودة داخل بيئة التطوير الجديدة، حيث يمكنك من خلالها إظهار النوافذ حسب الحاجة. فإذا انتهيت على سبيل المثال من استخدام مربع الأدوات، يتم تقليص هذه النافذة إلى إحدى حواف الشاشة. فإذا أردت إظهارها مرةً أخرى، قم بتوجيه المؤشر إلى النافذة المقلصة، تلاحظ تمدد النافذة وظهورها بحجمها السابق (انظر شكل ٣-١٧). لتطبيق سمة الإخفاء التلقائى على إحدى النوافذ، انقر الزر المجاور لزر إغلاق النافذة حيث يتحول الزر إلى الرمز . فإذا أردت تطبيق

سمة الإخفاء التلقائى على جميع النوافذ، افتح قائمة **Window** من شريط القوائم ثم اختر **Auto Hide All** من القائمة المنسدلة.

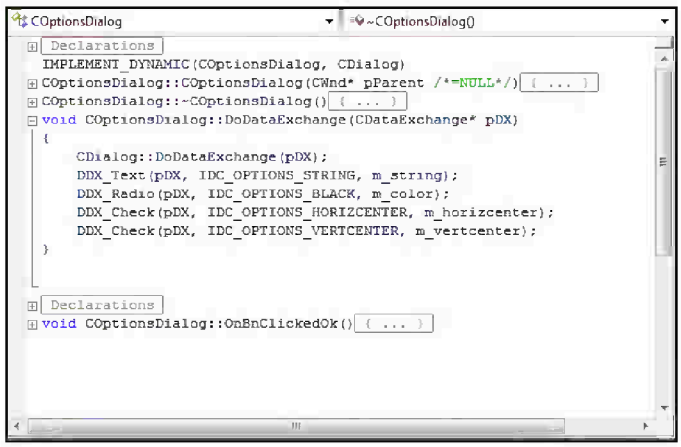


شكل ٣-١٧ الإخفاء التلقائى للنوافذ

- *تقليص الكود* **Code Collapsing** إذا قمت بفتح أحد المشروعات التى تم إنشاؤها من قبل، فربما رأيت عمل طريقة عرض الكود بإيجاز أو ما يسمى **Outlining** وهى نفس طريقة العرض **Outline View** المستخدمة فى برنامج **Word** وفيها يمكنك توسيع أو تقليص أى جزء أو كتلة من الكود بنقر زر + أو زر – على الترتيب (انظر شكل ٣-١٨).

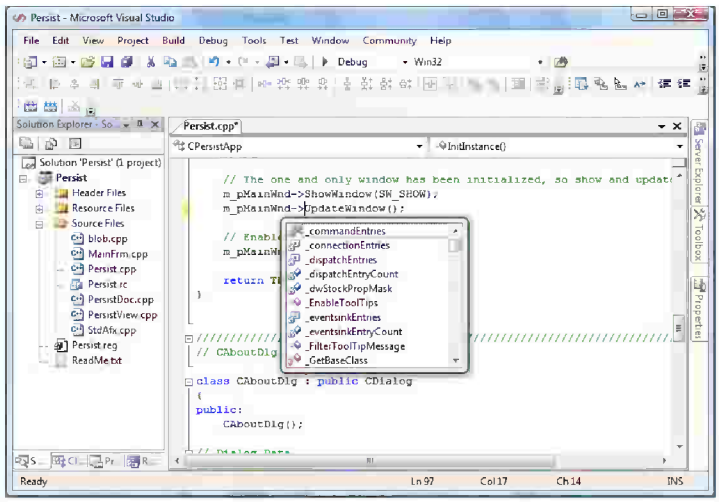
وحقيقةً لا يتم افتراضياً تنشيط طريقة عرض الكود هذه لمطورى **Visual C++** وإنما يمكنك تنشيطها بكل ملف على حدة. لتعطيل هذه السمة، افتح قائمة **Edit** من شريط القوائم ثم اختر **Outlining** ثم **Stop Outlining** من القوائم المنسدلة. فإذا أردت تنشيطها مرةً أخرى، قم بتكرار نفس الطريقة ولكن مع اختيار **Collapse to Definitions**.

إذا لم يرق لك استخدام طريقة عرض الكود هذه، يمكنك تعطيلها مرةً أخـــرى. لأداء ذلك، افتح قائمة Edit ثم اختر Outlining وStop Outlining مـــن القوائم المنسدلة.

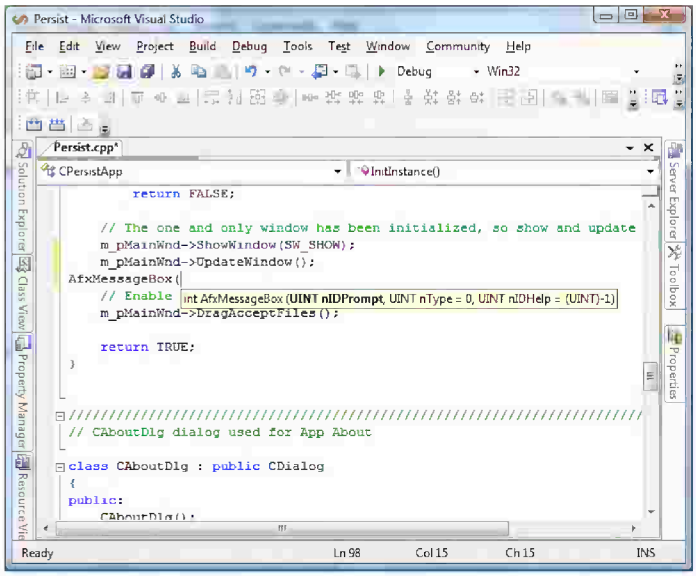


شكل ٣-١٨طريقة إيجاز عرض الكود

- *الحاسة الذكية* **IntelliSense** التى تساعدك على تذكر الصيغة الصحيحة للدوال. فحينما تقوم بكتابة اسم أحد الكائنات، تقوم هذه السمة تلقائياً بإظهار قائمة بجميع الدوال التى تخص هذا الكائن (انظر شكل ٣-١٩). وبمجرد اختيارك لإحدى هذه الدوال ثم فتح قوس بداية الدالة، تظهر قائمة بالمعاملات المستخدمة مع هذه الدالة (انظر شكل ٣-٢٠).



شكل ٣–١٩ تظهر قائمة بالدوال والوظائف والخصائص المستخدمة مع الكائن بمجرد كتابته



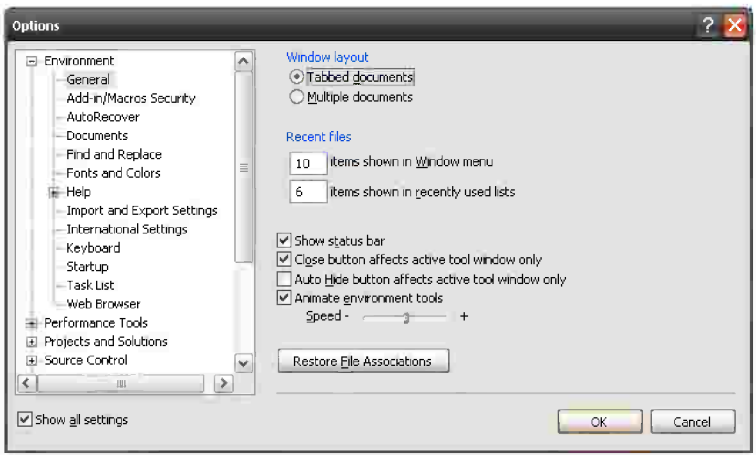
شكل ٣–٢٠ تظهر قائمة بالمعاملات المستخدمة مع الدالة بمجرد كتابة قوس بداية الدالة

ومهما كانت بيئات التطوير أو لغات البرمجة التى تعاملت معها من قبل، فإن بيئة التطوير **Visual Studio 2005** ستكون مريحةً لك إلى حد كبير حيث يمكنك توفيقها بالشـــكل الذى يروق لك كما سنرى فى البند التالى.

التحكم فى بيئة التطوير المتكاملة

يمكنك من خلال المربع الحوارى **Options** التحكم فى العديد من خيارات بيئة التطـــوير المتكاملة **Visual Studio 2005** والتى يتم إنشاء مشروعات **Visual C++** من خلالها كأن تحدد إذا ما كنت ترغب فى عرض النوافذ المفتوحة داخل المشروع فى صورة تبويبات أم فى صورة نوافذ مستقلة وإذا ما كنت ترغب فى إظهار شريط التمريـــر أم لا. كـــذلك يمكنك تحديد ما يظهر بمجرد فتح نافذة بيئة التطوير المتكاملة، يمكنك مثلاً إظهار صـــفحة البدء **Start Page** أو آخر مشروع (حل) قمت بفتحـــه أو المربـــع الحـــوارى **New Project** الذى يمكنك من خلاله إنشاء مشروع جديـــد. لإظهـــار المربـــع الحـــوارى **Options**، افتح قائمة **Tools** من شريط القوائم ثم اختر **Options** من القائمة المنسدلة (انظر شكل ٣-٢١).

سنقوم بالتعرف على خيارات المربع الحوارى **Options** فى موضع آخر مـــن الكتاب حينما تزداد خبرتك فى استخدام بيئة التطوير المتكاملة.



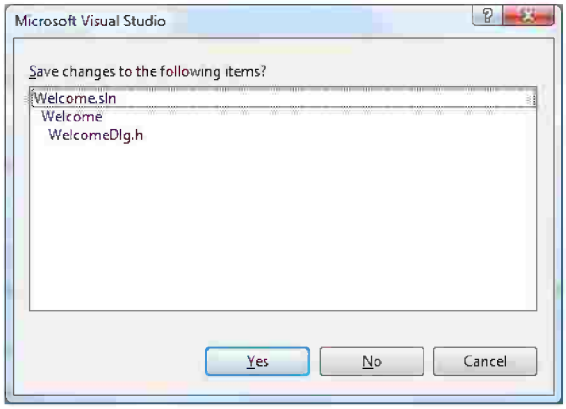
شكل ٣-٢١ المربع الحوارى **Options**.

الخروج من بيئة تطوير Visual Studio 2005

لإنهاء تطبيقاتك والخروج من بيئة التطوير المتكاملة، اتبع أيٍ من الطريقتين التاليتين:

- انقر زر الإغلاق الموجود بالركن الأيمن العلوى من نافذة بيئة التطوير المتكاملة.
- افتح قائمة **File** من شريط القوائم ثم اختر **Exit** من القائمة المنسدلة.

باتباع أيٍ من الطريقتين، تقوم بيئة تطوير **Visual Studio 2005** بتنبيهك إلى حفظ بياناتك فى حالة وجود أى بيانات غير محفوظة (انظر شكل ٣-٢٢).



شكل ٣- ٢٢ تطالبك بيئة التطوير بحفظ البيانات الجديدة

الباب الثانى
العمل مع أدوات التحكم

٤. مفهوم أدوات التحكم .

٥. استخدام أزرار التحكم .

٦. العمل مع أدوات النصوص .

٧. استخدام أدوات السرد List Controls.

٨. أدوات تحكم التاريخ والوقت.

٩. المزيد عن أدوات التحكم.