

## الفصل الثالث استخدام بيئة التطوير المتكاملة

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

بانتهاى هذا الفصل ستتعرف على:

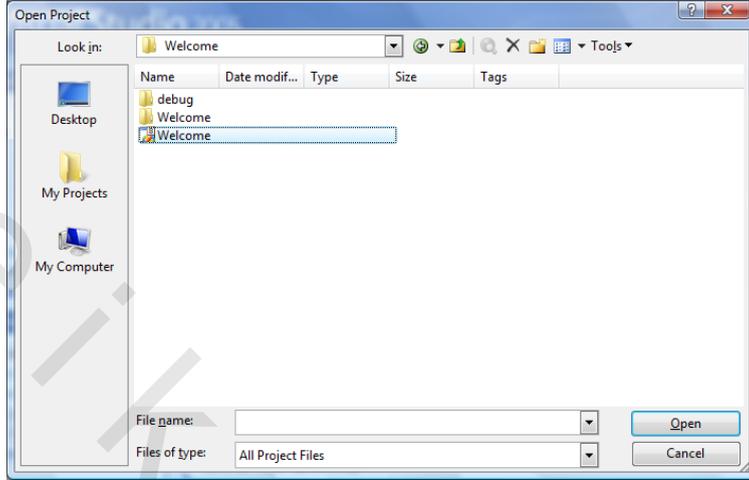
- ◆ فتح مشروع موجود.
- ◆ نافذة عرض التصنيفات **Class View**.
- ◆ نافذة عرض الموارد **Resource View**.
- ◆ نافذة مستكشف الحل **Solution Explorer**.
- ◆ السمات الهامة داخل بيئة تطوير **Visual Studio 2005**.
- ◆ توفير بيئة تطوير **Visual Studio 2005**.
- ◆ الخروج من بيئة تطوير **Visual Studio 2005**.

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 بما العديد من الخيارات الإضافية. وسنقوم أثناء عرضنا للكتاب بالتعرف على محتويات هذه البيئة بالتفصيل، لذا لا تتعجل فى معرفة كل شىء 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 أى 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

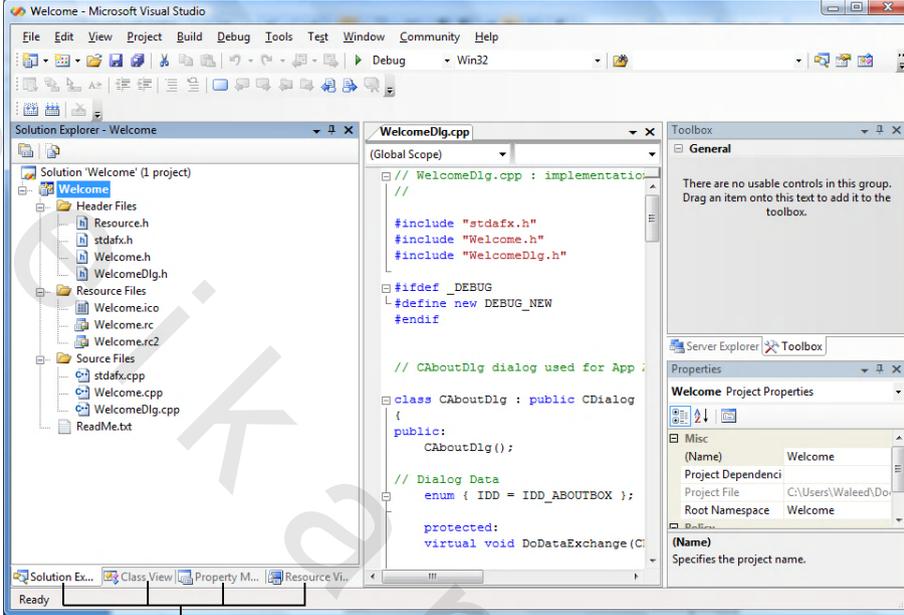
## شرح نوافذ المشروع

عند العمل في بيئة التطوير المتكاملة، تظهر لك العديد من النوافذ ولعل أهمها على

الإطلاق النوافذ الأربعة التالية التي تظهر في صورة تبويبات متجاورة:

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

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

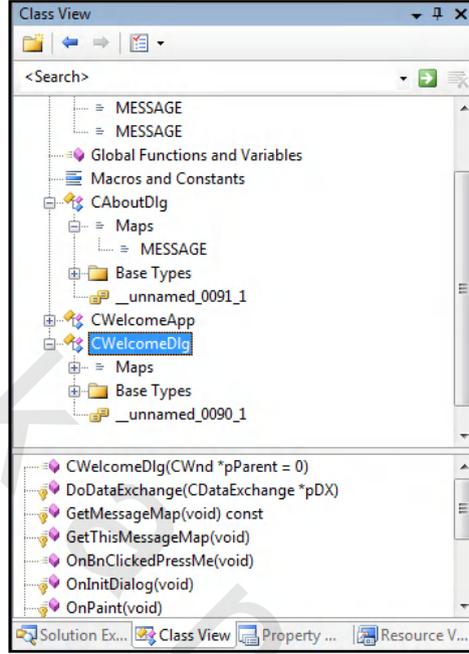


تظهر النوافذ الأربعة في صورة تبويبات متجاورة

شكل ٣-٢ النوافذ الرئيسية للمشروع

## نافذة عرض التصنيفات Class View

نشط التبويب Class View (تبويب عرض التصنيفات)، تظهر جميع التصنيفات المستخدمة في المشروع والتي قام معالج التطبيقات Application Wizard بإنشائها عند بدء إنشاء المشروع وتظهر هذه التصنيفات على شكل شجرة يمكن إظهار جميع فروعها 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 السفلي من نافذة التبويب يعبر كل منها عن دالة عضو Member function في التصنيف، أو متغير عضو Member variable في التصنيف. فمثلاً يحتوي التصنيف **CWelcomeDlg** بالجزء العلوي على الدالة **OnBnClickedPressMe()** بالجزء السفلي (راجع شكل ٣-٣) وهي الدالة التي أنشأناها في المثال **Welcome** في الفصل السابق. لاحظ ظهور رموز صغيرة يسار كل عنصر من العناصر يدل على نوعية هذا العنصر وذلك كما بالجدول ٣-١ التالي.

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

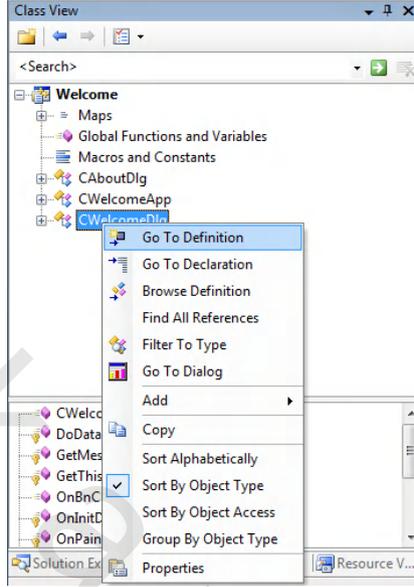
الرمز	المدلول
	تصنيف <b>Class</b>

الرمز	المدلول
	دالة عامة <b>Public function</b>
	دالة خاصة <b>Private function</b>
	دالة محمية <b>Protected function</b>
	متغير خاص <b>Private variable</b>
	متغير محمي <b>Protected variable</b>
	متغير عام <b>Public variable</b>

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 مباشرة إلى تعريف العنصر داخل التصنيف، أو استعمال القوائم الموضعية **Context menu** التي 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 التي يكثرتنفيذها عليه.

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

الفصل الثالث: استخدام بيئة التطوير المتكاملة



شكل ٣-٤ القائمة الموضوعية الخاصة بتصنيف

جدول ٣-٢ استخدام عناصر القائمة الموضوعية للتصنيف

الخيار	الاستخدام
Go To Definition	الذهاب مباشرة إلى ملف المصدر الذي يحتوي 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 مباشرة يتم الذهاب إلى تعريف التصنيف، أي نحصل على نفس نتيجة الخيار السابق
Browse Definition	فتح مستعرض الكائنات على التصنيف المختار
Find All References	البحث عن جميع الرموز والمراجع التي تتوافق مع اسم

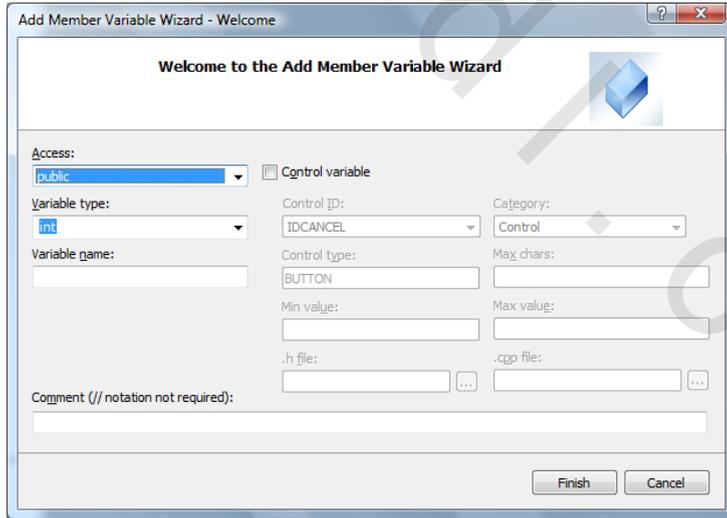
الخيار	الاستخدام
	التصنيف المختار حيث تشتمل هذه الرموز على أسماء التصنيفات والدوال والمتغيرات وغيرها، حيث يظهر الاسم ورقم السطر الذي يوجد به العنصر داخل الكود. ففي حالة التصنيف <b>CWelcomeDlg</b> ، يتم إيجاد تعريف التصنيف <b>CWelcomeDlg</b> بالإضافة إلى دالة إنشاء التصنيف <b>CWelcomeDlg</b>
<b>Filter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

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

وفيما يلي نقوم بإضافة متغير عضو في التصنيف يحتوى على عدد مرات نقر الزر **Press Me** ثم إضافة دالة عضو في التصنيف تقوم بإظهار عدد مرات نقر هذا الزر. إضافة متغير عضو جديد

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

1. انقر التصنيف **CWelcomeDlg** بزر الفأرة الأيمن ثم اختر **Add > Add Variable** من القائمة الموضوعية، 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** في هذه الحالة ثم اكتب اسم المتغير في خانة **Variable name** وليكن **m\_nClickCount** واختر درجة المتغير من مربع السرد **Access** وهو **Public** في هذه الحالة.

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

٤. من التبويب **Class View**، لاحظ ظهور المتغير الجديد داخل التصنيف **CWelcomeDlg** 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 **CWelcomeDlg**. 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:

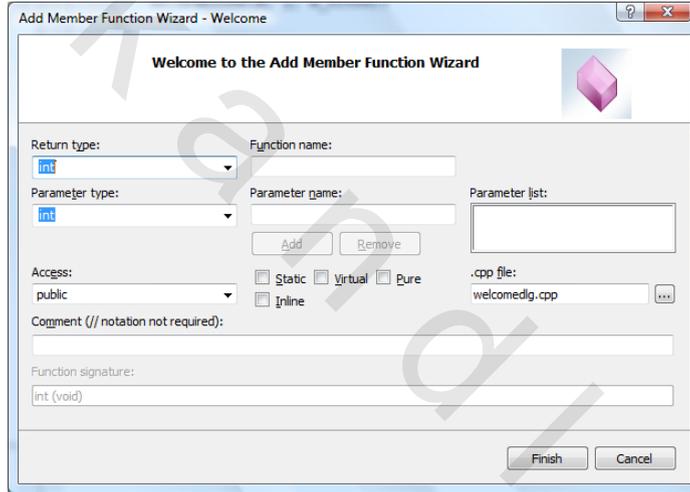
```
1. void CWelcomeDlg::OnBnClickedPressMe()
2. {
3.     // TODO: Add your control notification handler code here
```

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 الموضوعية، 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` في هذه الحالة وذلك لأن الدالة ستعيد إحدى القيمتين، صحيح `True` أو خطأ `False`.
٣. اكتب اسم الدالة في 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** في 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 **CWelcomeDlg** بالجزء السفلي من نافذة التبويب. انقر الدالة الجديدة نقراً مزدوجاً، تظهر نافذة المحرر وبها نقطة الإدراج عند بداية الدالة الجديدة.

٨. قم بإضافة الكود التالي إلى الدالة:

```

1.  bool CWelcomeDlg::ClickCountMessage(void)
2.  {
3.      if (m_nClickCount == 0)
4.      {
5.          AfxMessageBox("You haven't clicked
6.              me",MB_ICONSTOP);
7.          return false;
8.      }
9.      if (m_nClickCount > 1)
10.     {
11.         CString str;
12.         str.Format("Press Me was pressed %d
13.             times",m_nClickCount);
14.         AfxMessageBox(str);
15.     }
16.     else
17.     {
18.         AfxMessageBox("Press Me was clicked once");
19.     }
20.     return true;
21. }

```

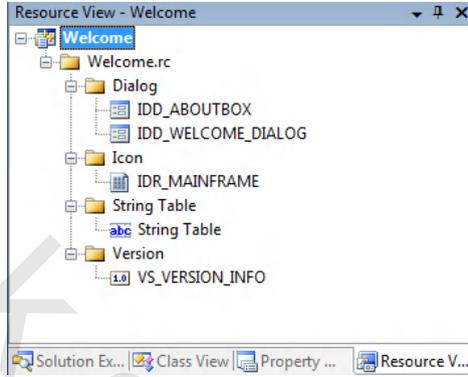
وعن هذا الكود نوضح ما يلي:

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

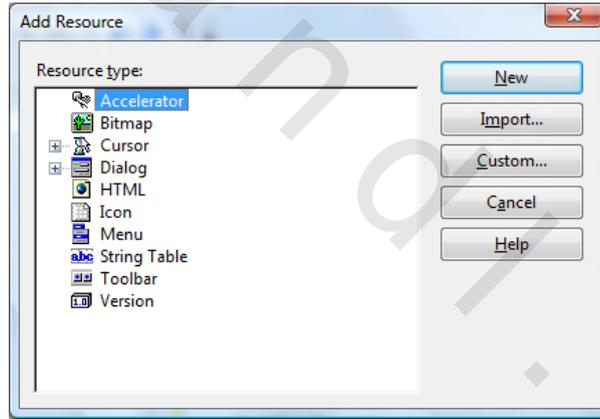
## نافذة عرض الموارد Resource View

كما ذكرنا في الفصل السابق، تقوم نافذة عرض الموارد `Resource View` بسرد جميع الموارد `Resources` الموجودة داخل المشروع، وهي 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` والذي

يحتوى على جميع أنواع الموارد التي يمكن استخدامها داخل مشروعات Visual C++ والتي يمكن بيانها كما يلي (انظر شكل ٣-٨):



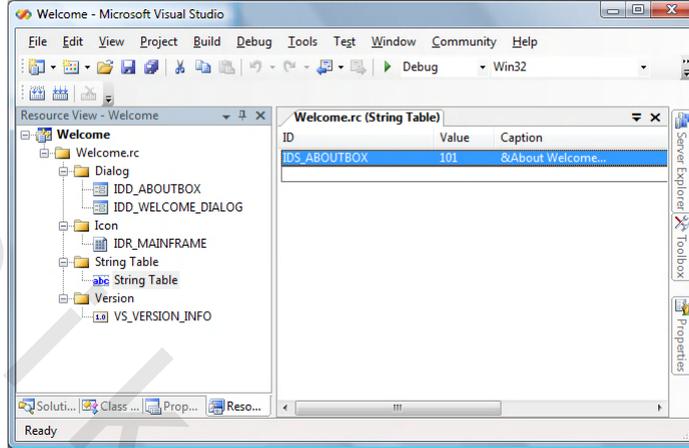
شكل ٣-٧ نافذة عرض الموارد



شكل ٣-٨ إضافة مورد جديد للمشروع

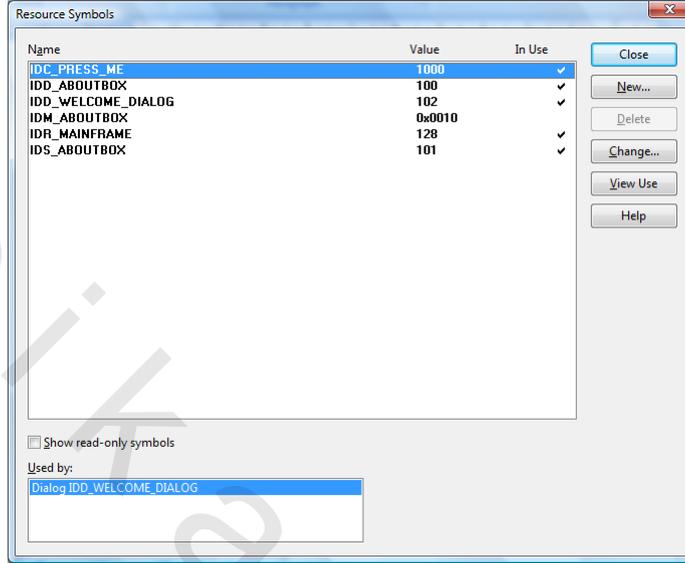
المعجل **Accelerator** وهو قائمة من المترادفات بين لوحة المفاتيح والأوامر المستخدمة بالبرامج وهو ما يطلق عليه الاختصار **Shortcut** وهو أسهل طريقة لتنفيذ الأمر بدلاً من استخدام القوائم.

- 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 نقرأ مزدوجاً من المجموعة **Icon** داخل نافذة عرض الموارد، تلاحظ ظهور الرمز داخل نافذة الخور.
- المربع الحوارى **Dialog** عبارة عن قالب لنافذة يمكنها احتواء باقى أنواع الموارد مثل القوائم والأزرار وهو ما ستعرض له بالتفصيل فيما بعد إن شاء الله.
- القائمة **Menu** وتستخدم لتمثيل القوائم الإضافية للمشروع مثل تلك الموجودة أعلى النافذة الرئيسية أو القوائم الموضوعية.
- 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 مباشرة 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 هذه الأزرار وإضافة تلميحات تظهر بمجرد مرور مؤشر الفأرة على الزر.
  - الإصدار **Version** ويحتوى على بيانات عن التطبيق مثل اسم الشركة المنتجة واسم التطبيق ورقم الإصدار.
- يمكنك في أى وقت إضافة موارد جديدة للمشروع، حيث يخصص لكل مورد جديد اسم فريد يعبر عن هذا المورد. فعند إضافة مربع حوارى جديد مثلاً، يخصص له الاسم **IDD\_DIALOG1**، إلا أنه ينبغي تغيير هذا الاسم ليحبر عن المهام المنوطة بالمورد الجديد حتى يمكنك الرجوع إليه بسهولة داخل الكود، وهذا ما فعلناه في المثال الذى بين 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 القائمة الموضوعية، يظهر المربع الحوارى **Resource Symbols** والذى يمكنك من خلاله البحث عن محتويات مورد معين باستخدام اسمه عن طريق اختياره من القائمة ثم نقر زر **View Use** (انظر شكل ٣-١٠).



شكل ٣-١٠ عرض الموارد الموجودة بالمشروع

لتعديل أي من الموارد الموجودة بالمشروع، انقر اسم المورد نقرًا مزدوجًا من نافذة عرض الموارد Resource View، يظهر المورد داخل نافذة المحرر ومن ثم يمكنك تعديله بسهولة.



1. لتكملة المثال الذي نحن بصدد الحديث عنه في هذا الفصل، تابع معنا الخطوات الآتية:  
انقر المربع الحوارى IDD\_WELCOM\_DIALOG نقرًا مزدوجًا، يظهر قالب المربع الحوارى داخل نافذة المحرر.
2. انقر زر Ok نقرًا مزدوجًا، يتم إنشاء هيكل دالة لحدث نقر الزر باسم OnBnClickedOk() داخل نافذة الكود.
3. قم بتعديل كود الدالة OnBnClickedOk() كما يلي:

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

```
OnOK();
```

```
}
```

```
}
```

٤. من شريط التبويبات العلوي، 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 يلي:

```
void CWelcomeDlg::OnBnClickedCancel()
```

```
{
```

```
    // TODO: Add your control notification handler code here
```

```
    if (ClickCountMessage() == TRUE )
```

```
    {
```

```
        OnCancel();
```

```
    }
```

```
}
```

وبهذا نكون قد أضفنا دالتين، الأولى تغطي الدالة الافتراضية **CDialog::OnOk()**

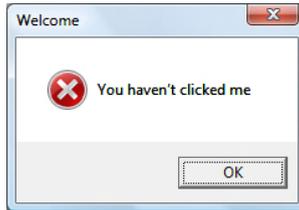
والثانية تغطي الدالة **CDialog::OnCancel()**، حيث تتسبب كلٌّ من الدالتين في إغلاق

المربع الحوارى حينما ترجع الدالة **ClickCountMessage()** القيمة **TRUE**.

لتجربة التعديلات الجديدة على المشروع بنفسك، قم ببناء المشروع وتشغيله. انقر زر **Ok**

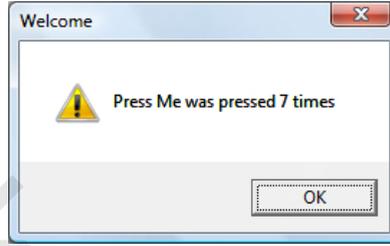
أو زر **Cancel** دون النقر على زر **Press Me**، تظهر لك الرسالة الموضحة بشكل ٣-١١-

١١ وسيستمر البرنامج في العمل.



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

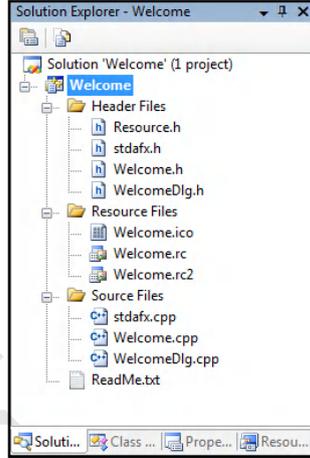
انقر زر **Press Me** عدة مرات ثم انقر زر **Ok** أو زر **Cancel**، تظهر رسالة تحريك بعدد المرات التي قمت فيها بنقر زر **Press Me** (انظر شكل ٣-١٢). انقر زر **Ok** لإغلاق الرسالة والتطبيق معاً.



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

### نافذة مستكشف الحل Solution Explorer

ذكرنا فيما سبق مصطلحين، أحدهما المشروع **Project** والآخر الحل **Solution**، فما الفرق بين الاثنين؟. يحتوي المشروع حقيقةً على جميع الملفات بينما يحتوي الحل على مشروع واحد أو أكثر وهو بذلك يشبه إلى حد كبير مساحة العمل **Workspace** الموجودة في الإصدارات القديمة من **Visual C++**. تقوم نافذة مستكشف الحل بسرد الملفات المستخدمة في بناء المشروع (أو المشروعات إذا احتوى الحل على أكثر من مشروع). لعرض قائمة بالملفات الموجودة بالمشروع، انقر 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 الذي يعبر عن نوعه. فمثلاً الملفات ذات الامتداد rc تقوم بتخزين معلومات عن الموارد، لذا عند فتحها تظهر نافذة عرض الموارد.

يقوم جدول ٣-٣ التالي 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 الذي يحتوي على مشروع أو أكثر
ملف مشروع Project File	.vcproj	تخزين تفاصيل مشروع
ملف عناوين Header File	.h	يحتوي على الكود اللازم لتمثيل التصنيفات
ملف مصدر	.cpp	يحتوي 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

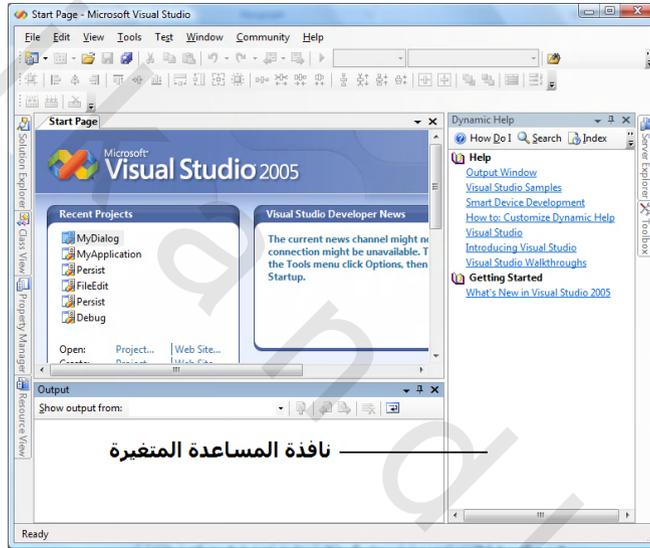
إذا قمت بفتح مجلد المشروع داخل مستكشف Windows، تلاحظ وجود أنواع أخرى غير الأنواع السابقة مثل الملفات المؤقتة Temporary Files المستخدمة من قِبل نظام التشغيل، بالإضافة إلى بعض الملفات الأخرى التي تعتمد على نوع المشروع ومحتوياته. يمكنك من نافذة مستكشف الحل Solution Explorer استخدام القوائم الموضعية كما سبق لأداء العديد من المهام على ملفات معينة دون الأخرى، أو إضافة ملفات من مشروعات أخرى إلى المشروع الحالي.

### السمات الهامة داخل بيئة تطوير Visual Studio 2005

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

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

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

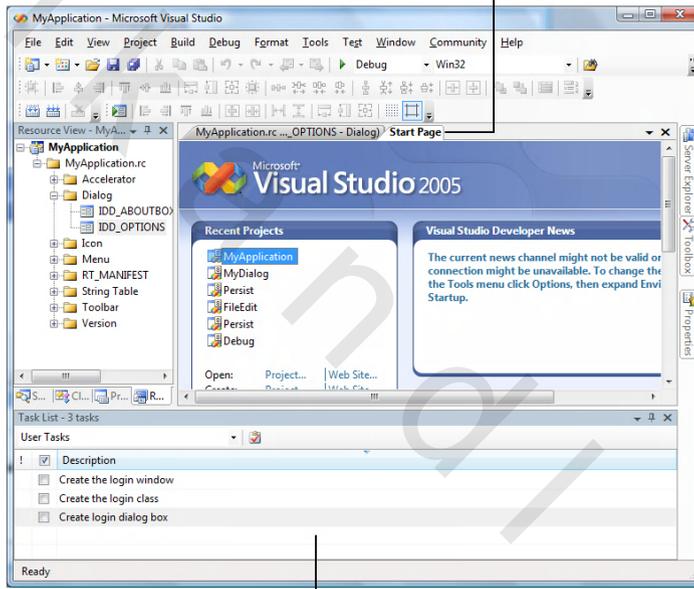


شكل ٣-١٤ الحصول على المعلومات من خلال نافذة المساعدة المتغيرة

- محرر صفحات الويب المرئي 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 البرمجي **Script**. وحقيقةً فإن هذا المحرر يحتوي على العديد من السمات التي تمكنك من إنشاء صفحات الويب بسهولة مثل الإكمال التلقائي للعبارة واختبار صيغة **XML** أثناء التشغيل والتحكم في وضع العناصر على الصفحة.
- نافذة قوائم المهام **Task Lists** التي يمكنك من خلالها إضافة المهام التي ترغب في تنفيذها داخل الكود. فإثناء كتابتك للكود قد تواجهك بعض المهام التي ترغب في

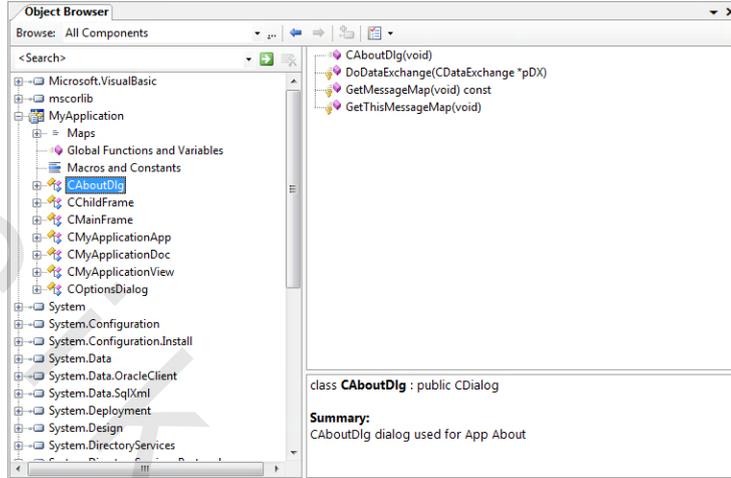
تنفيذها في أوقاتٍ أخرى. وهذه القوائم تعمل في الواقع كمعلومات مفيدة لمطور التطبيق والأعضاء الجدد الذين يستخدمون هذا التطبيق حيث تحتوي على معلومات مفيدة عن حالة الكود الموجود داخل التطبيق. فإذا أردت التعرف على الكود المصاحب لأحد التلميحات الموجودة في قائمة المهام، انقر هذا التلميح نقرأً مزدوجاً (انظر شكل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

#### النوافذ المبوبة



#### نافذة قائمة المهام

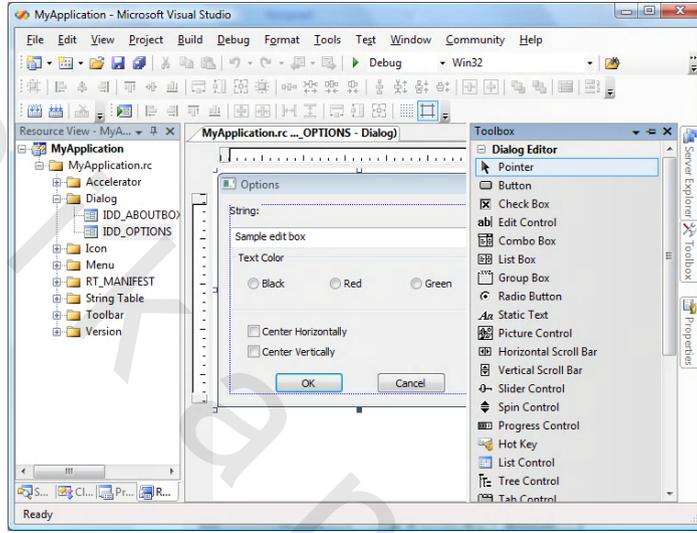
شكل ٣-١٥ استخدام قوائم المهام لتعيين تلميحات لأجزاء معينة من الكود مستعرض الكائنات *Object Browser* ويقوم بإظهار خريطة بجميع الكائنات الموجودة بالنظام. كما يمكنك من خلاله البحث عن الكائنات وترتيبها أو حتى تجميعها (انظر شكل ٣-١٦).



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

- **النوافذ الميَّونة *Tabbed Windows* والتي تتيح للنوافذ المفتوحة الارتصاص** فوق بعضها البعض عن طريق شريط من التبويبات يوجد أعلى النوافذ أو أسفلها. يمكنك استخدام هذا الشريط لتنشيط النافذة التي ترغب في إظهارها (راجع شكل ٣-١٦).
- **سمة الإخفاء التلقائي للنوافذ *Auto-hide* والتي يمكنك من خلالها الاحتفاظ** بأكبر مساحة للشاشة. وهذه السمة مفيدة جداً خاصةً مع زيادة كمية النوافذ والسماح 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 التلقائي على إحدى النوافذ، انقر الزر  المجاور لزر إغلاق النافذة حيث يتحول الزر إلى الرمز . فإذا أردت تطبيق

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



شكل ٣-١٧ الإخفاء التلقائي للنوافذ

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

وحقيقةً لا يتم افتراضياً تنشيط طريقة عرض الكود هذه لمطوري **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 أخرى. لأداء ذلك، افتح قائمة **Edit** ثم اختر **Outlining** و **Stop Outlining** من القوائم المنسدلة.



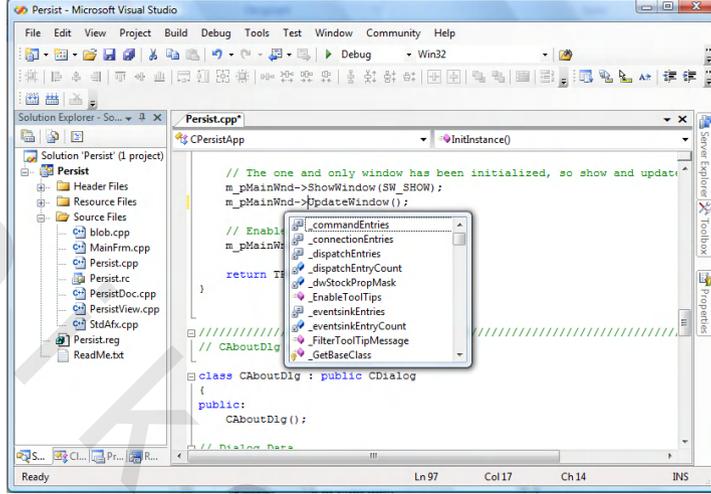
```

COptionsDialog
    = ~COptionsDialog()
    Declarations
    IMPLEMENT_DYNAMIC(COptionsDialog, CDialog)
    COptionsDialog::COptionsDialog(CWnd* pParent /*=NULL*/) { ... }
    COptionsDialog::~COptionsDialog() { ... }
    void COptionsDialog::DoDataExchange(CDataExchange* pDX)
    {
        CDialog::DoDataExchange(pDX);
        DDX_Text(pDX, IDC_OPTIONS_STRING, m_string);
        DDX_Radio(pDX, IDC_OPTIONS_BLACK, m_color);
        DDX_Check(pDX, IDC_OPTIONS_HORIZCENTER, m_horizcenter);
        DDX_Check(pDX, IDC_OPTIONS_VERTCENTER, m_vertcenter);
    }
    Declarations
    void COptionsDialog::OnBnClickedOk() { ... }
    
```

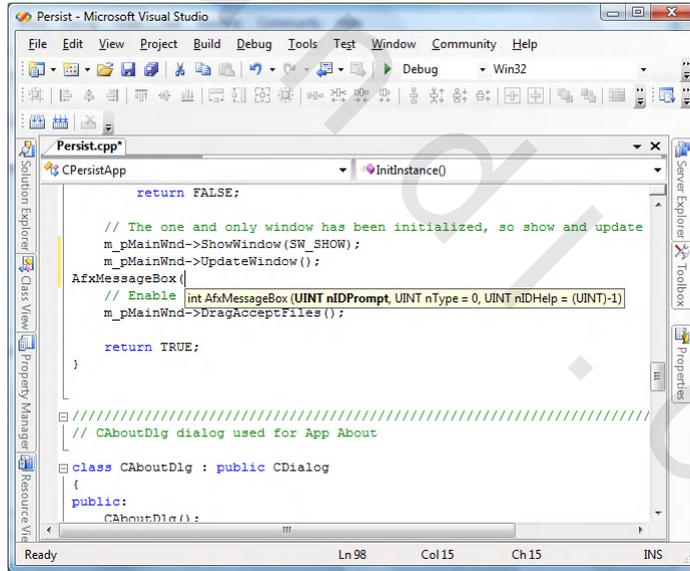
شكل ٣-١٨ طريقة إيجاز عرض الكود

- الحاسبة الذكية **IntelliSense** التي تساعدك على تذكر الصيغة الصحيحة للدوال. فحينما تقوم بكتابة اسم أحد الكائنات، تقوم هذه السمة تلقائياً بإظهار قائمة بجميع الدوال التي 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).

## الفصل الثالث: استخدام بيئة التطوير المتكاملة



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

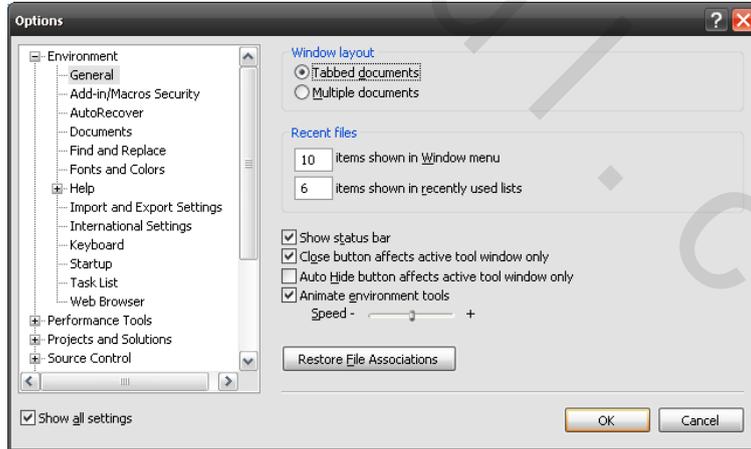


شكل ٣-٢٠ تظهر قائمة بالمعاملات المستخدمة مع الدالة بمجرد كتابة قوس بداية الدالة ومهما كانت بيئات التطوير أو لغات البرمجة التي تعاملت معها من قبل، فإن بيئة التطوير Visual Studio 2005 ستكون مريحة لك إلى حد كبير حيث يمكنك توفيقها بالشكل الذي يروق لك كما سنرى في البند التالي.

## التحكم في بيئة التطوير المتكاملة

يمكنك من خلال المربع الحوارى Options التحكم في العديد من خيارات بيئة التطوير المتكاملة Visual Studio 2005 والتي يتم إنشاء مشروعات Visual C++ من خلالها كأن تحدد إذا ما كنت ترغب في عرض النوافذ المفتوحة داخل المشروع في صورة تبويبات أم في صورة نوافذ مستقلة وإذا ما كنت ترغب في إظهار شريط التمرير أم لا. كذلك يمكنك تحديد ما يظهر بمجرد فتح نافذة بيئة التطوير المتكاملة، يمكنك مثلاً إظهار صفحة البدء Start Page أو آخر مشروع (حل) قمت بفتحه أو المربع الحوارى New Project الذى يمكنك من خلاله إنشاء مشروع جديد. لإظهار المربع الحوارى Options، افتح قائمة Tools من شريط القوائم ثم اختر Options من القائمة المنسدلة (انظر شكل ٣-٢١).

سنقوم بالتعرف على خيارات المربع الحوارى Options في موضع آخر من الكتاب حينما تزداد خبرتك في استخدام بيئة التطوير المتكاملة.



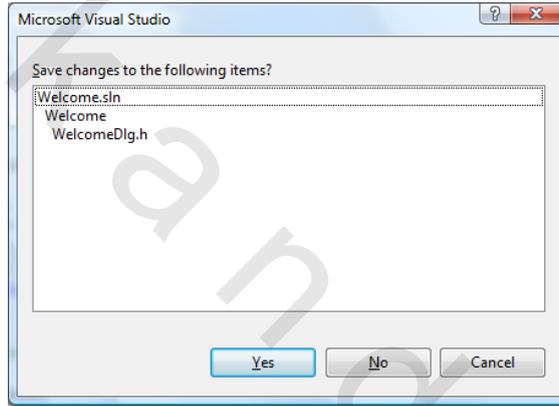
شكل ٣-٢١ المربع الحوارى Options.

## الخروج من بيئة تطوير Visual Studio 2005

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

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

باتباع أي من الطريقتين، تقوم بيئة تطوير **Visual Studio 2005** بتبنيهاك إلى حفظ بياناتك في حالة وجود أي بيانات غير محفوظة ( انظر شكل ٣-٢٢).



شكل ٣-٢٢ تطالبك بيئة التطوير بحفظ البيانات الجديدة



obeykandi.com

## المبابة الثانية

### 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 .