الفصل الثالث والعشرون

تطبيق متكامل

تعرضنا فيما سبق من فصول الكتاب إلى العديد من السمات والتقنيات المختلفة المستخدمة فى تطوير التطبيقات من خلال لغة **Visual C++ 2005**. ولأننا نؤمن إيماناً لا شك فيه أن التطبيق العملى هو الوسيلة المثلى للفهم، فقد قمنا بضرب العديد من الأمثلة البسيطة والتى تتناسب مع المفاهيم التى يتم شرحها فى كل فصل. وها نحن هنا نقوم بإنشاء تطبيق متكامل من البداية إلى النهاية لتغطية معظم المفاهيم التى تعرضنا لها من خلال فصول هذا الكتاب على أن نستكمل المسيرة معاً من خلال كتابنا "البرمجة المتقدمة باستخدام **Visual C++ 2005**".

سنقوم فى هذا الفصل كما أوضحنا بلم شتات المفاهيم التى تعرضنا لها من قبل من خـــلال فصول هذا الكتاب، حيث نتعرف على كيفية إضافة قائمة إلى التطبيق متعـــدد الوثيقـــة وكيفية إضافة العناصر إلى هذه القائمة وكيفية إضافة مربع حوارى إلى التطبيق وكـــذلك إظهار المعلومات على الشاشة بتنسيق معين يعتمد اعتماداً كلياً على خيارات المستخدم التى يقوم بتعينها أثناء عمل التطبيق. وعلى الرغم من بساطة الشكل النهائى للتطبيق، إلا أنـــه يحتوى بحق على معظم المفاهيم التى تعرضنا لها من خلال الكتاب وبنفس أسلوبنا المعتـــاد، خطوة بخطوة، فكن معنا.

إنشاء التطبيق الجديد

يقوم التطبيق الذى بين أيدينا بطباعة رسالة على النافذة الرئيسية، حيث تحتوى الوثيقة التى يتم تخزينها على ملف بقرصك الصلب على نص الرسالة إلى جانب بعض الإعدادات التى تتحكم فى تنسيق هذه الرسالة، كما يتم تمكين المستخدم من التحكم فى هذه الإعدادات من خلال مربع حوارى يظهر بدوره من خلال أحد عناصر القائمة الجديـــدة الـــتى ســـنقوم بإضافتها. لإنشاء التطبيق الجديد من خلال معالج التطبيقـــات **Application Wizard**، تابع معنا الخطوات الآتية:

١. تأكد أنك داخل بيئة التطوير المتكاملة وإلا قم بفتحها من قائمة **Start**.

٢. من صفحة البدء، انقر الارتباط **Project** بالسطر **Create:** أو افتح قائمة **File** من شريط القوائم واختر **New** ثم **Project** من القوائم المنسدلة، يظهر المربع الحوارى **New Project**.

٣. اختر المجلد الفرعى **MFC** من المجلد **Visual C++** بالجزء **Project Type** لأننا نرغب فى إنشاء مشروع باستخدام لغة **Visual C++ 2005** ثم اختر رمز **MFC Application** من المربع **Templates**.

٤. اكتب اسماً مناسباً للمشروع الجديد وليكن **MyApplication** فى مربع النص **Name**. قم أيضاً بتحديد المجلد الذى سيحتوى على جميع ملفات المشروع داخل مربع النص **Location**.

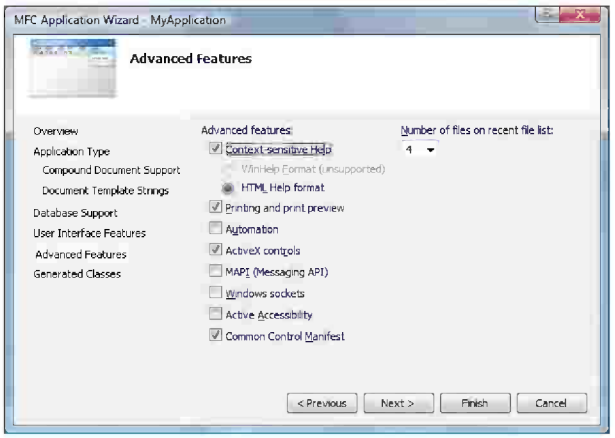
٥. انقر زر **Ok** ليبدأ معالج التطبيقات **Application Wizard** فى إنشاء هيكل المشروع باستخدام مكتبة **MFC** وكتابة الكود المطلوب نيابةً عنك وذلك بعد أن تحدد له نوع التطبيق المطلوب إنشاؤه حيث تظهر نافذة المعالج التى تحتوى على عدة تبويبات لاستخدامها فى تعيين خصائص المشروع الجديد، كما تحتوى أيضاً على الإعدادات الافتراضية للمعالج داخل التبويب النشط **Overview** (انظر شكل ٢٣-١).

٦. نشط التبويب **Application Type** ثم اختر نوع التطبيق. يمكنك هنا اختيار **Single document** لإنشاء تطبيق أحادى الوثيقة أو **Multiple documents** لإنشاء تطبيق متعدد الوثيقة، وكلاهما يؤدى إلى النتائج المرجوة إلا أن التطبيق متعدد الوثيقة يبين لك سهولة إنشاء أكثر من وثيقة فى وقتٍ واحد. لذا قم بتنشيط زر الاختيار **Multiple Documents** ولا تنسى تعطيل مربع الاختيار **Use Unicode libraries**.

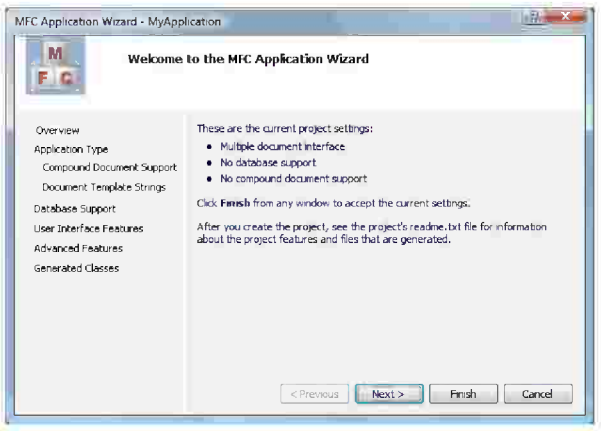
٧. لأننا لا نحتاج إلى التعامل مع قواعد البيانات ونرغب فى الإبقاء على الواجهة المعتادة للتطبيق، ابق على الإعدادات الافتراضية للتبويبات **Compound Document Support** و**Document Template Strings** و**Database Support** و**User Interface Features**.

٨. نشط التبويب **Advanced Features** ثم نشط مربع الاختيار **Context-sensitive Help** ونشط كذلك زر الاختيار **HTML Help Format** (انظر شكل ٢٣-١).

٩. لمعاينة الخصائص التى قمت بتعيينها، نشط التبويب **Overview** ثم انقر زر **Finish** لإغلاق نافذة معالج التطبيقات والبدء فى إنشاء كود التطبيق الجديد (انظر شكل ٢٣-٢).



شكل ٢٣-١ التبويب **Advanced Features** بمعالج التطبيقات



شكل ٢٣-٢ معاينة خصائص التطبيق الجديد

إظهار النصوص

يقوم التطبيق **MyApplication** كما ذكرنا بإظهار نص معين على النافـــذة الرئيســـية للتطبيق حيث يتم تخزين هذا النص داخل الوثيقة، لذا يجب أن نقوم بدايةً بإضافة متغير إلى

تصنيف الوثيقة **CMyApplicationDoc** ثم إضافة كود الحفظ والتحميل إلى الدالة **()Serialize**. يمكنك أيضاً تعيين قيمة النص الابتدائية داخل الدالة **()OnNewDocument** الموجودة بتصنيف الوثيقة والتى يتم استدعاؤها بمجرد إنشاء وثيقة جديدة. وأخيراً يتم إظهار النص من خلال الدالة **()OnDraw** الموجودة بتصنيف العرض.

إضافة المتغير وكود الحفظ والتحميل

لإضافة متغير خاص جديد باسم **string** ومن النوع **CString** إلى تصنيف الوثيقة وكذلك الدالة التى تقوم بقراءة قيمة هذا المتغير ولتكن **()GetString**، هناك طريقتان، إحداهما يدوياً والأخرى من خلال المعالج وذلك كما يلى:

أولاً الطريقة اليدوية:

لاستخدام الطريقة اليدوية، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع (نافذة مستكشف الحل) ثم انقر التصنيف **CMyApplicationDoc** نقراً مزدوجاً وحينئذٍ يظهر ملف تعريف الوثيقة وهو الملف **MyApplicationDoc.h** داخل نافذة الكود.

٢. قم بإدخال الكود التالى إلى الملف:

```
Private:
        CString string;
Public:
        CString GetString() {return string;}
```

حيث تم تعريف المتغير **string** داخل الجزء **Private** حتى لا يتم الوصول إليه إلى من قِبَل الدوال التى تنتمى لتصنيف الوثيقة ثم قمنا بتعريف الدالة **()GetString** التى تقوم بإرجاع هذا المتغير وقد قمنا بكتابة كود الدالة مباشرةً لحظة الإعلان عنها وهو ما يسمى **inline coding**.

ثانياً استخدام المعالج:

لإنشاء المتغير الجديد من خلال المعالج، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.

٢. انقر التصنيف **CMyApplicationDoc** بزر الفأرة الأيمن ثم اختر **Add** ⮞ **Add Variable** من القائمة الموضعية، يظهر معالج إضافة متغير جديد (انظر شكل ٢٣-٣).

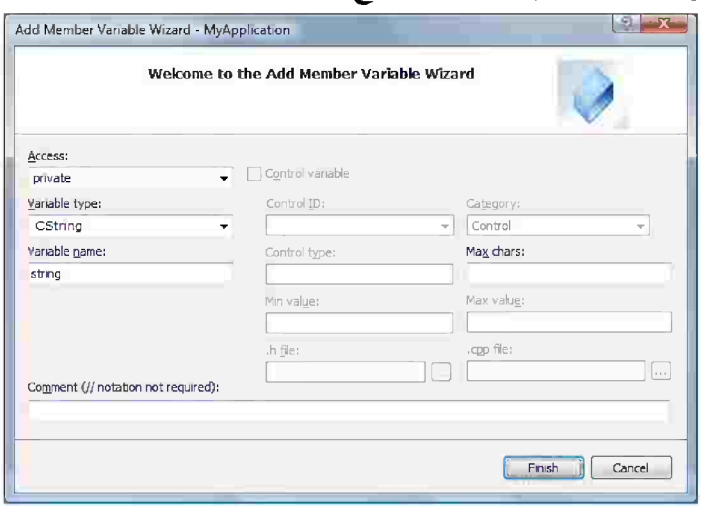
٣. اكتب نوع البيانات **CString** بمربع السرد والتحرير **Variable Type**.

٤. اكتب اسم المتغير فى مربع النص **Variable name** وهو **string** فى هذه الحالة.

٥. اختر درجة المتغير من مربع السرد **Access** وهو **private** فى هذه الحالة.

٦. يمكنك كتابة التعليق الذى يظهر بجوار تعريف المتغير وذلك داخل مربع النص **Comment**.

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



شكل ٢٣-٣ إضافة متغير جديد من خلال المعالج

ولإنشاء الدالة الجديدة من خلال المعالج، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع إذا لم يكن هو التبويب النشط.

٢. انقر التصنيف CMyApplicationDoc بزر الفأرة الأيمن ثم اختر Add≻ Add Function من القائمة الموضعية، يظهر معالج إضافة دالة جديدة (انظر شكل ٢٣-٤).

٣. قم بإدخال نوع البيانات التى تقوم الدالة بإرجاعها داخل مربع السرد والتحرير Return Type وهو CString فى هذه الحالة.

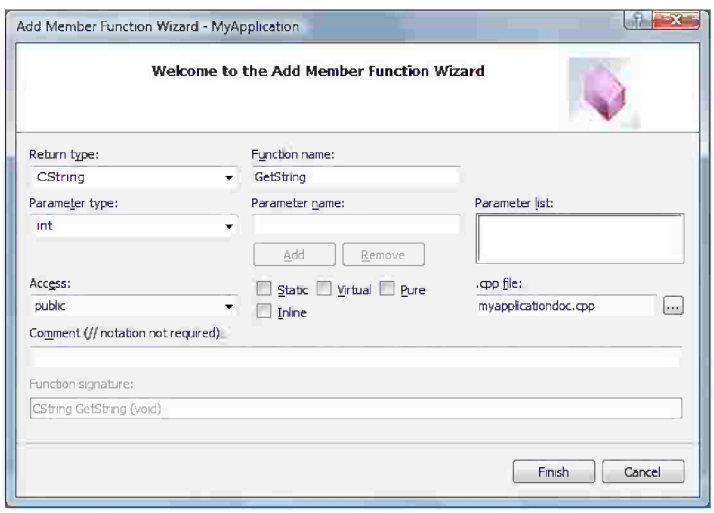
٤. قم بإدخال اسم الدالة داخل مربع النص Function name وهو GetString فى هذه الحالة.

٥. تأكد من اختيار Public داخل مربع السرد والتحرير Access.

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

٧. نشط التبويب Solution Explorer ثم قم بفتح الملف MyApplication.h وقم بإضافة الكود التالى بسطر الإعلان عن الدالة GetString():

```
{return string;}
```



شكل ٢٣-٤ معالج إضافة دالة جديدة

يتم تعيين كود تحميل وحفظ البيانات كما ذكرنا من خلال الدالة Serialize(). لأداء ذلك، تابع معنا الخطوات الآتية:

١. نشط التبويب Class View من نافذة عمل المشروع إذا لم يكن هو التبويب النشط.

٢. نشط التصنيف CMyApplicationDoc بالجزء العلوى من التبويب ثم انقر الدالة Serialize() بالجزء السفلى نقراً مزدوجاً، تظهر الدالة داخل نافذة الكود.

٣. قم بتعديل كود الدالة كما يلى:

```
void CMyApplicationDoc::Serialize(CArchive& ar)
{
	if (ar.IsStoring())
	{
		ar << string;
	}
	else
	{
		ar >> string;
	}
}
```

ويتم فى هذا الكود نقل البيانات من الأرشيف إلى المتغير string فى حالة الحفظ أو نقل البيانات من المتغير إلى الأرشيف فى حالة التحميل.

تعيين القيمة الابتدائية لسلسلة البيانات

بمجرد إنشاء وثيقة جديدة، يجب أن يقوم التطبيق بتخصيص قيمة ابتدائية للمتغير String بسلسلة بيانات معينة ولتكن عبارة "Welcome To Compuscience"، حيث يتم إنشاء الوثيقة الجديدة بمجرد اختيار المستخدم للخيار New داخل قائمة File بشريط قوائم التطبيق وذلك من خلال الرسالة التالية الموجودة داخل تصنيف التطبيق CMyApplicationApp ويتم التعامل معها من خلال الدالة CWinApp::OnFileNew():

```
BEGIN_MESSAGE_MAP(CMyApplicationApp, CWinApp)
	ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
	// Standard file based document commands
	ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
	ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
```

```
        // Standard print setup command
        ON_COMMAND(ID_FILE_PRINT_SETUP,
        CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()
```

يتم تخصيص القيمة الابتدائية للوثيقة كما ذكرنا داخل الدالـــة OnNewDocument() التى يقوم المعالج بتعيين كودها كما يلى:

```
BOOL CMyApplicationDoc::OnNewDocument()
{
        if (!CDocument::OnNewDocument())
                return FALSE;
        // TODO: add reinitialization code here
        // (SDI documents will reuse this document)

        return TRUE;
}
```

لتعديل هذا الكود، انقر الدالة OnNewDocument() نقراً مزدوجاً لفتحها داخل نافذة الكود ثم قم بإدخال هذا السطر بدلاً من التعليقات أو أسفل منها إذا أردت الإبقاء عليها:

```
string = "Welcome To Compuscience";
```

إظهار سلسلة البيانات على الشاشة

يتم استدعاء الدالة OnDraw() الموجودة بتصنيف العرض كلما احتاج هذا العـــرض إلى التمثيل كبداية التشغيل أو إعادة التحجيم أو استعادة الحجم حيث يقوم المعـــالج بإضـــافة هيكل الدالة إلى كود التطبيق كما يلى:

```
void CMyApplicationView::OnDraw(CDC* /*pDC*/)
{
        CMyApplicationDoc* pDoc = GetDocument();
        ASSERT_VALID(pDoc);

        // TODO: add draw code for native data here
}
```

وكما ترى، تحتوى الدالة على معامل واحد عبارة عن مؤشر لتصنيف بيئة العنصـــر CDC والذى يحتوى بدوره على الدالة DrawText() التى تقوم بإظهار النص علـــى الشاشـــة وتحتوى على الصيغة العامة التالية:

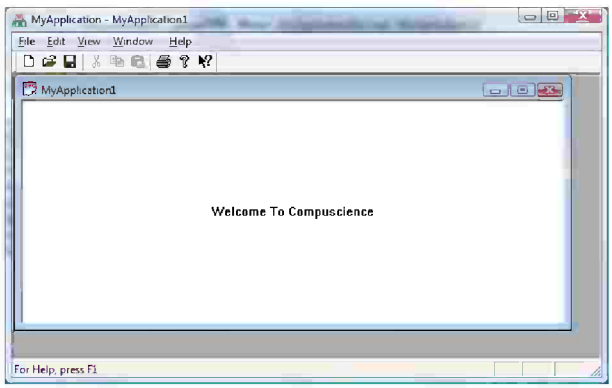
```
int DrawText( const CString& str, LPRCT lpRect, UINT nFormat )
```

وغالباً ما يكون المعامل الأول عبارة عن النص الموجود بتصنيف الوثيقة ويمكن الوصول إليه هكذا **pDoc->GetString()**. أما المعامل الثانى فيعبر عن المستطيل المحيط بالعرض ويتم الحصول عليه من خلال استدعاء الدالة **GetClientRect()**، أما المعامل الثالث والأخير فيعبر عن الطريقة التى يتم بها إظهار النص مثل توسيط النص رأسياً أو أفقياً مثلاً. قم بإضافة الكود التالى إلى الدالة **OnDraw()**:

```
CRect rect;
        GetClientRect(&rect);
        pDC->DrawText(pDoc->GetString(),&rect, DT_CENTER
                                |DTVCENTER|DT_SINGLELINE);
```

ولكن لا تنسى حذف علامات التعليق من رأس الدالة **OnDraw()**.

قم الآن ببناء التطبيق وتنفيذه، تلاحظ ظهور النص الذى قمنا بتحديده فى منتصف النافذة (انظر شكل ٢٣-٥).

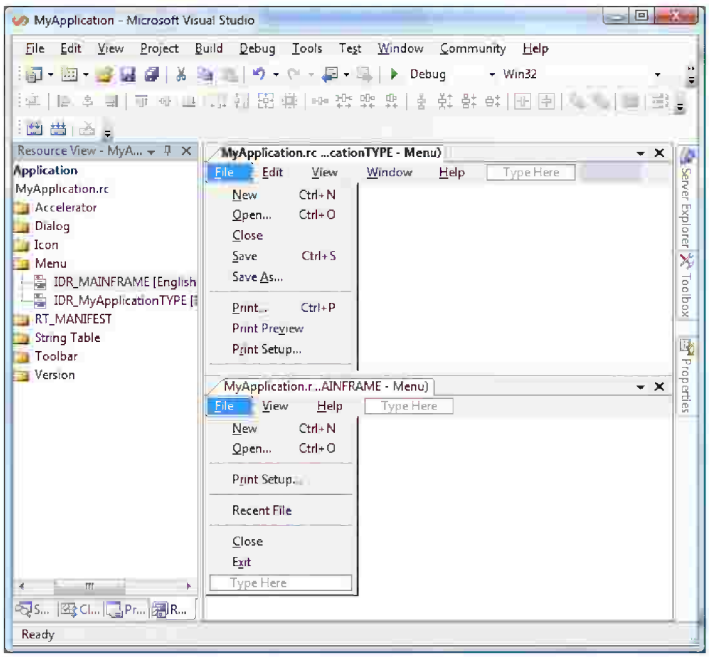


شكل ٢٣-٥ ظهور النص داخل نافذة التطبيق

إنشاء قوائم التطبيق

حينما تقوم بإنشاء تطبيق متعدد الوثيقة باستخدام معالج التطبيقات، يقوم المعالج بإضافة قائمتين إلى هذا التطبيق. إحداها **IDR_MAINFRAME** وتخص التطبيق ويتم إظهارها عند عدم فتح الملفات داخل العرض الحالى، والأخرى

IDR_MYAPPLICATIONTYPE وتظهر بمجرد فتح وثيقة جديدة (انظر شكل ٢٣-٦).



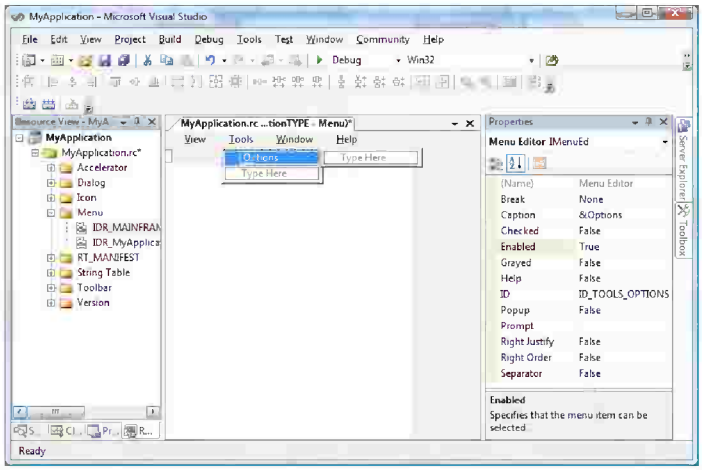
شكل ٢٣-٦ يقوم معالج التطبيقات بإضافة قائمتين إلى التطبيق

لأننا نحتاج أن يقوم المستخدم بتغيير النص المعروض وكذلك تغيير تنسيق هذا النص، فمن الممكن أن نقوم بإضافة عنصر جديد وليكن **Value** إلى قائمة **Edit** لإظهار مربع حوارى يتم من خلاله تغيير النص، وإضافة قائمة جديدة باسم **Format** تحتوى على عنصر واحد وليكن **Appearance** لإظهار مربع حوارى آخر يتم من خلاله التحكم فى مظهر النص. ولكن من الأفضل بالطبع أن نقوم بإضافة عنصر واحد باسم **Options** أسفل قائمة جديدة باسم **Tools** لإظهار مربع حوارى واحد يقوم المستخدم من خلاله بالتحكم فى مظهر النص إلى جانب القدرة على تغيير النص نفسه. ولكن هل يتم إضافة هذه القائمة للقائمتين ؟ لا بالطبع. يتم إضافتها إلى القائمة **IDR_MYAPPLICATION_TYPE** فقط لأننا نحتاج إليها عند وجود وثيقة مفتوحة بالفعل.

لإضافة القائمة الجديدة إلى التطبيق، تابع معنا الخطوات الآتية:

١. نشط تبويب عرض الموارد Resource View ثم قم بتوسيع المجلد Menu وانقر عنصر القائمة IDR_MYAPPLICATION_TYPE نقراً مزدوجاً لفتح القائمة داخل نافذة تحرير الموارد.

٢. انقر المربع الموجود يمين عنصر القائمة Help ثم اكتب &Tools واضغط مفتاح الإدخال، يظهر مربع الخصائص محتوياً على هذا العنوان داخل الخاصية Caption (انظر شكل ٢٣-٧).



شكل ٢٣-٧ إضافة عنصر القائمة الجديد

٣. اسحب العنصر الجديد وضعه بين قائمة View وقائمة Window.

٤. انقر العنصر الفرعى أسفل العنصر Tools ثم اكتب &Options لإضافة عنصر جديد أسفل قائمة Tools باسم Options. وقد استخدمنا الرمز & كى نتمكن من استخدام المفاتيح المختصرة لتنفيذ خيارات القوائم. فمن الممكن فى هذا المثال أن نستخدم الاختصار Alt+T من لوحة المفاتيح لفتح قائمة Tools.

٥. تقوم بيئة التطوير بتخصيص الاسم ID_TOOLS_OPTIONS لعنصر القائمة الجديد. يمكنك تغيير هذا الاسم من خلال الخاصية ID بمربع الخصائص.

٦. يمكنك أيضاً تعيين التلميح الذى يظهر بمجرد مرور مؤشر الفأرة على عنصر القائمة وذلك من خلال الخاصية **Prompt**. قم بتعيين تلميح مناسب وليكن **Change String and Appearance**.

وبذلك نكون قد أضفنا عنصر القائمة الجديد. يمكنك ربط عنصر القائمة باختصار من لوحة المفاتيح كالاختصار **Ctrl+C** المصاحب لعنصر القائمة **Edit,Copy** مثلاً. لأداء ذلك، تابع معنا الخطوات الآتية:

١. نشط التبويب **Resource View** من نافذة عمل المشروع.

٢. قم بتوسيع المجلد **Accelerator** ثم انقر العنصر **IDR_MAINFRAME** نقراً مزدوجاً، يظهر جدول التعجيل (الاختصارات) داخل نافذة التحرير.

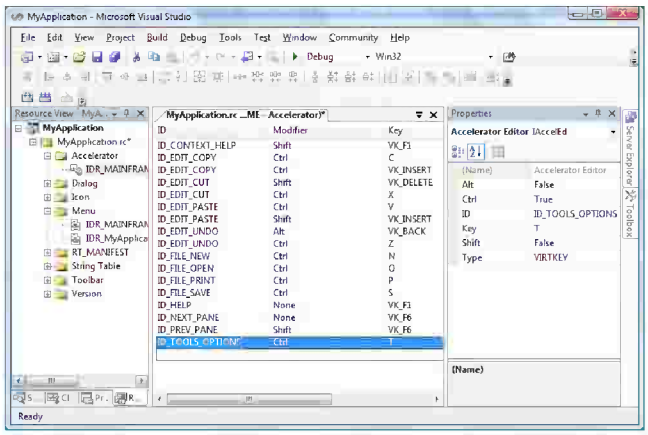
٣. انقر السطر الخالى أسفل الجدول، يتم إضافة سطر جديد.

٤. انقر العمود **ID** ثم اختر **ID_TOOLS_OPTION** من القائمة الناتجة (انظر شكل ٢٣-٨).

٥. انقر المربع الموجود أسفل العمود **Key** ثم اكتب **T** وتأكد من تخصيص القيمة **True** للخاصية **Ctrl** والقيمة **False** للخصائص **Alt** و**Shift** داخل مربع الخصائص.

٦. انقر أى سطر آخر داخل الجدول لحفظ تعديلاتك.

ولكن ماذا يحدث حينما يقوم المستخدم باختيار عنصر القائمة الجديد ؟ يتم هذا حقيقةً من خلال الكود ولكن دعنا نقوم أولاً بإضافة المربع الحوارى الذى يقوم هذا الخيار بإظهاره.



شكل ٢٣-٨ إضافة الاختصار الجديد إلى جدول الاختصارات

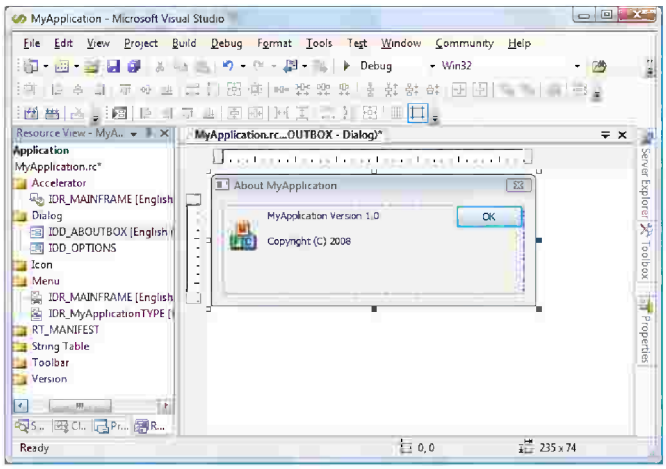
إنشاء المربعات الحوارية

يحتوى التطبيق **MyApplication** على مربعين حواريين، أحدهما المربع الحوارى **About** الموجود من قِبَل المعالج والذى يحتاج إلى بعض التعديل، والآخر هو المربع الحوارى **Options** الذى يتم إظهاره من خلال عنصر القائمة الجديد الذى قمنا بتعريفه منذ قليل وسنقوم بإنشائه من البداية.

المربع الحوارى About

يحتوى المربع الحوارى **About** الذى يقوم المعالج بإنشائه على اسم التطبيق والسنة الحالية. لمشاهدة هذا المربع، افتح نافذة **Resource View** ثم قم بتوسيع المجلد **Dialog** ومنه انقر العنصر **IDD_ABOUTBOX** نقراً مزدوجاً (انظر شكل ٢٣-٩).

لتعديل هذا المربع بإدخال اسم الشركة وليكن **Compuscience** وإضافة سطر جديد يدل على إنشاء التطبيق باستخدام لغة **Visual C++ 2005**، تابع معنا الخطوات الآتية:



شكل ٢٣-٩ المربع الحوارى About

١. انقر مربع العنوان Copyright 2008 ثم قم بإدخال اسم الشركة بنهاية النص بالخاصية Caption المصاحبة للأداة.

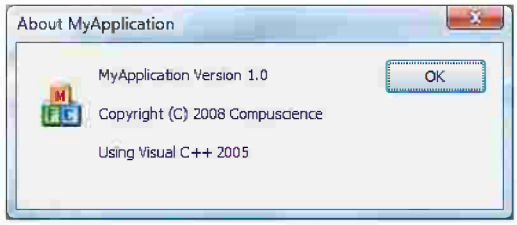
٢. قم بزيادة حجم المربع الحوارى قليلاً كى يتسع لإضافة سطر جديد.

٣. قم بإضافة أداة عنوان من مربع الأدوات إلى المربع الحوارى أسفل السطر الثانى.

٤. قم بتغيير محتويات الخاصية Caption إلى Using Visual C++ 2005.

٥. اختر سطور العناوين الثلاثة داخل المربع الحوارى وذلك باستخدام زر الفأرة الأيسر ومفتاح Ctrl ثم افتح قائمة Format من شريط القوائم واختر Align➢Lefts من القائمة المنسدلة.

٦. افتح قائمة Format مرةً أخرى ثم اختر Space Evenly➢Down من القائمة المنسدلة (يجب أن يظهر المربع الحوارى About الآن كما فى شكل ٢٣-١٠).



شكل ٢٣-١٠ المربع الحوارى About بعد التعديلات الأخيرة

المربع الحوارى Options

يحتوى المربع الحوارى Options على خيارات تحديد العبارة التى تظهر على الشاشة إلى جانب تعيين مظهر هذه العبارة. لإضافة هذا المربع إلى التطبيق، تابع معنا الخطوات الآتية:

١. نشط التبويب Resource View من نافذة عمل المشروع إذا لم يكن هو التبويب النشط.

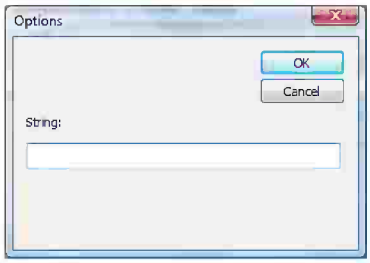
٢. انقر المجلد Dialog بزر الفأرة الأيمن ثم اختر Insert Dialog من القائمة الموضعية، يتم إضافة مربع حوارى جديد إلى المجلد وفتح هذا المربع داخل نافذة المحرر.

٣. من الخاصية ID بمربع الخصائص، قم بتغيير اسم المربع الجديد إلى IDD_OPTIONS وتغيير العنوان إلى Options من خلال الخاصية Caption.

٤. قم بإضافة مربع نص من مربع الأدوات إلى المربع الحوارى وأعد تسميته إلى IDC_OPTIONS_STRING.

٥. قم بتوسيع مربع النص إلى اليمين حتى يملأ المربع الحوارى بالكامل.

٦. قم بإضافة أداة عنوان من مربع الأدوات إلى المربع الحوارى أعلى مربع النص الذى قمت بإضافته. يجب أن يظهر المربع الحوارى الجديد الآن كما فى شكل ٢٣-١١.



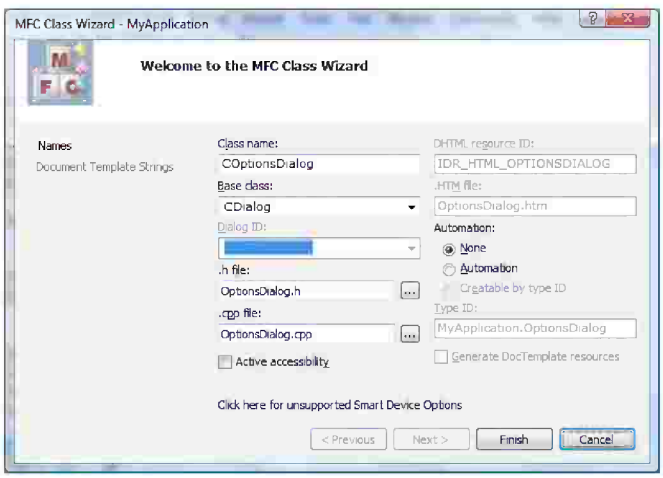
شكل ٢٣-١١ المربع الحوارى Options

إضافة الكود لعنصر القائمة الجديد

كى تتمكن من إظهار المربع الحوارى Options بمجرد اختيار المستخدم للخيار Options من قائمة Edit، يجب أن تقوم أولاً بإنشاء تصنيف المربع الحوارى. لأداء ذلك،

تابع معنا الخطوات الآتية:

١. انقر أى مكان خالى داخل المربع الحوارى **Options** بزر الفأرة الأيمن ثم اختر **Add Class** من القائمة الموضعية، تظهر نافذة معالج إضافة تصنيف جديد (انظر شكل ٢٣-١٢).



شكل ٢٣-١٢ إضافة تصنيف المربع الحوارى

٢. قم بإدخال اسم مناسب للتصنيف داخل مربع النص **Class name** على أن يبدأ بحرف **C** وليكن **COptionsDialog**.

٣. اختر **CDialog** من مربع السرد والتحرير **Base Class** ليكون تصنيفاً أساسياً للتصنيف المزمع إنشاؤه.

٤. انقر زر **Finish** لإغلاق نافذة المعالج وإضافة التصنيف الجديد إلى التطبيق.

لربط مربع النص الموجود بالمربع الحوارى بمتغير، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.

٢. انقر التصنيف **COptionsDialog** بزر الفأرة الأيمن ثم اختر **Add➢Add Variable** من القائمة الموضعية، 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 مربع النص **IDC_OPTIONS_STRING** من مربع السرد **Control ID**.

٥. اختر **Value** من مربع السرد **Category** لأننا نرغب فى التعامل مع قيمة مربع النص فقط.

٦. تأكد من اختيار نوع البيانات **CString** بمربع السرد والتحرير**Variable Type**.

٧. اكتب اسم المتغير فى خانة **Variable name** وليكن **m_string** واختر درجة المتغير من مربع السرد **Access** وهو **Public** فى هذه الحالة.

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

تعيين دالة احتواء عنصر القائمة

الخطوة التالية هى تعيين كود رسالة الأمر التى يتم إرسالها إلى التطبيق بمجرد اختيار المستخدم للخيار **Options** من قائمة **Tools** وهو ما نطلق عليه دالة احتواء عنصر القائمة. ولكن أين يتم تعريف هذه الدالة ؟ فالتطبيق يحتوى على سبعة تصنيفات، وعليك أن تقوم بتحديد التصنيف المناسب. ولأننا قمنا بتخزين سلسلة البيانات **string** والخيارات داخل تصنيف الوثيقة وعرضها من خلال تصنيف العرض، فعليك أن تختار من بين هذين التصنيفين. وتكون الأولوية للتصنيف الذى يحتوى على متغير من النوع **private** أى التصنيف **CMyApplicationDoc** فى هذه الحالة. لإضافة رسالة دالة الاحتواء، تابع معنا الخطوات الآتية:

١. قم بتنشيط التصنيف **CMyApplicationDoc** داخل التبويب **Class View**.

٢. انقر زر الأحداث من شريط أدوات مربع الخصائص.

٣. قم بتوسيع العنصر **ID_TOOLS_OPTIONS**.

٤. انقر COMMAND ثم اختر OnToolsOptions <Add> من القائمة المنسدلة المصاحبة، يتم إضافة هيكل الدالة OnToolsOptions() إلى نافذة كود التصنيف CMyApplicationDoc كما يلى:

```
void CMyApplicationDoc::OnToolsOptions()
{
        // TODO: Add your command handler code here
}
```

٥. قم بتعديل كود الدالة كما يلى:

```
1.    void CMyApplicationDoc::OnToolsOptions()
2.    {
3.      COptionsDialog dlg;
4.      dlg.m_string = string;
5.      if(dlg.DoModal() == IDOK)
6.      {
7.        string = dlg.m_string;
8.        SetModifiedFlag();
9.        UpdateAllViews(NULL);
10.     }
11.   }
```

وعن هذا الكود، نوضح ما يلى:

- فى السطر رقم ٣ يتم إنشاء حالة من التصنيف COptionsDialog الذى يمثل المربع الحوارى Options.
- فى السطر رقم ٤ يتم كتابة سلسلة البيانات الموجودة مسبقاً إلى المتغير m_string وبالتالى إلى مربع النص الموجود بالمربع الحوارى.
- فى السطر رقم ٥ يتم إظهار المربع الحوارى من خلال استدعاء الدالة DoModal() واختبار القيمة المرجعة. فإذا قام المستخدم بنقر زر Ok، يتم تنفيذ السطور التالية حيث يتم نقل محتويات مربع النص إلى المتغير string كى يتم إظهاره داخل النافذة بمجرد إغلاق المربع الحوارى.
- فى السطر رقم ٨ يتم استدعاء الدالة SetModifiedFlag() لتنشيط متغير التعديل الذى يطالب المستخدم بحفظ الوثيقة قبل الإغلاق.

- فى السطر رقم ٩ يتم استدعاء الدالة **UpdateAllViews()** مع تمرير القيمة **NULL** لإعادة تحديث العروض بالقيم الجديدة.
- كى يتم ترجمة كود الدالة، يجب أن يكون المترجم على دراية بتعريف التصنيف **COptionsDialog** وذلك بتضمين الملف **"OptionsDialog.h"** فى بداية ملف الوثيقة **MyApplicationDoc.cpp** هكذا:

```
#include "OptionsDialog.h"
```

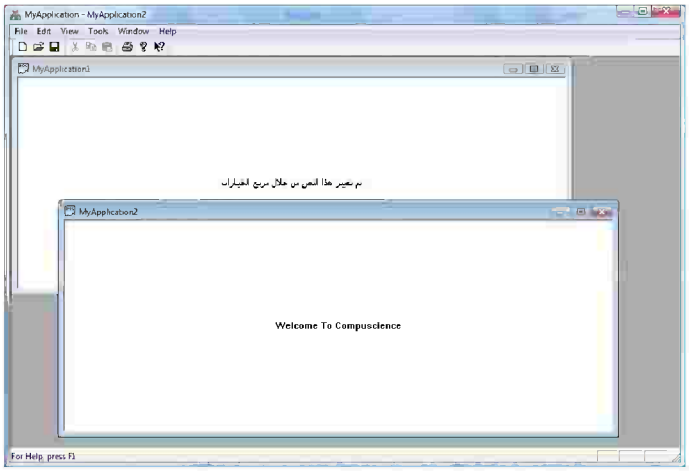
لتجربة التطبيق عند هذه النقطة، تابع معنا الخطوات الآتية:

١. قم بناء التطبيق وتنفيذه.

٢. افتح قائمة **Tools** ثم اختر **Options** من القائمة المنسدلة، يظهر المربع الحوارى **Options**.

٣. قم بتغيير النص الموجود بداخله ثم انقر زر **Ok** لإغلاق المربع الحوارى، تلاحظ تغيير النص الموجود بالنافذة ليحتوى على النص الجديد.

٤. قم بحفظ الملف ثم قم بفتح وثيقة جديدة، تلاحظ ظهور نافذتين تحتويان على نصوص مختلفة (انظر شكل ٢٣-١٣).



شكل ٢٣-١٣ تغيير النص المعروض وتخزينه داخل ملف مستقل

إضافة خيارات ظهور النص إلى المربع الحوارى

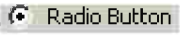
قمنا منذ قليل بإضافة المربع الحوارى **Options** إلى التطبيق **MyApplication** ثم قمنا بإضافة مربع نص وأداة عنوان إلى هذا المربع. سنقوم فيما يلى بإضافة الخيارات التى تمكن المستخدم من التحكم فى مظهر النص وذلك من خلال عدد من مربعات وأزرار الاختيار.

تعديل المربع الحوارى Options

من السهل أن تقوم بتمكين المستخدم من فتح المربع الحوارى **Fonts** ومن ثمَّ تعيين الخط المناسب بالأحجام والألوان المناسبة. ولكن نظراً لسهولة التطبيق، سنقوم باستخدام مجموعة من أزرار الاختيار لتمكين المستخدم من اختيار لون النص بالإضافة إلى مربع اختيار يحدد توسيط النص أفقياً وآخر لتوسيط النص رأسياً. لتعديل المربع الحوارى **Options**، تابع معنا الخطوات الآتية:

١. نشط تبويب عرض الموارد **Resource View** ثم انقر العنصر **IDD_OPTIONS** نقراً مزدوجاً لفتح المربع الحوارى داخل نافذة التحرير.

٢. قم بزيادة مساحة المربع الحوارى كى يتسع لأدوات التحكم الجديدة.

٣. قم بإضافة زر اختيار 

من مربع الأدوات إلى المربع الحوارى **Options**. أعد تسمية زر الاختيار إلى **IDC_OPTIONS_BLACK** مع استخدام العنوان **&Black**. قم بتخصيص القيمة **True** للخاصية **Group** وذلك لأن هذا الزر عبارة عن بداية مجموعة جديدة.

٤. قم بتكرار الخطوة السابقة لإضافة زر اختيار باسم **IDC_OPTIONS_RED** وعنوان **&Red** وآخر باسم **IDC_OPTIONS_GREEN** وعنوان **&Green**. مع التأكد من تخصيص القيمة **False** للخاصية **Group**.

٥. اختر الأزرار الثلاثة باستخدام زر الفأرة ومفتاح **Ctrl** ثم افتح قائمة **Format** واختر **Align➢Bottoms** من القائمة المنسدلة.

٦. افتح قائمة Format مرةً أخرى ثم اختـــر Across⇐Evenly Space مـــن القائمة المنسدلة (يمكنك اختيار الأزرار المصاحبة مـــن شـــريط الأدوات Dialog Editor.

٧. قم بإضافة مربع اختيار Check Box من مربع الأدوات إلى المربـــع الحـــوارى Options أسفل مجموعة أزرار الاختيار التى قمنا بإضافتها منذ قليل.

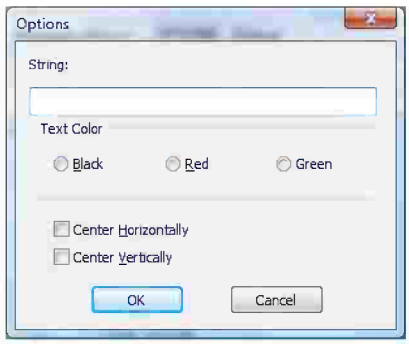
٨. قم بتغيير اســـم مربـــع الاختيـــار إلى IDC_OPTIONS_HORIZCENTER وعنوانه إلى Center &Horizontally. قم بتخصيص القيمة True للخاصـــية Group المصاحبة لمربع الاختيار وذلك لبدء مجموعة جديدة.

٩. أعـــد الخطـــوة الســـابقة لإضـــافة مربـــع اختيـــار آخـــر باســـم IDC_OPTIONS_VERTCENTER وعنوان Center &Vertically وتأكد من تخصيص القيمة False للخاصية Group.

١٠. قم بإضافة رمز مربع مجموعة Group Box من مربع الأدوات إلى المربع الحـــوارى Options حول أزرار الاختيار الثلاثة وقم بتغيير عنـــوان المجموعـــة إلى Text Color.

١١. قم بنقل زر Ok وCancel إلى أسفل المربع الحوارى.

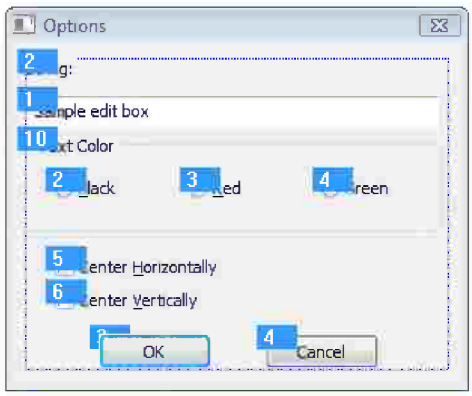
١٢. استخدم قائمة Format كما سبق لضبط محاذاة الأدوات داخل المربع الحـــوارى (انظر شكل ٢٣-١٤).



شكل ٢٣-١٤ المربع الحوارى Options بعد التعديل

لضبط ترتيب الجدولة Tab Order لأدوات المربع الحوارى، تابع معنا الخطوات الآتية:

١. افتح قائمة Format من شريط القوائم ثم اختر Tab Order من القائمة المنسدلة، يظهر المربع الحوارى محتوياً على ترتيب الوصول للعناصر من خلال مفتاح Tab (انظر شكل ٢٣-١٥).



شكل ٢٣-١٥ ترتيب الجدولة لعناصر المربع الحوارى

٢. انقر عناصر المربع الحوارى واحداً تلو الآخر بالترتيب التالى:

1. IDC_OPTIONS_STRING
2. IDC_OPTIONS_BLACK
3. IDC_OPTIONS_RED
4. IDC_OPTIONS_GREEN
5. IDC_OPTIONS_HORIZCENTER
6. IDC_OPTIONS_VERTCENTER
7. IDOK
8. IDCANCEL

٣. انقر أى مكان بعيداً عن المربع الحوارى للخروج من نطاق الترتيب.

إضافة المتغيرات إلى المربع الحوارى

كى نتمكن من استرجاع قيم مربعات الاختيار أو أزرار الاختيار، يجب ربط كل من مربعى الاختيار بمتغير وكذلك ربط زر الاختيار الأول بمتغير. لأداء ذلك، تابع معنا الخطوات الآتية:

١. نشط التبويب Class View من نافذة عمل المشروع.

٢. انقر التصنيف COptionsDialog بزر الفأرة الأيمن ثم اختر Add>Add Variable من القائمة الموضعية، 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 Value من مربع السرد Category.

٥. اختر زر الاختيار IDC_OPTIONS_black من مربع السرد Control ID.

٦. اختر نوع البيانات int بمربع السرد والتحرير Variable Type.

٧. اكتب اسم المتغير فى خانة Variable name وليكن m_color واختر درجة المتغير من مربع السرد Access وهو Public فى هذه الحالة.

٨. انقر زر Finish لإغلاق نافذة المعالج.

٩. قم بتكرار الخطوات السابقة لإضافة متغير مرتبطة بمربع الاختيار الأول باسم m_horizcenter وآخر مرتبط بمربع الاختيار الثانى باسم m_vertcenter.

إضافة كود احتواء المتغيرات الجديدة

بعد أن انتهينا من إضافة المتغيرات إلى المربع الحوارى، يجب أن نقوم بإضافة متغيرات الوثيقة وتعديل كود الدوال OnNewDocument() و Serialize() و OnDraw() لاحتواء المتغيرات الجديدة وذلك كما يلى:

١. إضافة متغيرات الوثيقة وهى نفس المتغيرات التى قمنا بإضافتها إلى المربع الحوارى. قم بإضافة الكود التالى إلى بداية تعريف التصنيف CMyApplicationDoc بملف MyApplicationDoc.h مع التعريف السابق للمتغير string والدالة GetString() كما يلى:

```
private:
        CString string;
        int color;
        BOOL horizcenter;
        BOOL vertcenter;
public:
```

```
        CString GetString(){return string;}
        int GetColor() {return color;}
        BOOL GetHorizcenter() {return horizcenter;}
        BOOL GetVertcenter() {return vertcenter;}
```

٢. قم بتعديل كود الدالة Serialize() لاحتواء المتغيرات الجديدة وذلك كما يلى:

```
void CMyApplicationDoc::Serialize(CArchive& ar)
{
        if (ar.IsStoring())
        {
                ar << string;
                ar << color;
                ar << horizcenter;
                ar << vertcenter;
        }
        else
        {
                ar >> string;
                ar >> color;
                ar >> horizcenter;
                ar >> vertcenter;
        }
}
```

٣. قم بتخصيص القيم الابتدائية للمتغيرات الجديدة داخل الدالة OnNewDocument() كما يلى:

```
BOOL CMyApplicationDoc::OnNewDocument()
{
        if (!CDocument::OnNewDocument())
                return FALSE;

        string = "Welcome To Compuscience";
        color = 0;
        horizcenter = TRUE;
        vertcenter = TRUE;
        return TRUE;
}
```

٤. قم أيضاً بتعديل دالة احتواء عنصر القائمة Options وهى الدالة OnToolsOptions() كما يلى:

```
void CMyApplicationDoc::OnToolsOptions()
{
        COptionsDialog dlg;
        dlg.m_string = string;
        dlg.m_color = color;
        dlg.m_horizcenter = horizcenter;
        dlg.m_vertcenter = vertcenter;

        if(dlg.DoModal() == IDOK)
        {
                string = dlg.m_string;
                color = dlg.m_color;
                horizcenter = dlg.m_horizcenter;
                vertcenter = dlg.m_vertcenter;
                SetModifiedFlag();
                UpdateAllViews(NULL);
        }
}
```

٥. وأخيراً قم بتعديل الدالة OnDraw() لإعادة رسم العرض وذلك كما يلى:

```
1.    void CMyApplicationView::OnDraw(CDC* pDC)
2.    {
3.      CMyApplicationDoc* pDoc = GetDocument();
4.      ASSERT_VALID(pDoc);
5.      // TODO: add draw code for native data here
6.      CRect rect;
7.      GetClientRect(&rect);
8.      COLORREF oldcolor;
9.      switch (pDoc->GetColor())
10.     {
11.       case 0:
12.                oldcolor = pDC->SetTextColor (RGB(0,0,0));
13.                 break;
14.       case 1:
15.                 oldcolor = pDC->SetTextColor (RGB(0xFF,0,0));
16.                 break;
17.        case 2:
18.                 oldcolor = pDC->SetTextColor (RGB(0,0xFF,0));
19.                 break;
20.     }
21.     int DTflags = 0;
```

```
22.     if (pDoc->GetHorizcenter())
23.     {
24.       DTflags |= DT_CENTER;
25.     }
26.     if (pDoc->GetVertcenter())
27.     {
28.       DTflags |= (DT_VCENTER | DT_SINGLELINE);
29.     }
30.     pDC->DrawText(pDoc->GetString(),&rect, DTflags);
31.     pDC->SetTextColor(oldcolor);
32.   }
```

وعن هذا الكود، نوضح ما يلى:

- فى السطر رقم ٨ تم تعريف المتغير **oldcolor** من النوع **COLORREF** ليحتوى على اللون المختار.
- تحتوى السطور من ٩ إلى ٢٠ على عبارة **switch** المستخدمة لتحديد اللون المختار بناءً على قيمة المتغير **m_color** المرتبط بزر الاختيار الأول **Black**. فإذا كانت القيمة الناتجة **0**، يكون اللون الناتج عبارة عن اللون الأحمر. وإذا كانت **1**، يكون اللون الناتج هو اللون الأحمر. أما إذا كانت **2**، فهذا يعنى أن اللون الناتج هو اللون الأخضر.
- فى السطر رقم ٢١ يتم تعريف المتغير **DTflags** مع تخصيصه بالقيمة **0** لاستخدامه فى احتواء خصائص سلسة البيانات تبعاً لمربعى الاختيار الموجودين بالمربع الحوارى.
- فى السطور من ٢٢ إلى ٢٥ يتم اختبار قيمة مربع الاختيار **Center Horizontal**، فإذا كانت صحيحة، يتم إضافة خاصية التوسيط الأفقى **DT_CENTER** إلى متغير الخصائص **DTflags**.
- بنفس الطريقة فى السطور ٢٦ إلى ٢٩ يتم اختبار قيمة مربع الاختيار **Center Vertically**.
- فى السطر رقم ٣٠ يتم استدعاء الدالة **DrawText()** لإظهار النص على النافذة.

- فى السطر رقم ٣١ تم استدعاء الدالة **SetTextColor()** لتعيين لون الخط المعروض طبقاً لنتيجة عبارة **switch** فى السطور من ٩ إلى ٢٠.

لتجربة التطبيق، تابع معنا الخطوات الآتية:

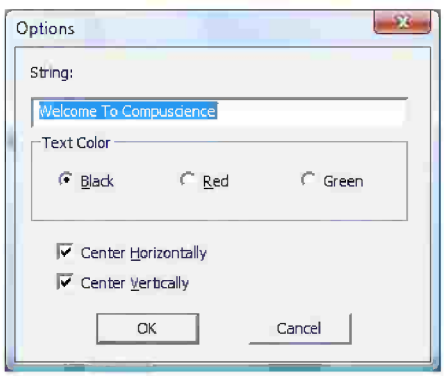
١. قم ببناء التطبيق وتنفيذه.

٢. افتح قائمة **Tools** من شريط القوائم ثم اختر **Options** من القائمة المنسدلة، يظهر المربع الحوارى **Options** محتوياً على النص الافتراضى والإعدادات الافتراضية (انظر شكل ٢٣-١٦).

٣. قم بتغيير النص الموجود ثم اختر لون آخر للنص بتنشيط أحد أزرار الاختيار الثلاثة.

٤. قم أيضاً بتغيير محاذاة النص داخل النافذة بتعطيل أحد مربعى الاختيار أو كليهما.

٥. انقر زر **Ok** لإغلاق المربع الحوارى **Options** والعودة مرةً أخرى إلى النافذة الرئيسية للتطبيق، تلاحظ تأثير إعداداتك على النص المعروض.



شكل ٢٣-١٦ المربع الحوارى محتوياً على الإعدادات الافتراضية