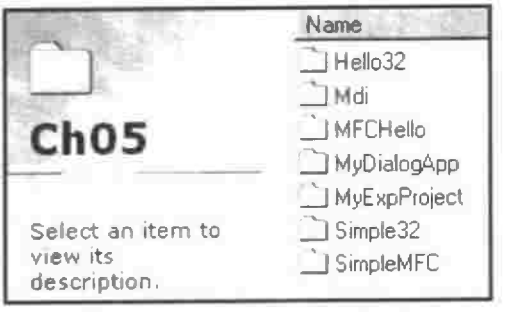
الباب الخامس

مشروعات تمهيدية

المشروعات التى وردت بهذا الباب ، كما تجدها على القرص:



مفتتح

حتى تستطيع أن تكون فكرة عن تصميم البرنامج النوافذى ، وعما يمكن أن يمنحك إياه ساحر الأستوديو ، سوف نقدم مجموعة من الأمثلة ، بعضها باستخدام دوال الوصلة البينية API والبعض باستخدام مكتبة مؤسسة الفصائل MFC.

ولا تتوقع فى هذا الباب أن تكتب الكثير من الكود ، فسوف ينصب اهتمامنا أساساً على دراسة هياكل المشروعات النوافذية بأنواعها بحيث يسهل عليك تعديلها وتطويعها لتطبيقك. وسوف نلاحظ فى تطبيقات MFC أنها دائماً تعتمد على خدمة الساحر ، أما مع برامج API فإن المعاونة التى يقدمها لك الأستوديو أقل ، ولذلك فإن اسم "الساحر" لا يوجد بطريقة صريحة ولكن العرف جرى على تسمية المعونات الخارجية باسم الساحر!

تذكر:

١. للاطلاع على الصيغة التفصيلية لإحدى الدوال أو الفصائل ، ضع مؤشر الفأر فوقها ثم اضغط زر اللوحة F1.

٢. لعرض الخريطة الكاملة لشجرة فصائل المؤسسة MFC ابحث فى شاشة النجدة (باستخدام البطاقة Search) عن "Hierarchy Chart".

٣. للاطلاع على الكود الكامل لأحد المشروعات ، افتح المشروع من القرص المصاحب للكتاب.

مثال (١) أصغر برنامج باستخدام دوال الوصلة API

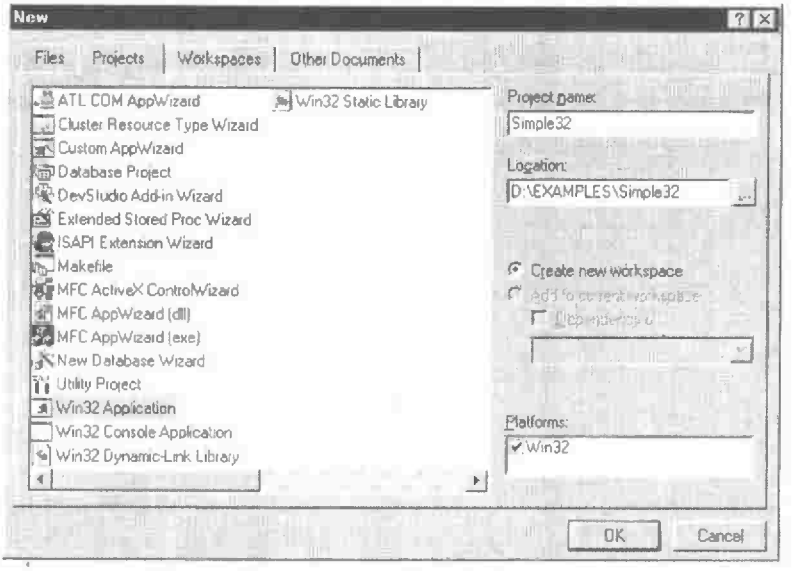
عندما تستخدم دوال الوصلة API لإنشاء مشروع من الطراز Win32 Application فإن هناك اختيارات ثلاث:

- أن تبدأ بمشروع خال تماماً ، وتكتب بنفسك كل سطور الكود.
- أن تبدأ بمشروع بسيط يحتوى على هيكل الدالة الرئيسية للبرنامج **WinMain()**.
- أن تبدأ بالمشروع المعروف "Hello World!" النوافذى ، وهو عبارة عن بناء كامل يمكنك تطويره لخلق تطبيقك الخاص.

وفى هذا المثال سوف نبدأ من المشروع الثانى الذى يحتوى على الدالة الرئيسية فقط.

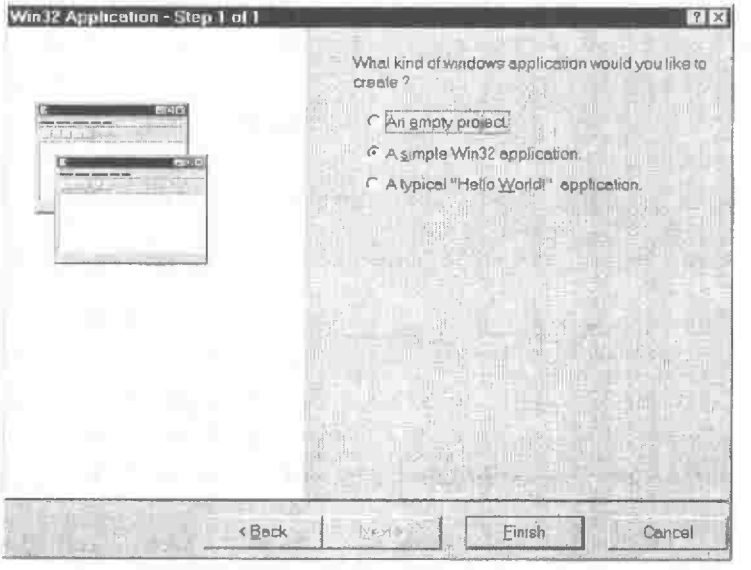
لتنفيذ هذا المثال اتبع الخطوات التالية:

١. قم بتشغيل أستوديو سى++.
٢. من قائمة الملفات (File) اختر الأمر New.
٣. اختر بطاقة المشروعات (Projects).
٤. اكتب اسم المشروع فى الصندوق الذى يحمل العنوان Project Name. وفى هذا المثال قد اخترنا الاسم Simple32.
٥. اكتب اسم الممر فى الصندوق الذى يحمل العنوان Location أو اقبل الممر سابق التعريف الموجود بالصندوق.
٦. اختر من القسم الأيسر للنافذة نوع التطبيق ليكون Win32Application. أنظر الشكل التالى.



شكل (٥-١) بدء مشروع تطبيق نوافذى

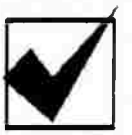
٧. بالضغط على الزر OK تنتقل إلى الشاشة التالية. اختر منـــها التطبيق "A simple Win32 application". أنظر الشكل التالى.



شكل (٥-٢) اختيار نوع المشروع

٨. اضغط على الزر Finish لإنهاء عمل الساحر ، ثـــم اضغـــط الزر OK فى النافذة التى تتلـــو والتـــى تحتـــوى علـــى بعـــض

المعلومات عن المشروع. بهذا تنتقل إلى الشاشة التالية حيث ترى محتويات الملف Simple32.cpp فى القسم الأيمن من نافذة الأستوديو.



لاحظ أن محتويات شاشتك قد تختلف قليلاً عما تراه بالشكل التالى حيث أن أقسام النافذة المختلفة يجوز تحريكها وتغيير مواضعها. لاحظ أيضاً ضرورة فتح فروع شجرة الملفات (FileView) فى القسم الأيسر حتى ترى جميع الملفات.

شكل (٥-٣) نص البرنامج البسيط

٩. قم ببناء وتنفيذ البرنامج باستخدام أمر القائمة Build - Execute Simple32.exe. لن يحدث شيء سوى ظهور نتيجة الترجمة والربط في القسم السفلي من الشاشة:

Simple32.exe - 0 error(s) - 0 warning(s)

ملفات المشروع

كما نرى في قسم الشجرة الموضح بالشكل السابق أن هناك عدة ملفات قد أنشأها البرنامج للمشروع وهي:

١. Simple32.cpp: ملف المصدر (Source File).

٢. الملفان StdAfx.cpp ، StdAfx.h: ملفات قياسية يتم خلقها مع كل مشروع. وتستخدم هذه الملفات في بناء ملف عناوين مترجم (Precompiled Header File) أي Simple32.pch.

٣. الملف Readme.txt: وهو يحتوي على معلومات عن ملفات المشروع ووظيفة كل منها.

تحليل البرنامج

```
#include "stdafx.h"
int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
        // TODO: Place code here.
        return 0;
}
```

كما نرى بالشكل السابق أن البرنامج يحتوى على دالة واحدة هى الدالة الرئيسية **()WinMain** التى تمثل قلب البرنامج النوافذى. كما نلاحظ أن نمط الدالة هو:

int APIENTRY

وهذا النمط يختص بالبرامج النوافذية فقط. ولتلاحظ أنه فى الطرازات القديمة من لغة سى++ ، كان النمط المستخدم للدالة الرئيسية هو:

int PASCAL

ولكنه قد أصبح غير مستخدم حالياً.

أما الدالة نفسها فهى فارغة من المحتويات إلاّ من العبارة return 0 علاوة على التعليق //TO DO الذى يخبرك بأنك تستطيع إضافة بعض سطور الكود فى هذا المكان.

وتستخدم الدالة الرئيسية أربعة بارامترات نوضحها بالجدول التالى:

البارامتر	النمط	الوظيفة
HInstance	HINSTANCE	مقبض التطبيق الحالى (لاحظ أن النمط HINSTANCE يكافئ النمط القديم HANDLE).
HPrevInstance	HINSTANCE	مقبض التطبيق السابق.
LpCmdLine	LPSTR	مؤشر طويل إلى خط الأوامر للتطبيق الحالى.
nCmdShow	int	لتحديد طريقة عرض نافذة التطبيق.

أما قيم هذه البارامترات فهى تمرر من برنامج النوافذ (نظام التشغيل) إلى الدالة الرئيسية للتطبيق عند بدء التشغيل. ولا نتوقع أن يكون هذا الوصف مشبعاً للبارامترات الأربعة ولكننا سوف نعود إليها فى الاستخدامات المقبلة.

إضافة الكود إلى البرنامج

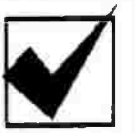
حتى تكتمل معالم البرنامج النوافذى فلابد له من إظهار نافذة ما على الشاشة ، وهذا للأسف يتطلب الكثير من الخطوات. ولكننا هنا سوف نضيف إلى البرنامج دالة واحدة تؤدى إلى ظهور صندوق رسالة (Message Box) على الشاشة وتحتوى على العبارة "Hello World" ، كما تحمل العنوان "My Message Box" ، وهذا أضعف الإيمان! أنظر هذا التعديل (الجزء الجديد مميز بالبنط الثقيل):

```
int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
  MessageBox(NULL, "Hello World", "My Message Box",0);
   return 0;
}
```

إن الدالة **MessageBox()** تستخدم أربعة بارامترات كالآتى:

- البارامتر الأول عبارة عن مقبض نافذة **HWND** وقد استخدمنا هنا القيمة **NULL** لأنه لا توجد نافذة فى هذا التطبيق.
- البارامتر الثانى هو محتوى صندوق الرسالة.
- البارامتر الثالث هو عنوان الصندوق.

- البارامتر الرابع يمثل نوع الصندوق ، وقد استخدمنا هنا القيمة 0 التى تناظر أبسط أنواع الصناديق وهو الصندوق الذى يحتوى على الزر OK فقط.



ليس مطلوب منك أن تحفظ عن ظهر قلب صيغة أى دالة من الدوال لأن بيئة الأستوديو تمدك بصيغة الدالة بمجرد أن تكتب اسمها وتفتح القوس الأيسر. أنظر الشكل التالى الذى حصلنا عليه عند كتابة اسم الدالة **MessageBox()**.

```
#include "stdafx.h"

int PASCAL WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR     lpCmdLine,
                   int       nCmdShow)
{
 MessageBox(
    return 0;
}
```

int MessageBox (**HWND hWnd**, LPCTSTR lpText, LPCTSTR lpCaption, UINT uType)

شكل (٥-٤) ظهور صيغة الدالة في بيئة الأستوديو أثناء كتابتها

قم بتنفيذ البرنامج فتحصل على صندوق الرسالة الموضح بالشكل التالى.

شكل (٥-٥) تنفيذ البرنامج

مثال (٢) برنامج "Hello World" باستخدام دوال API

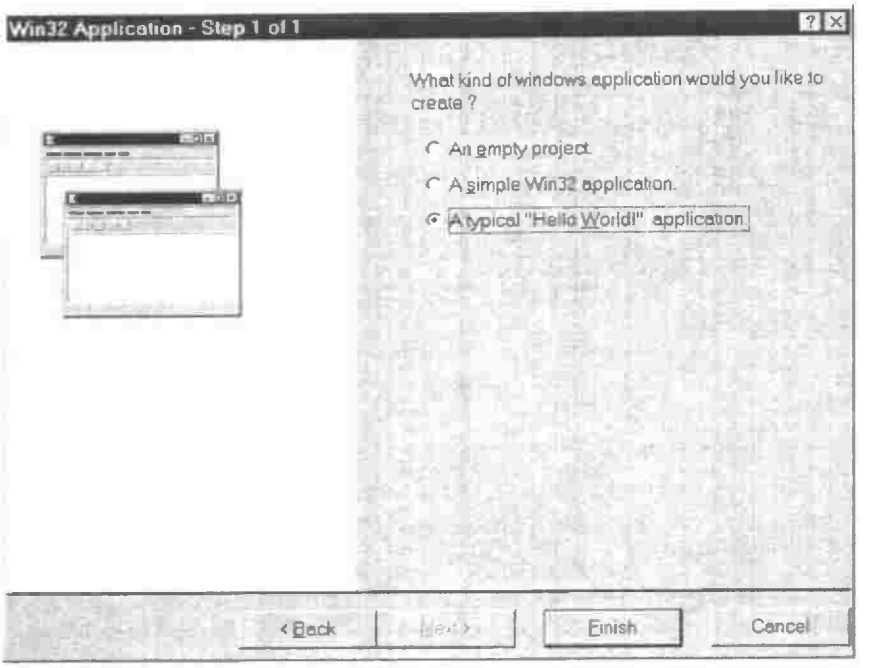
من المعروف أن البرنامج "Hello World!" هو أصغر برنامج يمكن كتابته بلغة سى ، وقد ارتبطت هذه اللغة بهذا البرنامج منذ ولادتها. ترى كيف يكون الحال عندما نكتب العبارة "Hello World!" بداخل نافذة؟ إن هذا هو موضوع التطبيق الحالى.

لتنفيذ هذا المشروع اتبع الخطوات الآتية:

١. ابدأ بإغلاق حيز العمل للمشروع السابق باستخدام أمر القائمة:
File - Close Workspace

٢. سوف تظهر رسالة تسألك إذا كنت ترغب فى إغلاق جميع النوافذ "Do you want to close all document windows?" ، أجب بنعم (Yes) فتختفى ملفات المشروع.

٣. كرر الخطوات السابقة لبدء مشروع بالاسم "Hello32" ، وعندما تصل إلى الشاشة الموضحة بالشكل التالى اختر منها الزر الثالث الذى يحمل الاسم: A typical "Hello World!" application.



شكل (٥-٦) اختيار مشروع البرنامج "Hello World"

٤. اضغط الزر Finish لإنهاء عمل الساحر ، ثم الزر OK بالشاشة التالية. فتظهر نافذة المشروع الموضحة بالشكل التالى.

ملاحظة:

افتح الدوسيهات المختلفة فى بطاقة مشهد الملفات (FileView) بقسم الشجرة حتى تحصل على نفس المشهد.

ملاحظة:

محتوية على قائمة تشمل اختيارات الملفات (Files) والنجدة (Help). ولو أنك فتحت قائمة الملفات لن تجد فيها سوى أمر الخروج من البرنامج (Exit) ، أما قائمة النجدة فهى تحتوى على الأمر About الذى يعرض بعض المعلومات عن التطبيق. كما نلاحظ أن النافذة تحتوى على أزرار التحكم القياسية للنوافذ (التكبير والتصغير والإغلاق) كما تحتوى على سطر للعنوان يحمل اسم المشروع Hello32.

ملفات المشروع

يحتوى هذا المشروع على الملفات الأساسية الآتية:

١. مجموعة ملفات الكود (Source Files) وهى تشمل الملف Hello32.cpp وملف الموارد Hello32.rc (سيلى شرح ملفات الموارد فى التطبيقات القادمة).

٢. مجموعة ملفات العناوين (Header Files) وتشمل الملف Hello32.h وملف عناوين الموارد resource.h.

٣. مجموعة ملفات الموارد (Resource Files) وهى تشمل الأيقونات المرتبطة بالبرنامج (تظهر الأيقونة فى الركن الأيسر العلوى من النافذة ، أو عند تصغير نافذة التطبيق).

لن نعرض هنا نصوص البرامج الكاملة حيث أنها محتوية على أكواد جاهزة وتستطيع مشاهدتها على شاشة الأستوديو مباشرة. ومع ذلك فعندما نضيف أى سطور جديدة إلى هذه البرامج سوف نشرحها بالتفصيل. وإذا وجدت صعوبة فى خلق هذه المشروعات فيمكنك تحميلها من القرص المصاحب للكتاب مباشرة.

تصميم البرنامج

بفحص الملف Hello32.cpp تستطيع أن تكون فكرة عن منطق البرنامج فهو يحتوى على خمسة دوال نوضحها فى الجدول التالى:

اسم الدالة	الوظيفة
الدالة الرئيسية **WinMain()**	لشحن التطبيق ، وتشغيل الحلقة التكرارية للرسائل.
دالة تسجيل نافذة التطبيق MyRegisterClass()	لتسجيل خصائص نافذة التطبيق فى بيئة النوافذ.
دالة شحن هدف التطبيق **InitInstance()**	لخلق وعرض نافذة التطبيق وحفظ مقبضها كمتغير عام.
روتين النوافذ **WinProc()**	للاستجابة لرسائل النوافذ.
دالة المعلومات **About()**.	عرض معلومة عن التطبيق الحالى.

إن جميع هذه الدوال تتكرر فى أى برنامج نوافذى (لاحظ أن البرنامج فى المثال السابق غير مكتمل المعالم لأنه بلا نافذة). أين

إذن الجزء المسئول عن كتابة العبارة "Hello World!" فى النـافذة؟ إن هذه العبارة هى جزء من عملية طلاء النافذة بالمحتويات ، ولو أنك فحصت الدالة **WinProc()** المخصصة للاستجابة للرسائل ، فسوف تجد بها السويتش "case WM_PAINT:" المخصص لطـلاء النـافذة. وبداخل هذا السويتش تجد الدالة **DrawText()** (وهى الدالة المسئولة عن كتابة النص) بالصورة الآتية:

```
DrawText(hdc, szHello, strlen(szHello), &rt, DT_CENTER);
```

إن العبارة "Hello World!" مخزنة فى المتغير szHello ، وفى إمكـانك أن تستبدل هذا المتغير بالعبارة "Hello World!" مباشرة فتحصل على نفس النتيجة. ومع ذلك فليست هذه هى الطريقة السليمة للتعامل مـع برامج النوافذ.

إن الثابت الحرفى "Hello World!" قد تم الإعلان عنه مع غيره مـن الثوابت الحرفية التى يتضمنها البرنامج (مثل عنوان النافذة) فيمـا يسمى بجدول الحرفيـات (String Table) ، بـالملف resource.h ، كالأتى:

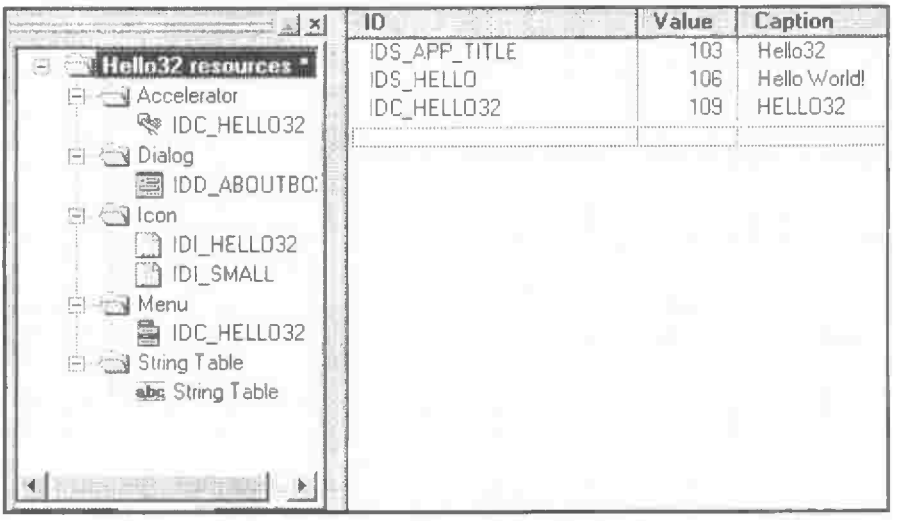
ID	Value	Caption
IDS_HELLO	106	Hello World!
...	...	...

وفى هذا الجدول قد تم منح العبارة "Hello World!" ثابتـاً للتعـارف بالاسم IDS_HELLO والقيمة العددية 106 . وقد تم تحميل هذا الثـابت الحرفى وربطه بالمتغير szHello باستخدام الدالة **LoadString()** فـى بداية الدالة **WinProc()** كالآتى:

```
LoadString(hInst, IDS_HELLO, szHello, MAX_LOADSTRING);
```

ولو أنك استبدلت الثابت IDS_HELLO بالعدد 106 لحصلت على نفس النتيجة.

واستخدام جداول الحرفيات يجعل البرنامج متماسكاً ومركزياً فـــى تصميمه وإدارته. ولو أنك أردت طباعة عبارة أخرى على الشاشـــة فيمكنك أن تغير محتوى العبارة فى جدول الحرفيات. والشكل التالى يوضح جدول الحرفيات كجزء من ملفات الموارد التى ســـنتعرض لها بالتفصيل فى حينه.



شكل (٥-٩) جدول الحرفيات على اليمين وملفات الموارد على اليسار

فى هذا المثال تعرفنا بهيكل البرنامج النوافـــذى وعرفنـــا الـــدوال الأساسية التى يحتوى عليها كل برنامج نوافذى مهما كان صغـيراً ، كما عرفنا قدر الخدمة الجليلة التى يقدمها لنا الساحر حيث أنه يكتب لنا هيكل البرنامج وكل ما يتكرر به من دوال. ومـــن البديـــهى أن

البرنامج "Hello World" يمكن تطويره فى اتجـــاه أى تطبيــق مــن التطبيقات.

مثال (٣) أصغر برنامج باستخدام مؤسسة الفصائل MFC

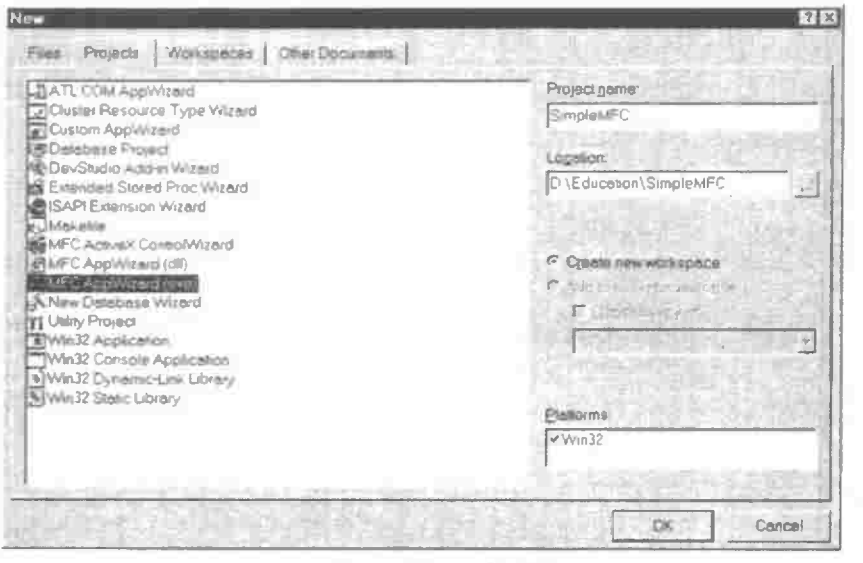
فى هذه الفقرة ننشئ مشروعاً بسيطاً باستخدام مؤسســـة الفصـــائل MFC. وهذه النوعية من المشروعات مصحوبة دائماً بخدمة الســاحر حيث أن جميع المشروعات تحتوى على مجموعــة معروفــة مــن الفصائل الموروثة من فصائل المؤسسة MFC.

لتنفيذ هذا المشروع اتبع الآتى:

١. أغلق المشروع السابق ثم ابدأ مشروعا جديداً بالأمر:

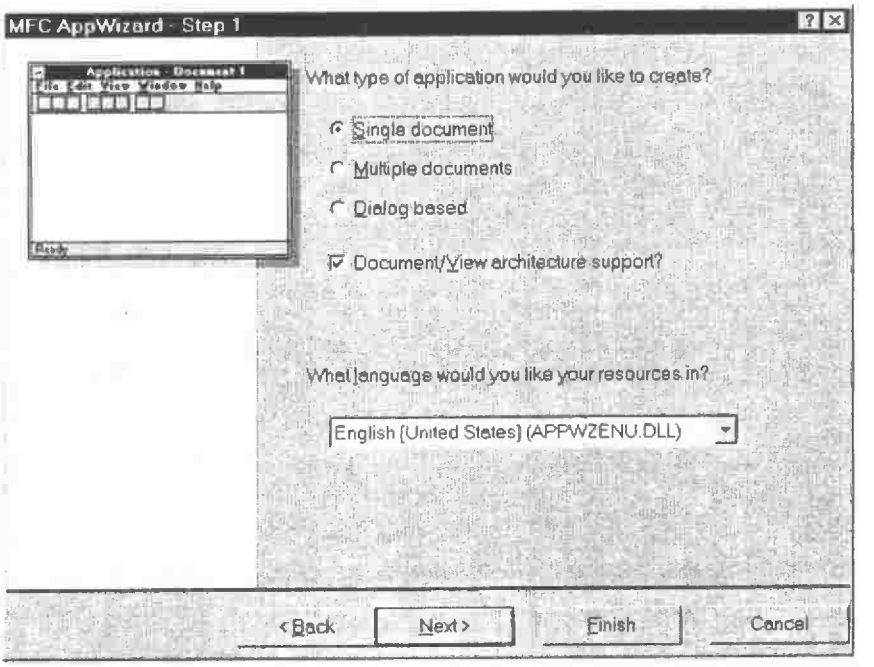
File – New

٢. أطلق الاسم SimpleMFC على المشروع ، واختر نوع المشروع ليكون MFC AppWizard(exe). انظر الشكل التالى.



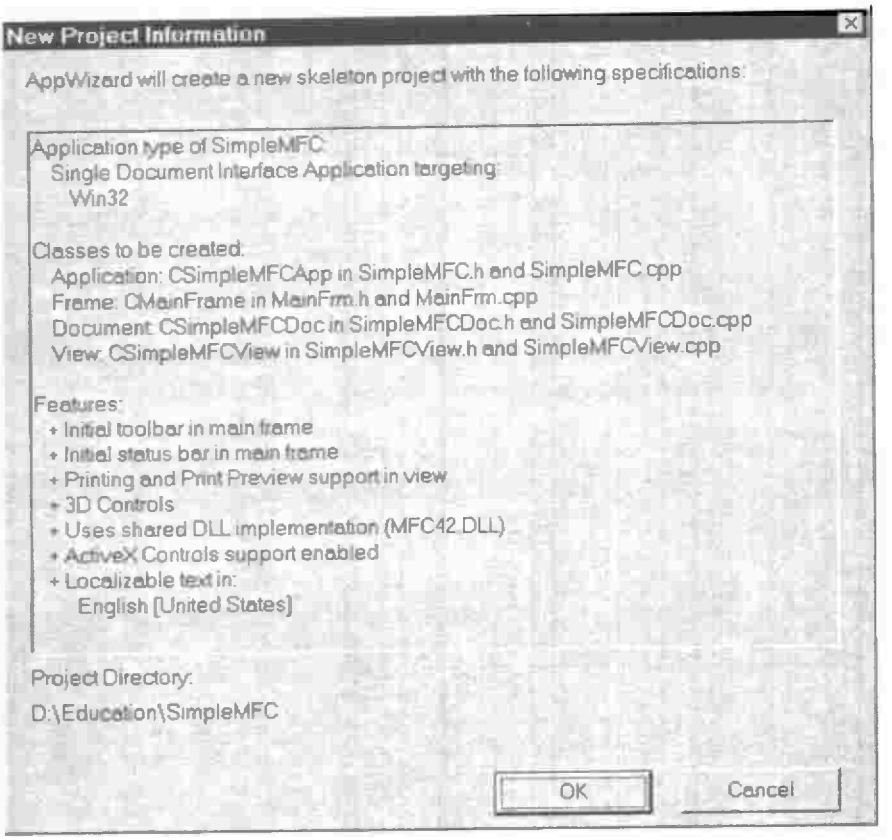
شكل (٥-١٠) بدء مشروع MFC

٣.بالضغط على الزر OK تنتقل إلى الشاشة التالية. اختر منها الوثيقة المفردة (Single Document) كما هو موضح بالشكل التالى.



شكل (٥-١١) تحديد اختيارات المشروع - الوثيقة المفردة

٤. اضغط على الزر Finish لإنهاء عمل الساحر عند هذا الحد وخلق المشروع الجديد. بهذا تحصل على الشاشة التالية التى تحتوى على تقرير بمكونات المشروع.



شكل (٥-١٢) الشاشة النهائية فى مشروع MFC

تنفيذ البرنامج

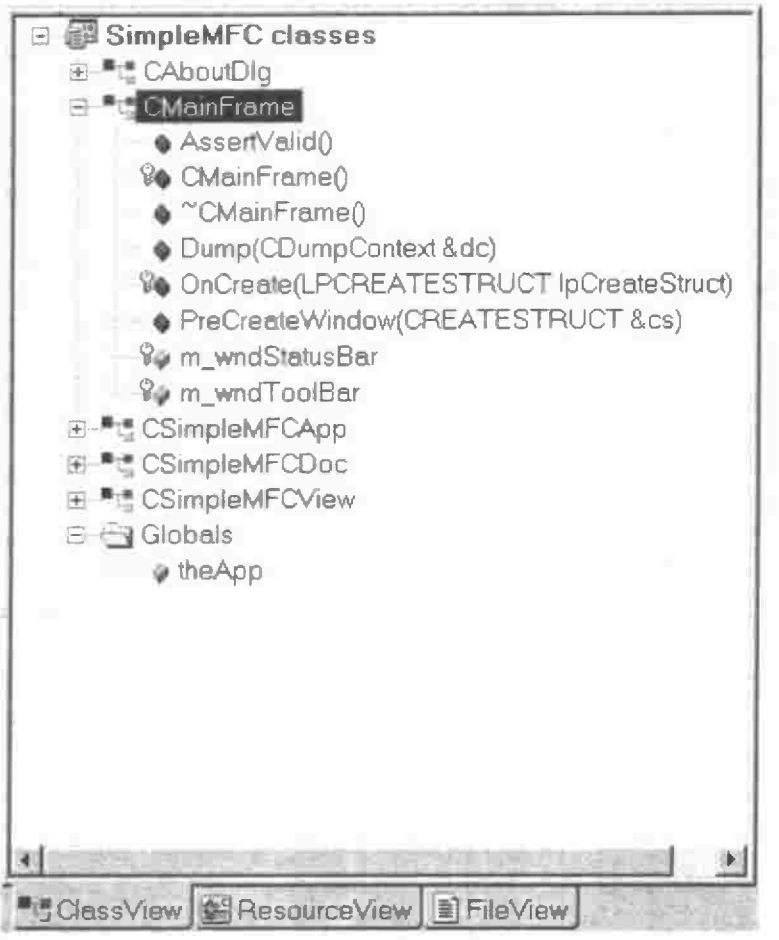
عند تنفيذ هذا البرنامج فإنك تحصل على النافذة الموضحة بالشكل التالى. ومن أول وهلة فإنك تستطيع أن تعرف أنها أثرى من نافذة البرنامج "Hello World" التى أنشأناها باستخدام دوال الوصلة API فى المثال السابق. فالقائمة هنا تحتوى على أربعة اختيارات ، ولو أنك فتحت قائمة الملفات (File) فسوف تجد بها كل الأوامر التى قد تحتاج إليها فى تطبيقاتك مثل New, Save, Save As, Print, ...

شكل (٥-١٣) تنفيذ البرنامج

كما أن النافذة تحتوى على سطر للأدوات يمكنك استخدامه كبديل سريع لأوامر القائمة.

فصائل البرنامج

لدراسة تصميم البرنامج المبنى على مكتبة مؤسسة الفصائل MFC ، فمن الأفضل أن نبدأ من شجرة الفصائل. اضغط على بطاقة مشهد الفصائل (ClassView) بقسم الشجرة فتشاهد الشجرة الموضحة بالشكل التالى. (لاحظ أن أفرع الشجرة كلها مغلقة ما عدا فرع الفصيلة CMainFrame).



شكل (٥-١٤) فصائل المشروع

لقد أنشأ لنا الساحر فى هذا البرنامج أربعة فصائل موروثة من فصائل المؤسسة MFC. (هناك فصيلة خامسة CAboutDlg سوف تراها فى كل البرامج ولكنها لا تهمنا فى عمليات المعالجة).

وتشتق أسماء الفصائل من اسم التطبيق نفسه ، وتبدأ أسماؤها بالحرف C كما هو العرف فى تسمية الفصائل ، وهذه هى الفصائل التى يحتوى عليها التطبيق "SimpleMFC":

١. **فصيلة الإطار العام** CMainFrame: وهى تحمل هذا الاسم دائماً بصرف النظر عن اسم التطبيق.

٢. **فصيلة التطبيق** CSimpleMFCApp: ويتكون الاسم من الحرف C فى المقدمة ، يليه اسم التطبيق SimpleMFC ، يليه الحروف App (اختصار كلمة Applicatipon).

٣. **فصيلة الوثيقة** CSimpleMFCDoc: ويتكون الاسم من الحرف C ، ثم اسم التطبيق متبوعاً بالحروف Doc (اختصار كلمة Document).

٤. **فصيلة المشهد** CSimpleMFCView: ويتكون الاسم من الحرف C ، ثم اسم التطبيق ، ثم الكلمة View.

وفيما يلى ملخص بوظائف هذه الفصائل ، علاوة على أسماء فصائل المؤسسة MFC المشتقة منها:

فصيلة المشروع "SimpleMFC"	موروثة من الفصيلة	الوظيفة
الإطار العام CmainFrame	CFrameWnd	تستخدم فى إدارة النافذة الرئيسية للتطبيق. وتعتبر نافذة المشهد (View) نافذة ابنة للنافذة الرئيسية.
التطبيق CSimpleMFCApp	CWinApp	إدارة التطبيق ككل.
الوثيقة CSimpleMFCDoc	CDocument	تخزين بيانات البرنامج ، وقراءة البيانات من القرص أو كتابتها عليه.

المشـــــــــــــــهد CSimpleMFCView	CView	عـــرض المعلومـــات علـــى الشاشة واستقبال البيانات مـــن المســـتخدم ، وإدارة نـــــافذة المشهد.

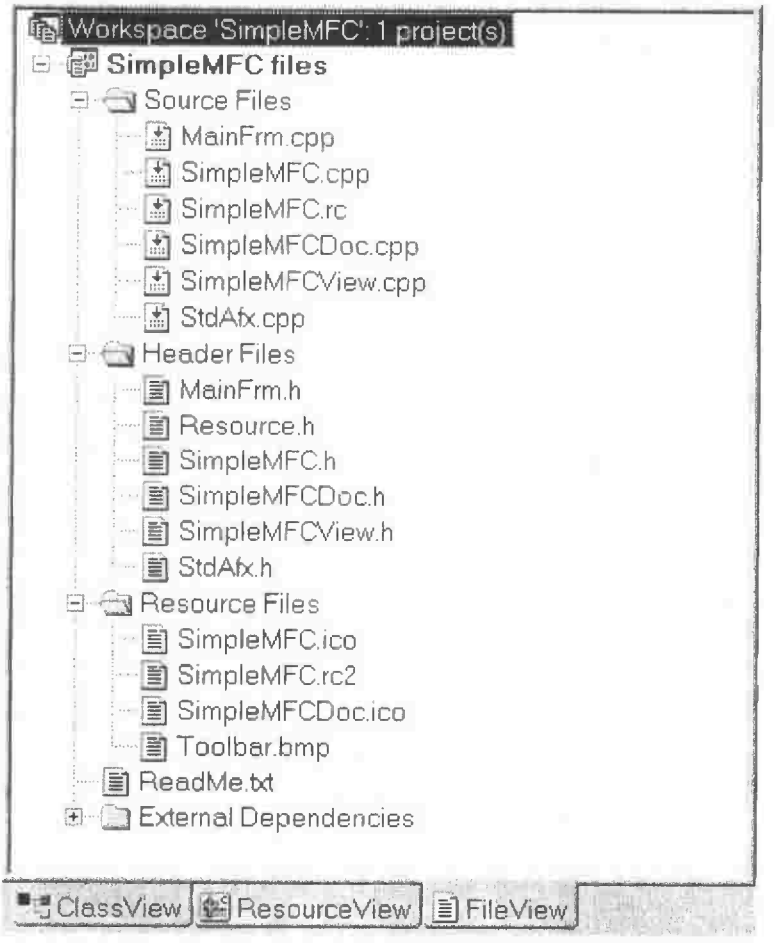
ملفات البرنامج

افتح فروع الشجرة بمشهد الملفات (FileView) كما بالشكل التـــالى ، فتشاهد بها الأقسام الرئيسية الآتية:

١.ملفات المصدر (Source Files)

٢.ملفات العناوين (Header Files)

٣.ملفات الموارد (Resource Files)

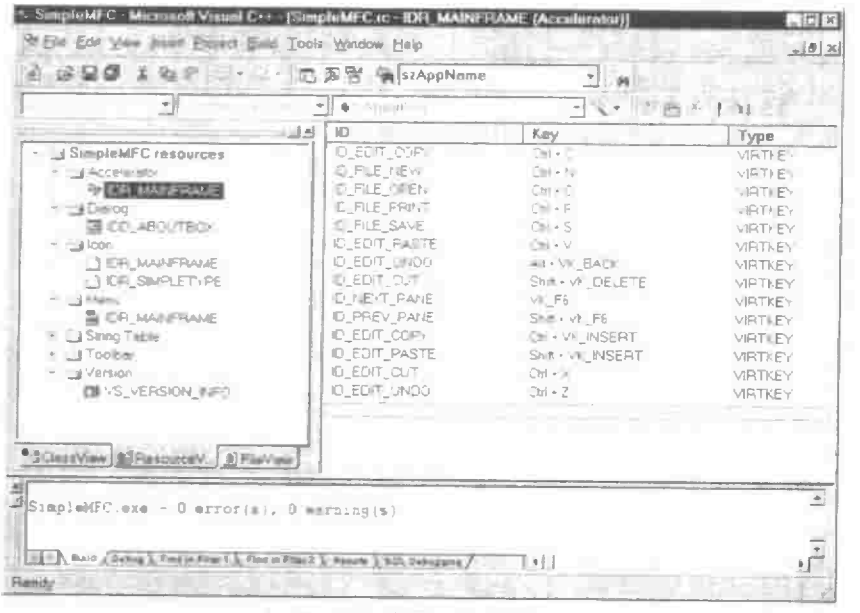


شكل (٥-١٥) ملفات المشروع

ونلاحظ أن لكل فصيلة ملف للإعلان (ملف عناوين .h) وآخر للتطبيق (ملف كود .cpp). وهذا هو بيان الملفات وعلاقتها بالفصائل:

الفصيلة	ملف العناوين (.h)	ملف التطبيق (.cpp)
CmainFram	MainFrm.h	MainFrm.cpp
CsimpleMFCDoc	SimpleMFCDoc.h	SimpleMFCDoc.cpp
CsimpleMFCView	SimpleMFCView.h	SimpleMFCView.cpp
CsimpleMFCApp	SimpleMFC.h	SimpleMFC.cpp

بخلاف ملفات التطبيق والإعلانات فإن هناك العديد من ملفات الموارد التى تستخدم فى تصميم القوائم والأيقونات وسطور الأدوات وجداول الحرفيات. ويوضح الشكل التالى جانباً من شجرة ملفات الموارد حيث نرى ثوابت التعارف التى تبدأ بالحرفين ID والتى تستخدم فى أزرار التعجيل (Accelerators) وهى الأزرار المستخدمة كبدائل لاختيارات القائمة. فعلى سبيل المثال تُستخدم مجموعة الأزرار Ctrl+C للنسخ ، وهى تناظر أمر قائمة التحرير Edit - Copy ، لذلك فإنها تأخذ رقم التعارف ID_EDIT_COPY. وكما نرى أن الساحر قد أعد جميع بيانات هذه الأزرار بملف الموارد ، ولن نحتاج إلى التعامل مع هذا الملف إلاّ إذا أردنا إضافة وظيفة جديدة غير موجودة بالقائمة.



شكل (٥-١٦) ملفات الموارد

وبصفة عامة فالملفات التى يولدها الساحر يمكن تمييزها بامتداد اسم الملف كالآتى (لاحظ أن بعض هذه الملفات لا يظهر فى الشـــــجرة ولكنك تجدها فى دوسيه المشروع):

امتداد الملف	الوظيفة
.cpp	ملف كود (سى++).
.h	ملف عناوين.
.clw	ملف معلومات الساحر.
.txt	ملف نصوص.
.rc	ملف موارد. يحتوى على الموارد التى يولدها الساحر.
.rc2	ملف موارد لتعريف الموارد بطريقة يدوية. لا يستخدمه الساحر.
.res	ملف موارد مُترجم.
.dsp	ملف معلومات المشروع.
.dsw	ملف حيز العمل للمشروع. بضغطة مزدوجة عليه يتم فتح المشروع فى بيئة الأستوديو.
.ico	ملف أيقونات.
.bmp	ملف رسم.
.exe	الملف التنفيذى (يوجد بالدوسيه Debug لكل مشروع). بضغطة مزدوجة عليه يتم تنفيذ البرنامج.

ولن نتعرض فى هذا الباب إلى تفصيلات الدوال الأعضاء التى تحتوى عليها الفصائل ، ولكننا سوف نتعرف بها بالتدريج من خلال الاستخدام.

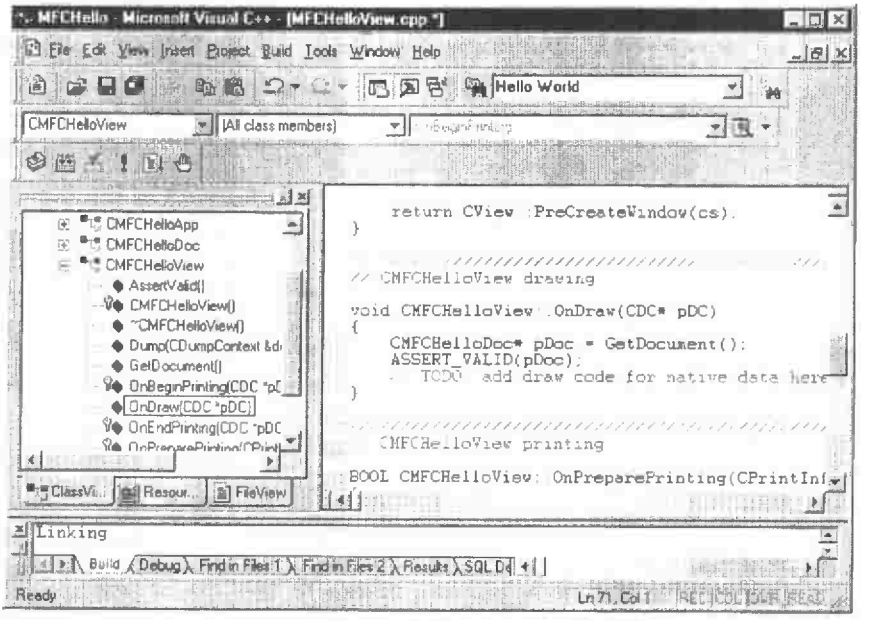
مثال (٤) إضافة الكود إلى برنامج MFC

فى هذه الفقرة سوف نرى كيف يمكننا إضافة بعض العبارات إلى البرنامج لكى نطبع على الشاشة عبارة مثل "Hello My MFC Friends". ولكى تنفذ هذا المشروع ابدأ بخلق برنامج MFC (على غرار البرنامج السابق) بالاسم MFCHello.

إن الفصيلة المسئولة عن الكتابة فى النافذة هى فصيلة المشهد (View). ولذلك فإننا سوف نقوم بإضافة الكود إلى دوال هذه الفصيلة.

لتحقيق ذلك اتبع الآتى:

١. افتح شجرة الفصائل ، ثم اضغط ضغطة مزدوجة على اسم الفصيلة CMFCHelloView ، فتحصل على مشهد مماثل لما بالشكل التالى.



شكل (٥-١٧) فصيلة المشهد للبرنامج MFCHello

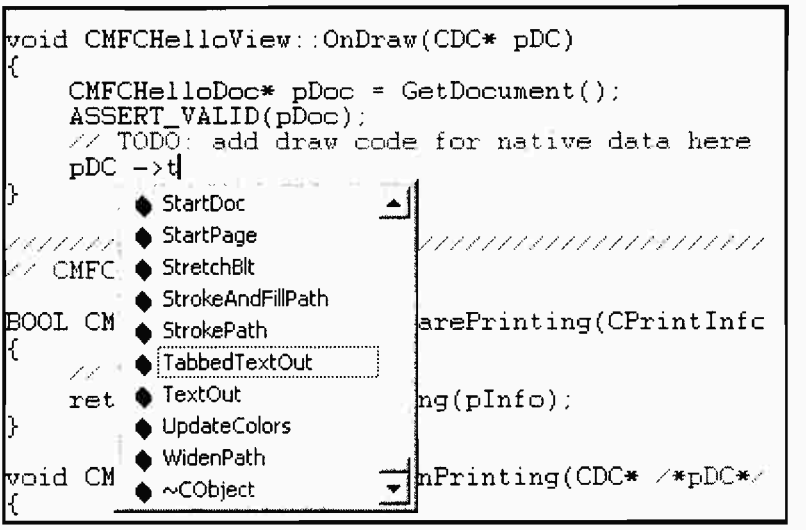
٢.حرك قضبان الانزلاق فى الجزء الأيمن من النافذة حتى تصل إلى الدالة **OnDraw()** ، ونصها كالآتى:

```
//////////////////////////////////////////////////////////////
// CMFCHelloView drawing

void CMFCHelloView::OnDraw(CDC* pDC)
{
        CMFCHelloDoc* pDoc = GetDocument();
        ASSERT_VALID(pDoc);
        // TODO: add draw code for native data here
      أضف هنا الكود المطلوب
}
```

إن هذه الدالة ، كما نرى فى شريحة البرنامج السابق ، تستخدم البارامتر pDC وهو عبارة عن مؤشر إلى فصيلة منطقة العرض CDC. وباستخدام هذا المؤشر يمكنك التوصل إلى الدوال الأعضاء

اسم الدالة مباشرة من القائمة بضغطة مزدوجة عليـــها. انظـــر الشكل التالى.



شكل (٥-١٩) قائمة المساعدة تعرض أسماء الأعضاء التي يشير إليها مؤشر الفصيلة

٢. كذلك عندما تفتح قوس الدالة فإن صيغة الدالة تظـــهر بداخـــل النافذة المساعدة ، كما ذكرنا مـــن قبـــل. وفـــى حالـــة الدالـــة **TextOut()** ، وهى دالة ذات صيغتيـــــن ، تظـــهر الصيغتـــان بالصورة الموضحة بالشكل التالى. وبالضغط علـــــى الرقـــم ١ تظهر الصيغة الأولى ، وبالضغط على الرقم ٢ تظهر الصيغـــة الثانية.

```
// CMFCHelloView drawing

void CMFCHelloView::OnDraw(CDC* pDC)
{
    CMFCHelloDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    pDC ->TextOut(
}
```

1 of 2 ▸ BOOL TextOut (int x, int y, const CString &str)

شكل (٥-٢٠) نافذة المساعدة تعرض صيغتى الدالة TextOut()

٣. إذا أردت الاطلاع على جميع محتويات إحدى الفصائل من الدوال الأعضاء ، ابحث فى شاشة النجدة (الاختيار Search) عن اسم الفصيلة (مثل **CDC** أو **CDC** Class Members).

استخدام فصيلة الوثيقة (Document Class)

من الجائز تنفيذ عملية الطباعة على الشاشة بطريقة أفضل حيث نضيف الحرفى المراد طباعته كعضو من أعضاء فصيلة الوثيقة ، ونشحنه بدالة بناء الفصيلة. ثم نطبعه على الشاشة من خلال فصيلة المشهد. وهذه هى الخطوات:

١. اضغط على اسم الفصيلة CMFHelloDoc فى قسم شجرة الفصائل. فيظهر فى القسم الأيمن محتويات الملف MFCHelloDoc.h المخصص لإعلانات الفصيلة. أضف العضو الآتى إلى أعضاء الفصيلة (العضو الجديد مميز بالبنط الثقيل):

```
class CMFCHelloDoc : public CDocument
{
protected: // create from serialization only
        CMFCHelloDoc();
```

```
        DECLARE_DYNCREATE(CMFCHelloDoc)
// Attributes
public:
        CString m_strHelloText; ----> العضو الجديد
```

لقد تم إعلان هذا العضو باستخدام النمط **CString** وهو اسم فصيلة الحرفيات بالمؤسسة MFC ، أى أنه هدف حرفى. ويمكنك معاملة هذا النمط كمتغير حرفى ، وهو يتميز عن متغيرات لغة سى بسهولة الاستعمال فهو يماثل متغيرات لغة بيسك التقليدية ، ويمكنك التخصيص له مباشرة. كما نلاحظ استخدام البادئة m_ فى اسم المتغير وهذا مجرد عرف لتمييز البيانات الأعضاء.

٢. اشحن العضو بالبيان الحرفى باستخدام دالة بناء الفصيلة CMFCHelloDoc::CMFCHelloDoc() كالآتى:

```
// CMFCHelloDoc construction/destruction
CMFCHelloDoc::CMFCHelloDoc()
{
        // TODO: add one-time construction code here
                                                شحن الهدف الحرفى:
        m_strHelloText = "Hello my MFC friends!";
}
```

٣. افتح فصيلة المشهد مرة أخرى ، وعدّل صيغة الدالة **TextOut()** إلى الآتى:

```
void CMFCHelloView::OnDraw(CDC* pDC)
{
        CMFCHelloDoc* pDoc = GetDocument();
        ASSERT_VALID(pDoc);
        // TODO: add draw code for native data here
        pDC ->TextOut(100,100, pDoc->m_strHelloText);
}
```

لاحظ هنا أننا قد استدعينا عضو فصيلة الوثيقة (m_strHelloText) باستخدام مؤشر إلى هذه الفصيلة pDoc وهو موجود أصلاً بداخل الدالة. كما نلاحظ أن هذه هى الصيغة الثانية للدالة **TextOut()** ، فالأولى تستخدم حرفى ، والثانية تستخدم مؤشراً.

٤. جرب تشغيل البرنامج مرة أخرى فتحصل على نفس النتيجة.

الموجز

١. تعرفنا فى هذا الباب بأنواع المشروعات التى يمكن أن يبنيه ساحر الأستوديو سواء المشروعات المبنية على دوال الوصلة البينية API ، أو على مؤسسة الفصائل MFC.

٢. كما تعرفنا بالدالة الأساسية فى البرنامج النوافذى WinMain() وعرفنا وظائف بارامتراتها الأربعة:

hInstance
hPrevInstance
lpCmdLine
nCmdShow

٣. عرفنا دالة صندوق الرسالة **MessageBox()** التى يمكن استخدامها فى طباعة نص سريع بصندوق رسالة من داخل الدالة الرئيسية.

٤. تعرفنا أيضاً بالبرنامج "HelloWorld!" المبنى باستخدام دوال الوصلة البينية API ، وتعرفنا فيه بالدوال الأساسية التى يبنى منها البرنامج وهى:

WinMain()
MyRegisterClass()

InitInstance()
WinProc()
About()

٥. عرفنا أيضاً كيفية طباعة رسالة على الشاشة باستخدام البرنامج النوافذى المبنى على دوال API وذلك باستخدام جدول الحرفيات (String Table). وقد استخدمنا لذلك الدوال:

- لكتابة الحرفى فى النافذة **DrawText()**
- لتحميل الحرفى من جدول الحرفيات **LoadString()**

٦. مررنا بملفات الموارد مروراً سريعاً وعرفنا أن الساحر ينشئ ملفات الموارد بما فيها الأيقونات واختيارات القائمة ، كما عرفنا أزرار التعجيل (Accelerators) المرتبطة باختيارات القائمة ، وعرفنا أن لكل منها رقم تعارف يبدأ بالحرفين ID.

٧. تعرفنا بتصميم البرنامج النوافذى المبنى على مؤسسة الفصائل MFC وعرفنا الفصائل التى يبنى منها المشروع:

- فصيلة الإطار العام MainFrame
- فصيلة التطبيق CxxxApp
- فصيلة المشهد CxxxView
- فصيلة الوثيقة CxxxDoc

حيث: xxx هو اسم المشروع.

٨. عرفنا أيضاً أن كل فصيلة من فصائل المشروع ترث إحدى فصائل المؤسسة MFC ، وتعرفنا بملفات إعلان الفصائل (.h) وملفات تطبيق الفصائل (.cpp) تعارفاً عاماً.

٩. عرفنا كيفية إضافة سطور الكود إلى مشروع MFC لطباعة عبارة ما فى النافذة باستخدام الدالة **TextOut()** كما عرفنا كيفية

استخدام المؤشرات إلى الفصائل المختلفة للتوصل إلى أعضائها. كما عرفنا هدف الحرفيات **CString** المستخدم كنمط للحرفيات فى برنامج MFC ، وقد رأينا أنه أسهل فى الاستخدام من متغيرات لغة سى التقليدية.

١٠. تعرفنا فى هذا الباب بفصائل مؤسسة MFC الآتية:

- **CFrameWnd** الإطار العام
- **CWinApp** التطبيق
- **CDocument** الوثيقة
- **CView** المشهد
- **CDC** منطقة العرض

١١. الدوال التى وردت فى هذا الباب:

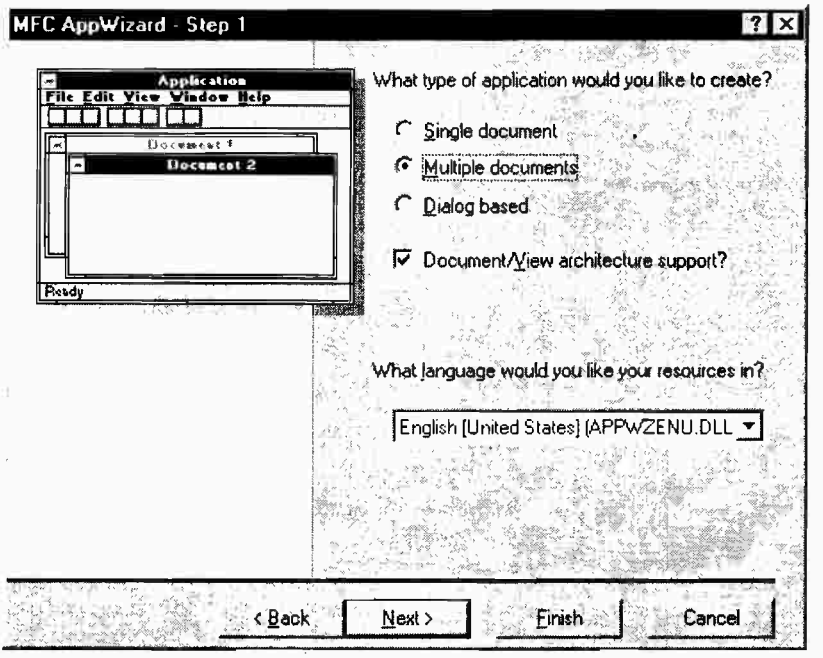
About()
CDC::TextOut()
DrawText()
InitInstance()
LoadString()
MessageBox()
MyRegisterClass()
WinProc()
WniMain()

١٢. الأنماط التى وردت فى هذا الباب:

APIENTRY
PASCAL
HINSTANCE
LPSTR
CString

حتى تستكشف كل ما يمكن أن يقدمه الأستوديو من تطبيقات أنشئ المشروعات الآتية ، ونفذ برامجها ، وادرس محتوياتها:

١. جرب إنشاء برنامج مماثل للمثال الرابع مع استخدام خاصية الوثائق المتعددة (Multiple documents) بالخطوة رقم ١ (Step 1) من خطوات الساحر. أنظر الشكل التالى. بعد إنشاء التطبيق جرب استخدام أمر القائمة File - New عدة مرات وشاهد النتائج.



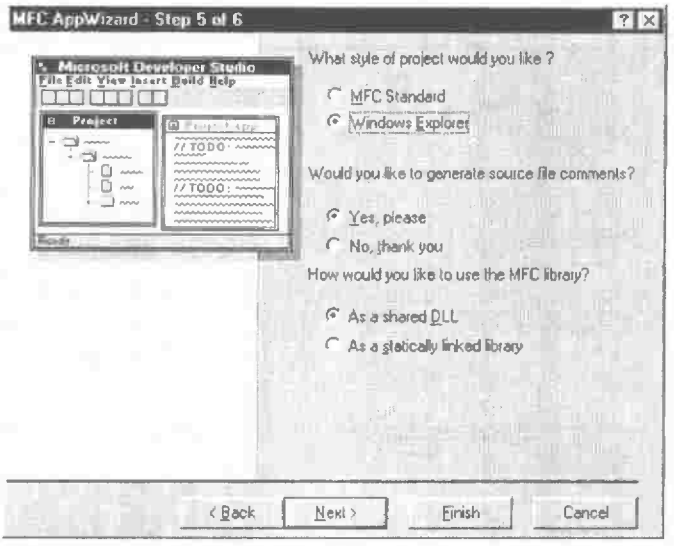
شكل (٥-٢١) تطبيق متعدد الوثائق (Multiple Documents)

ملاحظة:

يوجد هذا المشروع على القرص المصاحب بالاسم Mdi. ويمكنك فتحه من بيئة الأستوديو أو بضغطة مزدوجة على اسم الملف Mdi.dsw. كما يمكنك تنفيذه مباشرة بدون فتح الأستوديو بضغطة مزدوجة على الملف التنفيذى Mdi.exe بالدوسيه \ExamplesVCPP\Ch05\Mdi\Debug. وهذا ينطبق على جميع مشروعات القرص.

ملاحظة:

٢. من أنواع التطبيقات التى يمكن أن ينشئها لك الساحر ، التطبيق الذى يأخذ صورة الكشاف (Explorer) ، بحيث تحتوى نافذة التطبيق على قسم أيمن وقسم أيسر. لخلق هذا لتطبيق استمر فى الضغط على الزر Next بنافذة الساحر حتى تصل إلى الخطوة رقم ٥ (Step 5 of 6). من هذه الشاشة اختر زر الراديو Windows Explorer ثم اضغط الزر Finish عند هذا الحد. أنظر الشكل التالى.

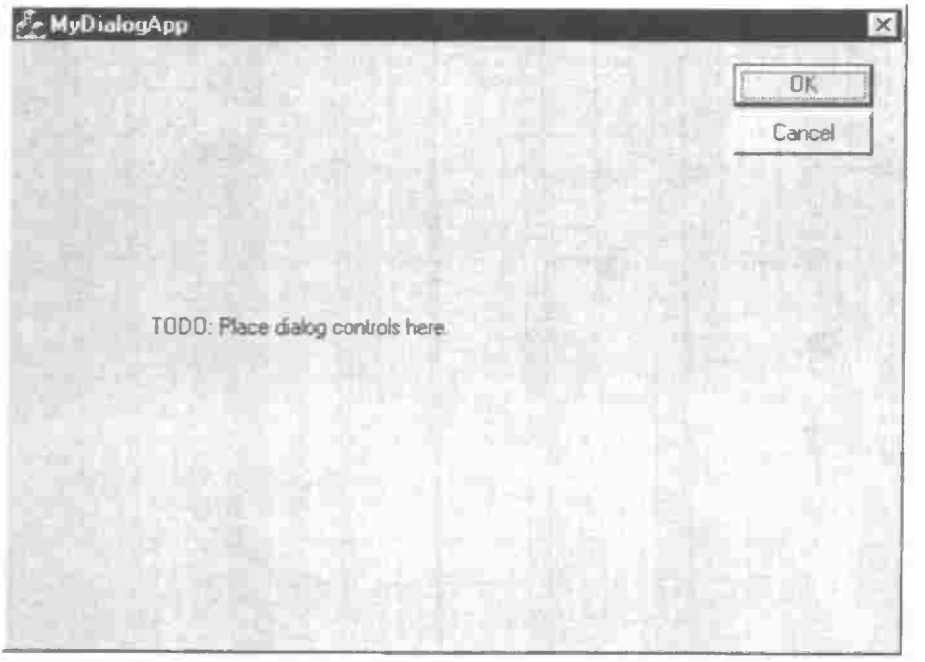


شكل (٥-٢٢) تطبيق على طراز الكشاف (Windows Explrer)

نفذ التطبيق وشاهد ملامحه ، ثم ألق نظـــرة علـــى مكوناتـــه مـــن الفصائل والملفات. سوف تلاحظ وجود فصيلة جديدة يختص بالقسم الأيسر من مشهد نافذة التطبيق بالاســـم **CLeftView**. يوجـــد هـــذا المشروع على القرص تحت الاسم:

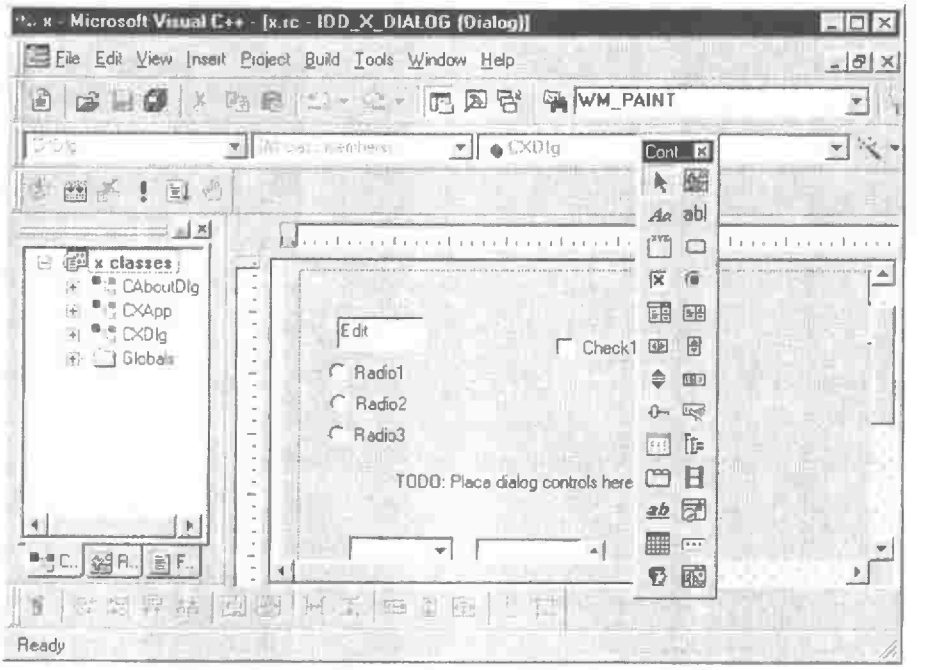
\ExamplesVCPP\Ch05\MyExpProcect

٣.من التطبيقات التى يبنيها الساحر أيضاً ، التطبيق المبنى علـــى صندوق حوار (Dialog Box). لبناء هـــذا المشـــروع اخـــتر زر الراديو Dialog based من نافذة الساحر فى الخطو رقم ١ (Step 1) ، ثم تمّم المشروع عند هذا الحد بالزر Finish. نفذ المشـــروع فتشاهد الشكل التالى:



شكل (٥-٢٣) مشروع مبنى على نافذة حوار (Dialog based)

ألق نظرة على الفصائل والملفات وسوف تلاحظ أن هذا المشروع أصغر بكثير من المشروعات النوافذية المعتادة ويحتوى على عدد أقل من الفصائل والملفات. كما تلاحظ ظهور سطر للأدوات في بيئة الأستوديو حيث يمكنك أن تسحب الأدوات وتسقطها فوق الصندوق لتكوين محتويات الصندوق. أنظر الشكل التالى.



شكل (٥-٢٤) بيئة المشروع المبني على صندوق حوار

يوجد هذا المشروع على القرص المصاحب بالدوسيه:

ExamplesVCPP\Ch05\MyDialogApp