掌握 Visual C++
——MFC 程序设计与剖析

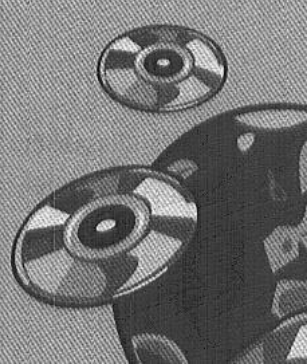
胡哲源　编著

清华大学出版社

第 1 章

Visual C++ 的集成开发环境

- Visual C++ 的外观
- 如何以 Application Wizard 生成一个新的项目
- 执行项目
- 这个新项目产生了什么文件
- 资源的编辑
- 菜单与选项的编辑
- 对话框的编辑
- 工具栏的编辑
- 鼠标指针 (cursor) 的编辑
- 图标的编辑
- 加速键的编辑
- 字符串表的编辑
- 查看 Source Symbol 与 ID 数值的更改
- Class Wizard
- Visual C++ 的调试功能
- 进入调试模式
- Visual C++ 的在线帮助说明



Visual C++ 集成开发环境 (IDE, Integrated Development Environment) 提供程序的编辑、编译、链接、执行与调试。另外也提供 Icon、对话框 (Dialog Box)、菜单 (Menu)与工具栏 (Toolbar) 等资源的编辑，以及强大的在线帮助说明 (On-line Help)。本章将一般常用的部份予以概略说明。

1.1 Visual C++ 的外观

刚进入 Visual C++ 时，我们只会看到一个典型的窗口，还有一些菜单与工具栏。等到我们载入程序或一个项目 (Project) 之后，整个窗口似乎动了起来。这是一个项目载入之后显现的窗口：

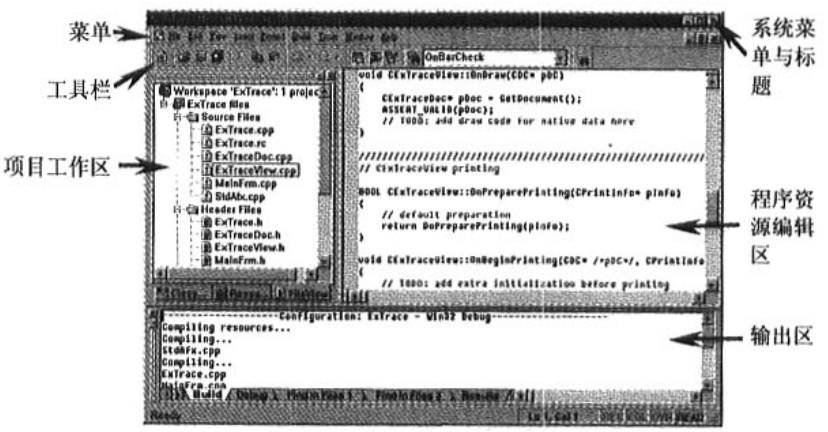


图 1.1

1.1.1 项目工作区

本区显示项目 (Project) 所包括之文件 (FileView)、类结构 (ClassView)与“资源”(Resource) 的清单 (ResourceView)。项目的内容，请见本章后文“如何生成一个新的项目”。

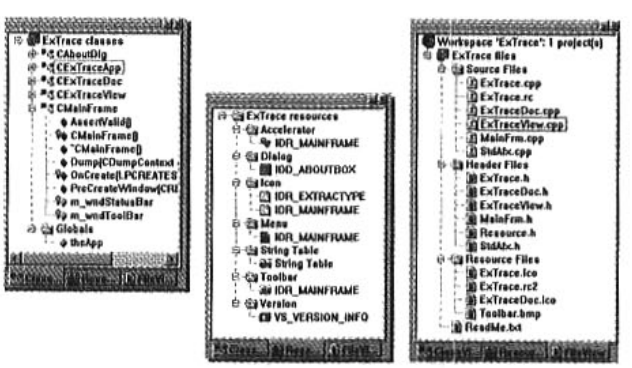


图 1.2

ClassView 显示项目中所包括的各个类(Class)的成员变量 (Member Variable) 与成员函数 (Member Function)，还显示全局变量与函数（含其 private、protected、public 属性，

请见第 2、3 章）。以鼠标左键双击某一个显示项，就会在“程序与资源编辑区”出现这个显示项的程序位置。如果以鼠标右键单击一个类，则会出现一个弹出菜单，允许您加入这个类新的成员变量、成员函数或派生类 (Derived Class) 等。ResourceView 显示项目中使用到的所有资源，在某一个资源项以鼠标左键双击，会在“程序与资源编辑区”出现这个资源的编辑页。以鼠标右键单击的话，则会出现一个弹出菜单，我们可以插入新的资源或进入这个资源的属性页等。

FileView 显示项目中用到的程序文件（含资源描述文件与资源文件）与 ReadMe.txt。在某一个程序文件以鼠标左键双击，会在“程序与资源编辑区”出现这个程序文件的内容。

1.1.2　输出区

Build 显示程序编译 (Compile)、链接 (Link) 的结果。Debug 显示程序调试执行时打印输出的结果。Find in Files 显示从文件搜寻一段文字的结果。

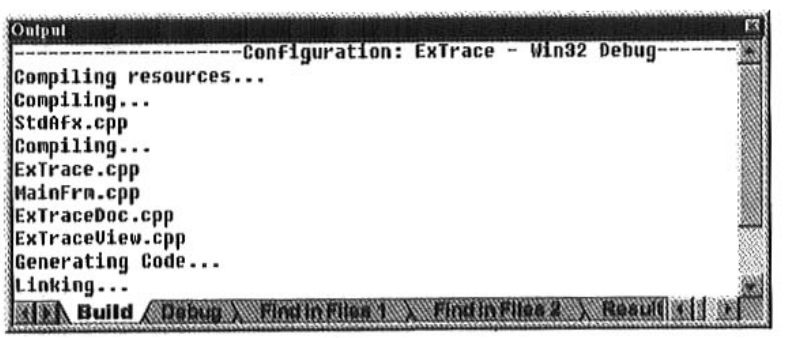


图　1.3

1.1.3　程序与资源编辑区

编辑程序与 (Dialog Box) 等。资源。资源是用户界面组件的集合，它包括 Icon、菜单 (Menu)、工具栏 (Toolbar)、对话框 (Dialog Box) 等。

1.1.4　调试时可查看的窗口

在程序调试 (Debug) 时，Visual C++ 另提供数种协助调试的工具窗口，我们在本章后文谈到调试时再作介绍。

1.2　如何通过 Application Wizard 生成一个新的项目

所谓项目，指的是您为了达成某个预期目标、在计划之下的所有作为。Visual C++ 项目的建立也是为此目的；它包括项目内所有文件的清单、使用的类、准备的资源、以及这个项目在 Visual C++ 的集成开发环境的设置状况等。只要取出一个项目文件，一切的内

容与设置都会立刻载入集成开发环境。

我们可以使用集成开发环境生成一个新的项目，或增减一个旧项目的内容。MFC Application Wizard 提供许多不同的项目，我们只介绍一个新项目的生成方法；以下是一个例子：

1. 在菜单上，按 File | New...
2. 选择 Projects 页、选择 MFC AppWizard [exe]、输入项目的目录并在 Project name: 输入 ExMFC，并按 OK

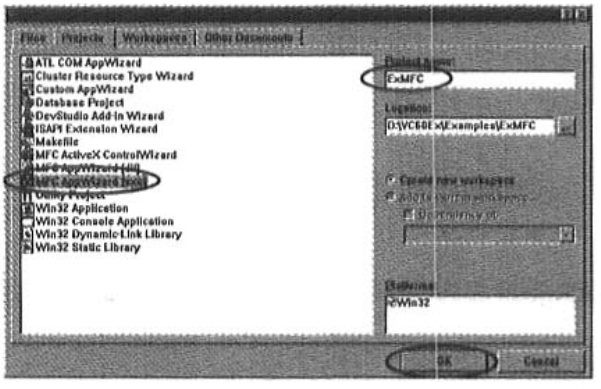


图 1.4

我们可以看到有多种不同的项目可以选择，也看到 Project name 自动默认为项目及其相关文件存放的目录名称。

3. 选择 Single document、【中文[中国][APPWZCHS.DLL]】与 Next

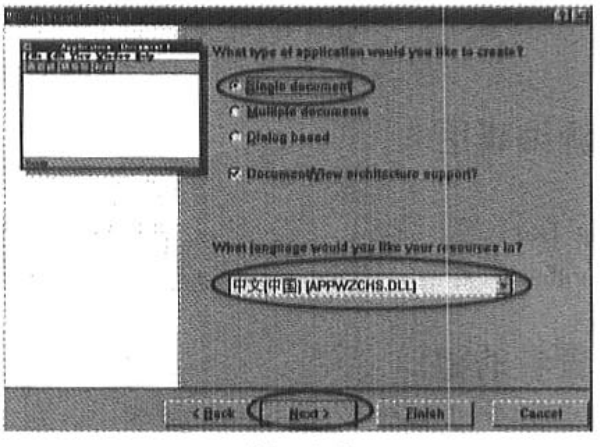


图 1.5

我们打算使用 document/view 的架构，也选择了 single document 机制（即项目中只允许一个 document 存在）。当然，我们使用中文系统。

4. 选择 Next>——我们暂时不需要数据库支持

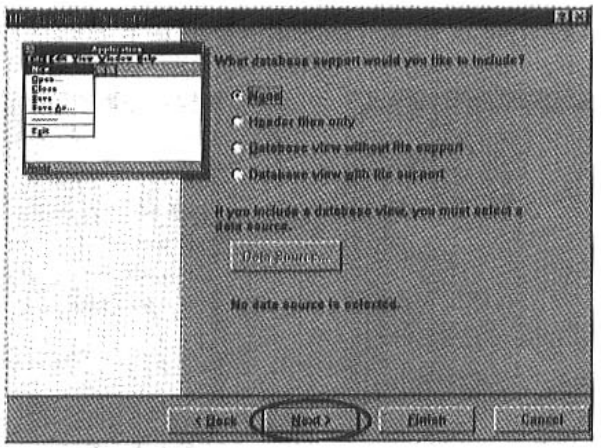


图　1.6

5.　选择 Next——默认

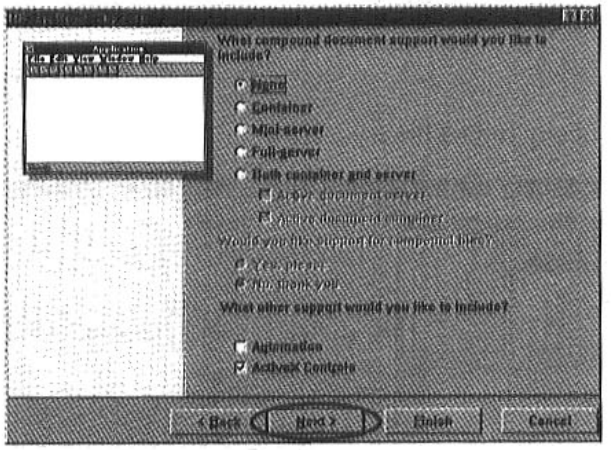


图　1.7

6.　选择 Next——默认

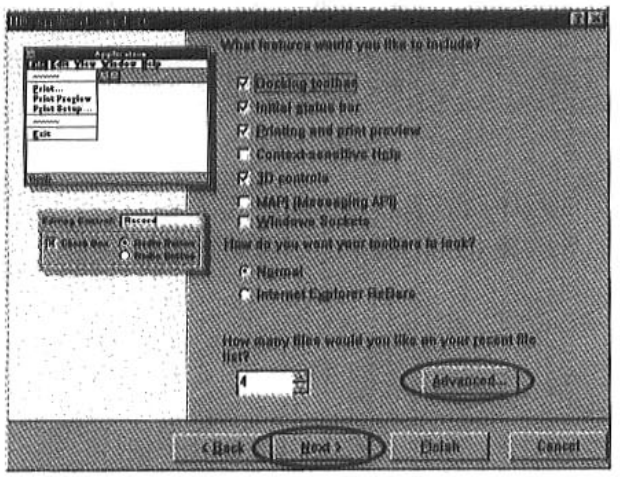


图　1.8

请读者试着将打 √ 的项目去除再恢复，看看左上角的窗口有何变化。最后还是要设置成上图的状态。按下 Advanced...之后更可定义一些与文件有关的字符串（亦见本章字符串

表的编辑）与选择窗口的 style 设置（请参考第 4 章与第 6 章的应用）：

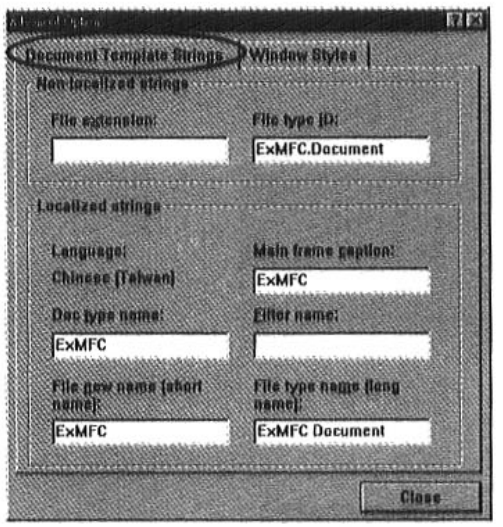


图 1.9

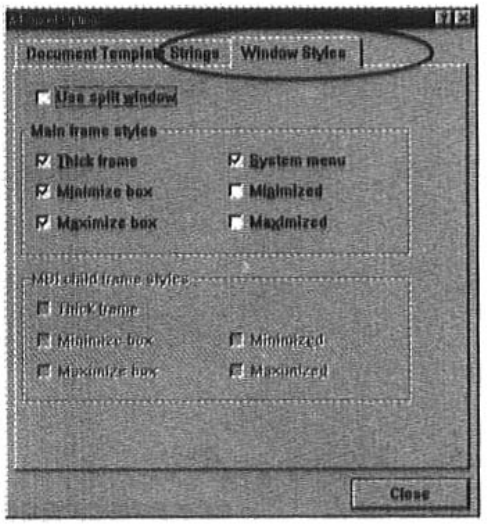


图 1.10

7. 选择 As a statically linked library、Next

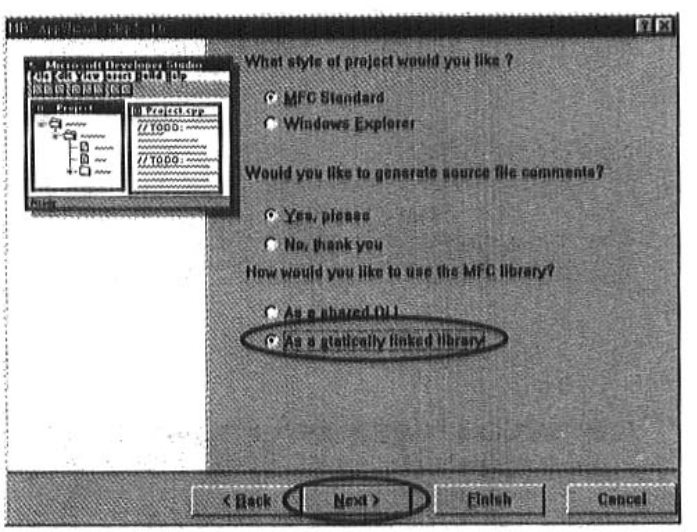


图 1.11

这是选择将 MFC 的程序以静态函数库的方式链接。

8. 在 Base class 处选择 CEditView 之后按 Finish，并在下一页按 OK 就完成了一个新的项目

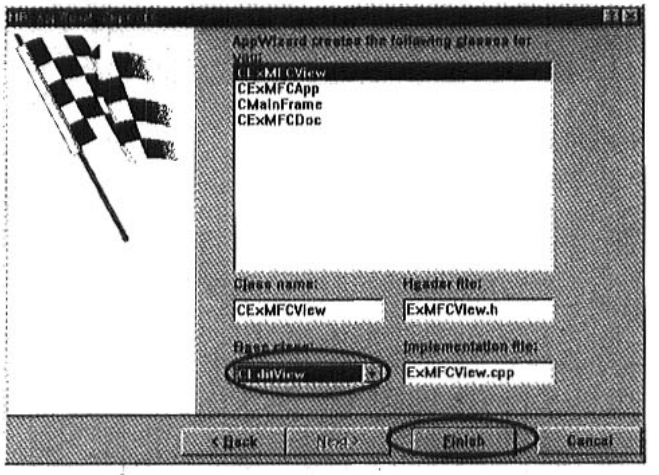


图　1.12

我们当然可以选择其他各种的视图，CEditView 窗口可以用来编辑文字。

1.3　执行项目

生成项目之后，我们便可以来执行这个项目。按下菜单的 Build | Execute ExMFC.exe 之后，我们在输出区看到程序被编译，链接：

```
Compiling resources...
Compiling...
StdAfx.cpp
Compiling...
ExMFC.cpp
ExMFCDoc.cpp
ExMFCView.cpp
MainFrm.cpp
Generating Code...
Linking...

ExMFC.exe - 0 error(s), 0 warning(s)
```

图　1.13

最后生成执行文件，并自动执行。结果我们看到一个似曾相识的窗口。它有一个标题列，当然也包括常见的系统菜单、一个菜单、一个工具栏、一个状态栏以及一个客户窗口区 (client area)。更神奇的是，这个窗口还可以编辑文字。

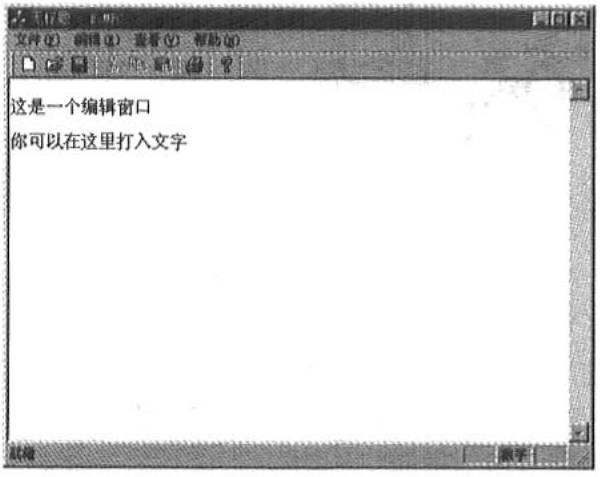


图 1.14

我们再看看菜单与工具栏给我们提供了什么功能。其实工具栏中的功能在菜单中都有，所以我们只要查看菜单的内容即可。

这一个简单的程序不但提供了文件存取、文件编辑、打印预览与打印的功能，同时也附加了显示 / 取消显示工具栏与状态栏的功能。

至此，不知读者有没有注意到，如此强大的“程序”竟然在不需要我们写下一行程序（也就是连 C++ 语言是什么都不需要知道）之下，只简简单单的利用 Application Wizard 就完成了。

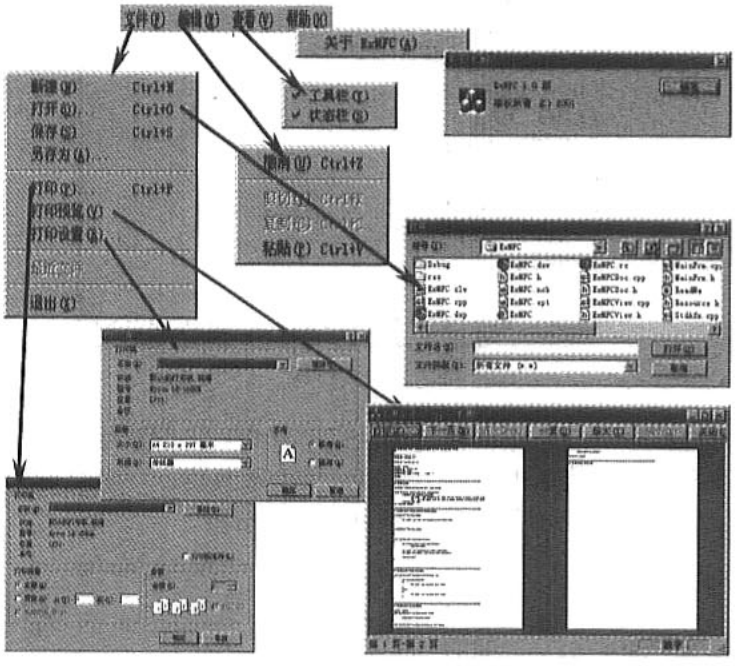


图 1.15

1.4　这个新项目产生了什么文件

如果我们到刚刚产生的 ExMFC 这个目录底下看看，我们会发现原来 Application Wizard 已经为我们产生了不少的程序文件。把 ReadMe.txt 这个文件打开来，您会看到一些说明，摘录如下：

ReadMe.txt

Application Wizard 已经为您产生了这个 ExMFC 项目，它不但展现 MFC 机制为您打下的框架，更提供您以此来扩展的程序。

以下是 Application Wizard 产生的程序文件摘要：

ExMFC.dsp — 定义项目 (Project) 的环境

ExMFC.clw — 存储 Class Wizard 所需数据

- ExMFC.h, ExMFC.cpp — 框架的主程序的头文件 (header file)与实现文件
- MainFrm.h, MainFrm.cpp — 窗口的主框架 (frame) 的头文件与实现文件
- ExMFCDoc.h, ExMFCDoc.cpp — document 的头文件与实现文件
- ExMFCView.h, ExMFCView.cpp — view 的头文件与实现文件

ExMFC.rc — 资源 (resource) 的描述文件

Resource.h — 定义资源中使用的一些 ID

另外在 res 子目录之下有：

ExMFC.ico, ExMFCDoc.ico — icon 文件

ExMFC.rc2 — 有些无法放置在 ExMFC.rc 的资源描述可以放在这里

Toolbar.bmp — 工具栏的 bitmap 文件

【注】

程序执行文件依您的设置（以【Set Active Configuration】），会产生并放入 Debug 或 Release 子目录中。

这些文件可以分成 3 类，一是配合 Visual C++ IDE 存储相关的环境与项目内容设置 (dsp、clw)，二是支持项目的资源描述文件 (rc、rc2) 与资源文件 (ico、bmp)，三是此项目的实现文件 (h、cpp)。

项目中，有 4 个主要的类，分别放在 4 个程序文件中，我们称为 SDI (Single Document Interface) Document/View 机制之下的 4 大类：

文　件	类　别	继承至（类）	简　称
ExMFC.h ExMFC.cpp	CExMFCApp	CwinApp	App
MainFrm.h MainFrm.cpp	CMainFrame	CframeWnd	Frame

续 表

ExMFCDoc.h ExMFCDoc.cpp	CExMFCDoc	Cdocument	Doc
ExMFCView.h ExMFCView.cpp	CExMFCView	CeditView （或其他 view 类）	View

SDI 指的是项目中只有一个负责 Document 的类；Document /View 机制的主要作用是将数据管理与数据显示的部份切分开来，让程序更有可读性。

App 是项目的主程序（不含一般常见的 main 或 WinMain 函数），它拥有项目唯一的一个对象，项目的其他对象都是从这个对象延伸出来的。Frame 安排窗口的外框，基本上包括菜单、工具栏、状态栏与一个客户区 (client area)。客户区的内容通常是由 View 来规划，可以显示文字或绘图，而应用数据的管理是交由 Doc 来负责的。在设计项目时，对于这种任务的安排虽然不见得要去遵守，但既然要使用 Document/View 的机制，我想，维护这个机制的基本精神，应该会让您的项目更友善一些。

1.5 资源的编辑

在 Visual C++ 集成开发环境中，定义一些用户界面或信息成为程序的“(Resource)，它包括加速键、对话框、icon、菜单、字符串表、工具栏、鼠标指针 (鼠标)、bitmap 图、HTML 及版本信息等。针对这些资源，Visual C++ 提供强大的编辑功能，在编辑之后自动的放入资源文件，真正的成为程序的资源。这些资源文件在程序编译时也是编译的对象。

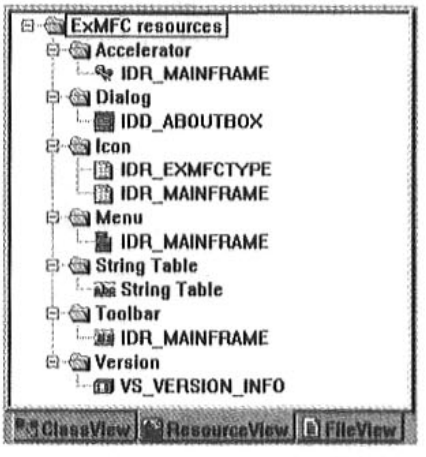


图 1.16

我们不但可以编辑出现在窗口的文字与图形，还可产生对应于某个用户界面组件的 ID、标题 (Caption)、提示 (Prompt)与其 style 设置等。除了默认的资源之外，我们也可以在 Visual C++ 的菜单上按 Insert、Resource...来产生这些新的资源：

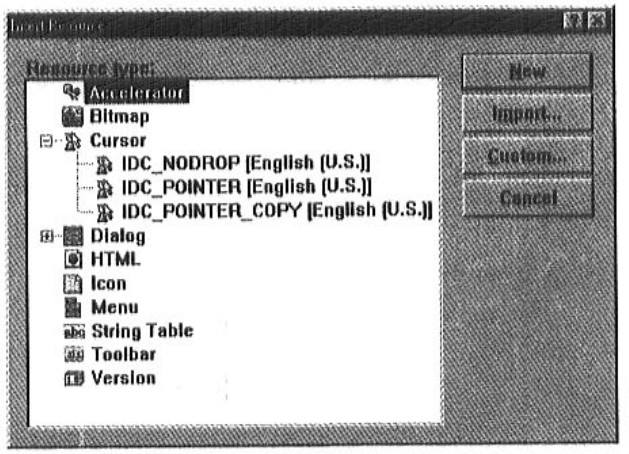


图　1.17

1.6　菜单与选项的编辑

菜单 (menu) 是由选项 (menu item) 所组成。选项可以是一般选项，也可以是附有子菜单 (submenu) 的选项。子菜单可以层层迭迭的延伸下去：

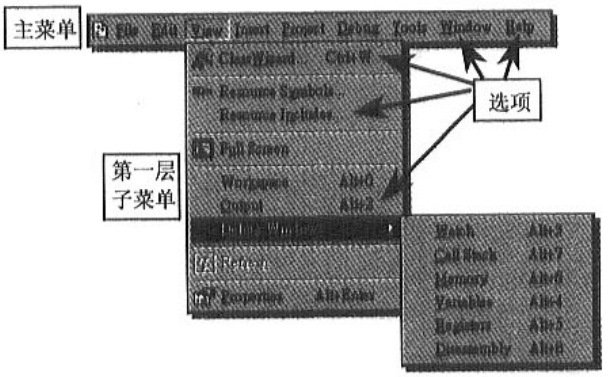


图　1.18

以前述的 ExMFC 项目为例，它自动默认了一个菜单给我们使用。当我们在“资源”的菜单中单击 IDR_MAINFRAME，将进入这个默认菜单的编辑页。

一个选项可以在这个编辑器之下增减、修改。让我们以新加入的一个选项名称为“新加选项”的选项为例，先看看加入之后的结果：

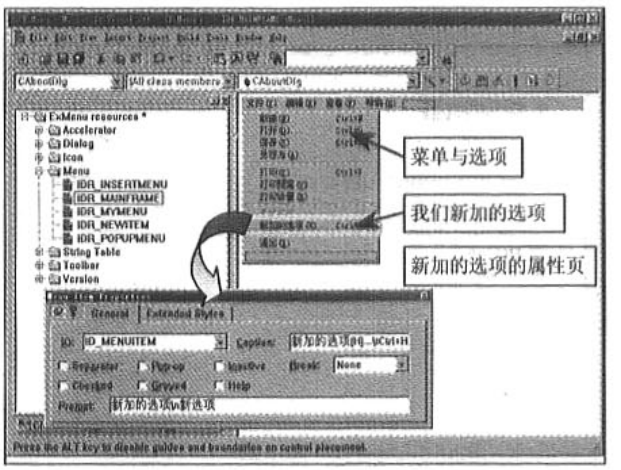


图 1.19

这个选项加在【退出(X)】选项之前，所以我们要将鼠标放在【退出(X)】这个选项，再按一下键盘上的 Insert 就可以产生一个空白的新选项了。在这新选项上按两下鼠标左键或按下鼠标右键选择 Properties，就可以出现如上图的选项属性页。

我们在选项属性页上设置了选项 ID、定义字 (Caption) 与提示(Prompt)。在定义字中，我们设置“新加选项(&H)...\tCtrl+H”。其中，第一个设置为选项的名称，它出现在菜单中选项的左方。第二个定义选项在菜单下拉之后此选项的热键 (Mnemonic key)，本例给予 (&H)，代表当菜单已经下拉之下，在键盘上按下字母 H 时，此选项被选择；它跟在选项名称的后面。第三个定义选项在窗口中的加速键 (Accelerator key)，本例给予 Ctrl+H。当程序的窗口为工作窗口时，在菜单不用下拉之下，在键盘同时按 Ctrl 及 H 键可以快速的选择此选项；它出现在菜单中选项的右方，而所谓的左方及右方其实是以 tab (\t) 来分开。如果您在此定义了加速键，则它的信息也会出现在加速键资源之中。

此例的提示为“新加的选项\n 新选项”。所谓的提示，是定义当鼠标指针 (鼠标) 指在某个选项时，出现的选项功能提示。第一个设置为选项出现在状态栏的提示，本例给予“新加选项”。如果此选项的 ID 也是工具栏上的一个按键，当鼠标指针移到这个按键时，第二个设置就成为此按键的提示，它会出现在这个按键的旁边--本例给予“新选项”，这两个提示之间是以 \n 隔开。

如果将这个选项设置为 Pop-up，它代表后面将跟着一个子菜单，且这个选项不必给予 ID，例如：

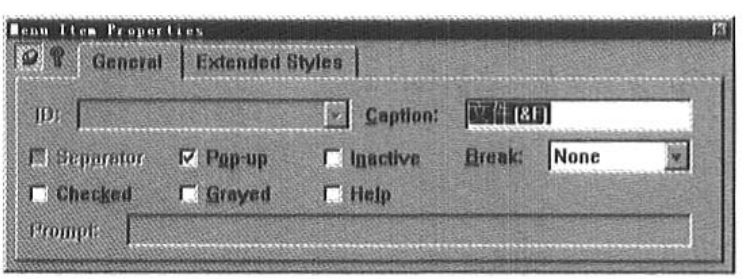


图 1.20

选择 Help 时，这个选项与后面跟着的选项，将在执行时依序出现在菜单条的最右边。

当您输入的中文选项名称无法在执行时显示时，请在 IDR_ MAINFRAME 的地方按鼠标右键，再选择 Properties 将会出现一个菜单的内容属性页。在 Language 处选择 Chinese [P.R.C] 即可：

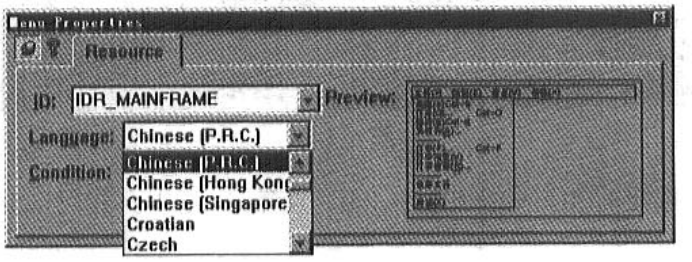


图　1.21

1.7　对话框的编辑

在菜单上按 Insert、Resource...，将会出现一个要插入的资源类型选择页。如果只是要插入一般的对话框，直接单击 Dialog 即可产生一个新的对话框。除了一般的对话框，我们可以选择 对话栏 Form View 的对话与分层的属性页对话（可选择 小/中/大三种大小）。

不管选择那一种 dialog，都可以选用控件 (Controls) 工具栏上的控件，以鼠标左键拉到对话之上使用。控制工具栏在打开对话时会一起出现。如果您不小心删除了它，请在菜单或一般工具栏以鼠标右键重新选取。

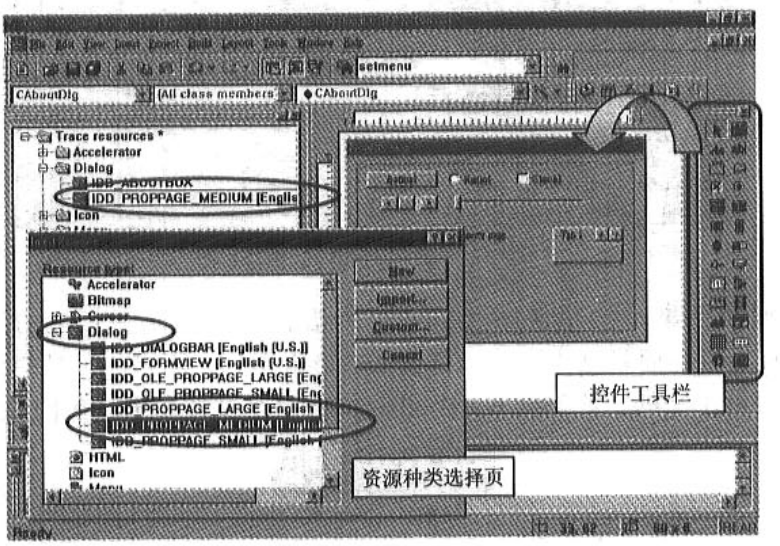


图　1.22

以本例来说，我们利用资源种类选择页产生了一个 ID_PROPPAGE _MEDIUM dialog，也在图右方的控件工具箱中选用了一些控件，如 pushbutton、单选按钮、check box、slider、tab control 等。每一个控件都有一个属性页，我们只要在某一个控件上以鼠标右键单击 properties 即会出现。以一个单选按钮为例，它的属性页为：

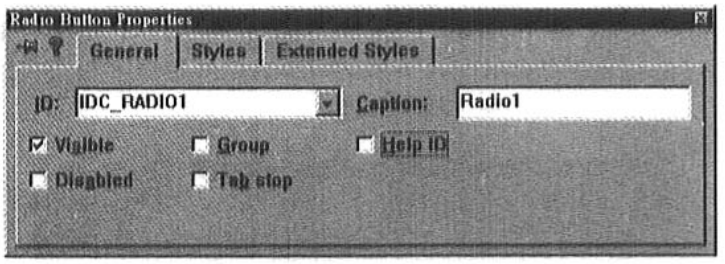


图 1.23

这个属性页有按钮的 ID、按钮的文字等；在第 7 章中会有对话框与控件的详细使用说明。

1.8 工具栏的编辑

如同菜单一样，利用 Application Wizard 产生的基本程序也可以帮我们自动产生一个 ID 名称为 IDR_MAINFRAME 的工具栏 (toolbar)，它执行的时候是这样的：

图 1.24

如果我们在资源中这个 ID 之上以鼠标右键单击，将会出现这个工具栏的属性页：

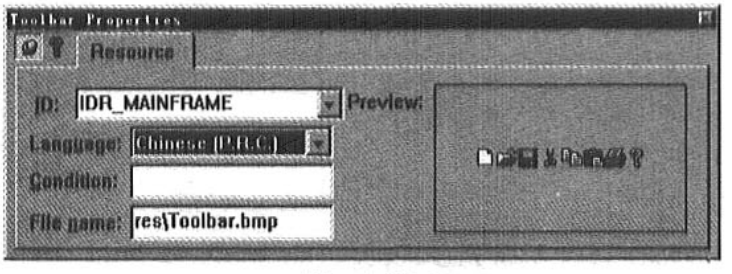


图 1.25

它告诉我们两件事，一是使用 Chinese 为语言，二是此工具栏的 bitmap 图存在 res\Toolbar.bmp 之中，而且这张图的预览也在右方看得到。原来，工具栏上按键的图是由一张 bitmap 图分解出来的。我们把 res\Toolbar.bmp 这个文件拿出，我们看到：

图 1.26

这是一张宽 128 × 高 15 的位图图像。似乎是把这张图按照一个“规定”裁剪，贴到工具栏的按键上的。可是“规定”是什么？应该是每个工具栏的按键分配到的图的大小吧？如果我们关掉资源编辑器，再以 Open 将 ExMFC 项目的资源文件 ExMFC.rc 以 text 类型打开，我们会找到这些叙述：

程序清单　1.1

```
// Bitmap
IDR_MAINFRAME  BITMAP  MOVEABLE PURE  "res\\Toolbar.bmp"

// Toolbar
IDR_MAINFRAME TOOLBAR DISCARDABLE  16, 15
BEGIN
.....
```

原来，这边定义每个工具栏的按键分配到的图的大小为宽 16 点，高 15 点。这个定义不是这么麻烦才看得到，在工具栏编辑器编辑的时候也看得到。我们先以工具栏编辑器将 IDR_MAINFRAME 工具栏打开。我们看到一个常见的工具栏：

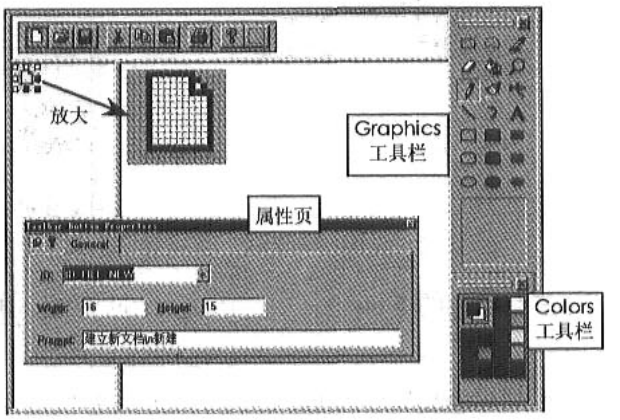


图　1.27

单击第一个按键【新建】，属于这个按键的图就会放大出现。右边会出现两个工具栏，一个叫作 Graphics 工具栏，另一个叫作 Colors 工具栏。如果您不小心删除了它们，请在菜单或一般工具栏以鼠标右键重新选取。我们可以利用这两个工具栏来编辑 IDR_MAINFRAME 工具栏的每一个按键。

如果您数一数这个放大的【新建】按键上的方格数，也应该是宽 16 点，高 15 点。更简单的方法是，以鼠标左键在工具栏编辑器的空白处点两下，【新建】按键的属性页就会出现（如上图）。在属性页上，我们就看到按键的大小的定义。除此之外，此按键的 ID 与提示也会出现，此按键定义为“建立一个新文件\n 新文件”。其中，第一个定义为出现在状态栏的提示，第二个定义是出现在工具栏按键旁的提示，也就是所谓的 tooltip。

工具栏上的按键位置可以移动，也可以添加、删除、复制、改变大小等。

1.9　鼠标指针的编辑

在菜单上单击 Insert、Resource...，可以选择产生一个新的空白的鼠标或三种默认的鼠标，以供编辑。新定义的鼠标可以提供给程序使用。

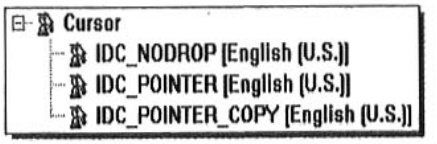


图 1.28

鼠标图形的墨绿色背景，代表透明颜色，也就是窗口画鼠标的程序遇到这个颜色则不显示。鼠标图形默认为允许黑白两色(monochrome)，所以在 Colors 工具栏只呈现一些中间色调的样式以供使用。

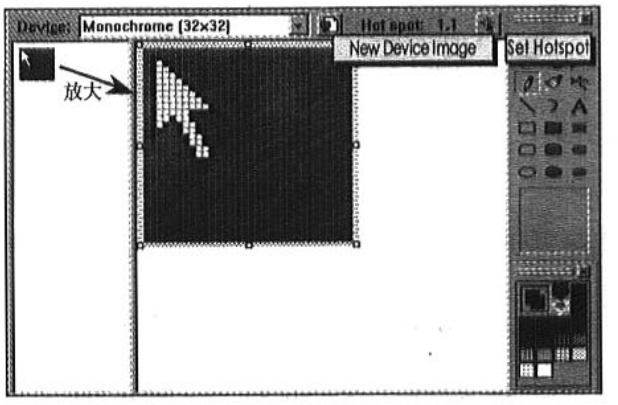


图 1.29

我们当然也可以利用在图上方的 New Device Image (Ins)按键 来改变颜色多少的设置。

我们另外看到一个 Set Hotspot 按键 ，它是用来设置鼠标的中心坐标。例如上图鼠标的热点（hotspot）定在 (1,1)，代表这个图的坐标 (1,1) 的点，就是在项目窗口上鼠标的位置。我们只要先单击 按键，再以鼠标左键在鼠标图形上的某一点单击一下，这一点就成为此鼠标新的中心坐标。

1.10 图标的编辑

图标的编辑，工具栏的编辑，与鼠标的编辑同属图形编辑。在编辑器的右边都会出现 Graphics 工具栏与 Colors 工具栏。只不过其生成的图形文件格式不同，而且颜色的限制也不同。以 ExMFC 项目来说，它自动提供两个图标 ID。一个是 IDR_ EXMFCTYPE，它的图形 ，通常是在 MDI (Multiple Document Interface) 之下，给文档窗口做为系统菜单的图标。另一个 ID 是 IDR_MAINFRAME，它是 ExMFC 主窗口系统菜单的图标 (16×16)，同时也是 About Dialog 的图标 (32×32)。