

Visual C++ .NET 学习教程

冯 永 王占斌 孟凯翔 编著



2C-43

北京大学出版社
<http://cbs.pku.edu.cn>

Visual C++.NET 学习教程

冯 永 王占斌 孟凯翔 编著



A1000102

北 京 大 学 出 版 社

北 京

内 容 简 介

Visual Studio.NET 是 Microsoft 公司推出的新一代可视化开发工具, 又称为 Visual Studio 7.0。而 Visual C++.NET (Visual C++ 7.0) 是这个工具集的重要组成部分, 它主要用于开发 Windows 应用程序以及企业规模的 Web 应用程序。它改善了传统的编程手段, 使得程序员可以直接在用户界面良好的可视化开发环境中进行工作。.NET 的全称是 .NET Framework (.NET 框架), 它是一种用于创建、配置和运行 Web 应用程序的多语言环境, 此框架提供了一种通用语言运行库 (Common Language Runtime)。Visual C++.NET 还集成了多种有用的工具, 从而大大提高了应用程序的开发效率。

本书以简洁、通俗的语言, 全面系统地介绍了 Visual C++.NET 的语言基础和编程技巧。本书图文并茂、通俗易懂, 既适合初学者入门, 也适合中等程度的读者使用。

图书在版编目 (CIP) 数据

Visual C++.NET 学习教程/冯永等编著. —北京: 北京大学出版社, 2001.11
ISBN 7-301-05304-5

I. V… II. ①冯…②王…③孟… III. Visual C++.NET—程序设计—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 076920 号

书 名: Visual C++.NET 学习教程

著作责任者: 冯 永 王占斌 孟凯翔

责任编辑: 段志刚 黄庆生

标准书号: ISBN 7-301-05304-5/TP·0625

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电子信箱: xxjs@pup.pku.edu.cn

印 刷 者: 河北省滦县印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 17.5 印张 448 千字

2001 年 11 月第 1 版 2001 年 11 月第 1 次印刷

定 价: 28.00 元

前 言

随着电脑的日益普及以及计算机硬件的迅猛更新,计算机可视化技术得到了越来越广泛的应用和发展,越来越多的计算机专业人员甚至非计算机专业人员都开始致力于可视化的研究与开发。在种类繁多的可视化工具中,Microsoft 公司的 Visual Studio 一枝独秀,受到广大开发人员的青睐。

最近,Microsoft 公司推出了其最新版本——Visual C++.NET,与以前版本相比,Visual C++.NET 具有以下新功能:

◆ 改善和优化了编译器和链接器

在编译器方面,增加了/CLR、/Fx、/GH、Gz 等选项,通过/Gz 编译选项在程序代码中嵌入运行期错误检验,可以检测出常见错误;在链接器方面,增加了一些新的链接选项,如/ALLOWBIND、/IDLOUT、/IGNOREIDL 等。

◆ 调试器功能更加强大

Visual Studio 7.0 为其中包含的所有开发语言提供了共同的集成调试器。新的调试器不仅包括原来的 Visual C++ 6.0 和 Visual Basic 6.0 的调试器功能,还添加了许多新的特性。这些新特性包括: Visual Basic、Visual C++、C#、C++、脚本和 SQL 语言的交叉调试; Win32 应用程序的调试; Visual C++ 动态错误检查和缓存区安全检查等。

◆ 更方便的、更实用的 IDE 环境

在 Visual C++.NET 中,扩展了 ClassView 的功能;属性窗口提供了更优化的表格,用来查看和修改对象的各种属性。

◆ 编辑器功能更加完善

首先,在文本编辑器中提供了自动语句补充的功能。当用户编辑代码时,在光标的位置将显示出类的成员、函数原形、标志符声明以及代码注释等,大大减轻了程序员负担;其次,资源编辑器提供了对 IE 5.5 控件的支持。

◆ 更多的 Visual C++ 实例程序

在 Visual C++.NET 中增加了许多与 ATL 服务器、属性和 C# 等内容有关的实例。

本书以简洁、通俗的语言,全面系统地介绍了 Visual C++.NET 的语言基础和编程技巧。全书共分 9 章。分别介绍了 Visual C++.NET 的编程环境 (IDE); Visual C++.NET 中的一种新语言——C#; Windows 的消息机制; MFC AppWizard 的使用; 菜单、工具栏和状态栏的制作及其使用; 可视化编辑中常用的工具之一——对话框和控件; 怎样在程序窗口中绘图; Visual C++ 在数据库中的应用; 多线程编程的基本原理及其实现方法。

本书由冯永、王占斌、孟凯翔主编,另外,韩永鹏、魏宝晶、朱明峰、袁怡翔、肖依永、邱亚希、苗泽纯、孟鸽、喻湘宁、张晟、盖宇、黄涛、朱凯、刘梓超、翟丽君、裴浴、叶萱、谢宏威、杨继运等也参加了本书的编写工作。由于编者水平有限,本书难免有不足之处,欢迎广大读者批评指正。

编 者

2001 年 11 月

目 录

第 1 章 Visual C++ 集成开发环境	1
1.1 启动 Visual C++ .NET	1
1.2 Visual C++ .NET 的用户界面	2
1.2.1 Visual Studio .Net 起始页	2
1.2.2 Visual Studio .Net 各窗口功能	6
1.3 Visual C++ .NET 的菜单功能	22
1.3.1 File 菜单	23
1.3.2 Edit 菜单	25
1.3.3 View 菜单	26
1.3.4 Project 菜单	27
1.3.5 Build 菜单	28
1.3.6 Debug 菜单	29
1.3.7 Tools 菜单	30
1.3.8 Windows 菜单	31
1.3.9 Help 菜单	32
第 2 章 掌握 C# 语言	34
2.1 C# 语言简介	34
2.1.1 简单性	35
2.1.2 现代性	35
2.1.3 面向对象性	35
2.1.4 类型安全性	36
2.1.5 版本处理技术	36
2.1.6 兼容性	36
2.1.7 灵活性	36
2.2 公用语言运行环境	37
2.2.1 中间语言和元数据	37
2.2.2 即时编译器	38
2.3 虚拟对象系统	38
2.3.1 VOS 类型系统	39
2.3.2 元数据	39
2.3.3 公用语言规范	39
2.3.4 虚拟运行系统	41

2.4 C#的数据类型	42
2.4.1 简单类型	42
2.4.2 结构类型	44
2.4.3 枚举类型	44
2.4.4 对象类型	45
2.4.5 类 (Class) 类型	45
2.4.5 接口	45
2.4.6 代表 (Delegate)	46
2.4.7 string 类型	46
2.4.8 数组 (Arrays)	46
2.4.9 装箱 (Boxing) 和拆箱 (UnBoxing)	47
2.5 控制语句	48
2.5.1 if 语句	48
2.5.2 switch 语句	50
2.5.3 for 语句	53
2.5.4 foreach 语句	55
2.5.5 while 语句	56
2.5.6 do 语句	57
2.6 第一个 C# 小程序	58
2.6.1 Hello World 程序代码	58
2.6.2 编译程序	60
2.6.3 输入和输出	60
2.6.4 为程序添加注释	61
2.7 C# 中的类	62
2.7.1 构造器与析构器	62
2.7.2 类中的方法	63
2.7.3 C# 中的事件	70
2.7.4 C# 中的修饰符	71
2.7.5 类的属性	73
2.7.6 索引指示器	74
2.8 异常处理	76
2.8.1 检测语句和不检测语句	76
2.8.2 异常处理语句	78
2.8.3 抛出异常	83
2.8.4 异常处理中应注意的事项	84
第 3 章 Windows 基本概念与消息机制	86
3.1 与窗口有关的基本概念	86
3.1.1 窗口	86
3.1.2 桌面窗口	87

3.1.3 父窗口	87
3.1.4 子窗口	87
3.1.5 窗口名	87
3.1.6 窗口句柄	87
3.1.7 框架、客户以及子窗口	88
3.1.8 实例句柄	88
3.1.9 子窗口和父窗口的关系	88
3.1.10 禁止窗口	88
3.1.11 活动窗口	89
3.1.12 窗口可见性	89
3.1.13 窗口过程	89
3.1.14 窗口属性	89
3.2 有关消息的概念	89
3.2.1 消息	89
3.2.2 消息路由	90
3.2.3 投递消息和发送消息	90
3.2.4 消息种类	91
3.2.5 消息过滤	91
3.3 MFC 的消息和消息处理	91
3.3.1 MFC 事件驱动程序设计的引擎 —— CWinApp 类	92
3.3.2 消息映像和消息处理	96
3.4 鼠标消息处理	100
3.4.1 用户区鼠标消息	100
3.4.2 非用户区消息	102
3.4.3 鼠标捕捉	104
3.5 键盘和键盘消息处理	106
3.5.1 键盘消息	106
3.5.2 按键消息及其处理	106
3.5.3 字符消息及其处理	107
第 4 章 应用程序框架	108
4.1 MFC Application Wizard 的启动	108
4.2 使用 MFC Application Wizard 生成应用程序框架	110
4.3 应用程序框架分析	115
4.3.1 应用程序对象	116
4.3.2 文档模板	119
4.3.3 主框架窗口对象	120
4.3.4 文档对象	123
4.3.5 视图对象	126
4.3.6 子框架窗口类	128

第 5 章 菜单、工具栏和状态栏	131
5.1 设计和使用菜单	131
5.1.1 编辑菜单资源	132
5.1.2 菜单消息响应	137
5.1.3 菜单的 UI 机制	138
5.1.4 动态改变菜单	139
5.1.5 创建浮动式菜单	140
5.2 设计和使用工具栏	141
5.2.1 CToolBar 类的主要成员函数	141
5.2.2 工具栏的创建	144
5.2.3 处理命令消息	146
5.2.4 工具栏的 UI 机制	147
5.2.5 工具栏的停靠和漂浮	147
5.2.6 显示和隐藏工具栏	149
5.2.7 实例：在应用程序中添加工具栏按钮	149
5.3 设计和使用状态栏	154
5.3.1 CStatusBar 类的主要成员函数	154
5.3.2 获得对状态栏的控制	156
5.3.3 管理消息栏和状态指示窗格	157
5.3.4 StatusBar 例程	157
第 6 章 对话框和控件	165
6.1 对话框资源	165
6.2 CDialog 类简介与对话过程函数调用顺序	167
6.3 使用消息框	168
6.4 创建自定义的对话框	170
6.4.1 创建应用程序框架和对话框资源	170
6.4.2 在对话框中添加控件	172
6.4.3 创建对话框类	175
6.4.4 实现对话框	176
6.4.5 实现对话数据交换和验证	179
第 7 章 Windows 绘图	183
7.1 设备环境	183
7.1.1 设备环境 DC 简介	183
7.1.2 CDC 类及其派生类	184
7.1.3 CDC 类的成员函数	186
7.1.4 CDC 类派生类之间的区别	187
7.1.5 CDC 类及其派生类的使用方法	187

7.2 GDI 对象	189
7.2.1 画笔对象	189
7.2.2 画刷对象	192
7.2.3 字体对象	196
7.2.4 区域对象	198
7.2.5 位图对象	202
7.2.6 调色板对象	207
7.3 映射模式和坐标变换	217
7.3.1 映射模式简介	217
7.3.2 设置映射模式	218
7.3.3 设备坐标与逻辑坐标简介	218
7.3.4 设备坐标与逻辑坐标之间的变换	219
7.3.5 设备坐标与逻辑坐标的映射关系	219
7.4 用鼠标绘图	221
7.4.1 鼠标消息的发送	221
7.4.2 捕获鼠标和设置鼠标形状	224
7.4.3 用鼠标绘制基本图形	225
第 8 章 数据库编程	228
8.1 ODBC	228
8.1.1 常用术语	228
8.1.2 ODBC 与标准化	229
8.2 SQL 语法基础	231
8.2.1 结构化查询语言	231
8.2.2 关系对象	232
8.2.3 SQL 分类	233
8.2.4 SQL 语法	233
8.3 ODBC 的实现	238
8.3.1 配置 ODBC 环境	238
8.3.2 连接数据源	239
8.3.3 查询数据和数据源	241
8.3.4 装备和执行 SQL 请求	242
8.3.5 检索数据	242
8.3.6 释放 ODBC 环境	243
8.4 ODBC 的应用	244
8.4.1 数据库的创建	244
8.4.2 配置 ODBC 环境	246
8.4.3 创建 ODBC 应用程序	249
8.4.4 对 ODBC 应用程序进行配置	252

第 9 章 多线程编程	262
9.1 Win32 基础上的多线程编程	263
9.1.1 用 Win32 函数创建和中止线程	263
9.1.2 用 Win32 函数控制线程对共享资源的访问	263
9.2 基于 MFC 的多线程编程	265
9.3 MtRecalc 例程分析	267

第 1 章 Visual C++ 集成开发环境

本章包括:

启动 Visual C++.NET

Visual C++.NET 的用户界面

Visual C++.NET 的菜单功能

这一章详细介绍 Visual C++.NET 可视化集成开发环境。Visual C++.NET、Visual Basic.NET 和 Visual Foxpro 使用相同的用户界面,即 Visual Studio.NET 集成开发环境 (IDE Integrated -Development Environment)。使用同一个集成开发环境为应用程序开发者提供了很大方便。Visual Studio.NET 集成开发环境功能非常强大,具有丰富的工具集、众多的窗口、详细的菜单,因此在开始就全部掌握比较困难,建议初学者可先快速浏览这一章,对 Visual Studio.NET 开发环境有一个初步的总体认识就行了,没有必要搞清楚每一个菜单项和工具条按钮的功能,对有些名词一时不能理解也没有关系,以后的章节会有详细的介绍。当读者进一步编程的过程中遇到问题的时候,或者经过一段时间对整个环境比较熟悉了但还想知道 Visual C++.NET 提供了自己所不知道的新功能,这时可以回过头来再参阅本章内容,相信读者一定会有新的收获。

1.1 启动 Visual Studio.NET

在启动 Visual C++.NET 以前,必须安装 Microsoft Visual C++.NET。作为新一代强有力的开发工具,Microsoft Visual C++.NET 能够运行的条件如下:

- 使用 486 或更高级处理器 (推荐使用 Intel Pentium 处理器)。
- Microsoft Windows NT Server 3.15 或更高版本 Windows 2000 /sp-1。
- 16MB 内存 (复制需要 32MB 内存)。
- 380MB 以上硬盘空间。
- CD-ROM 驱动器。
- Microsoft Mouse 或兼容设备。

Visual Studio.NET 的启动步骤如下:

(1) 单击 Windows 2000 的 “Start” 按钮。

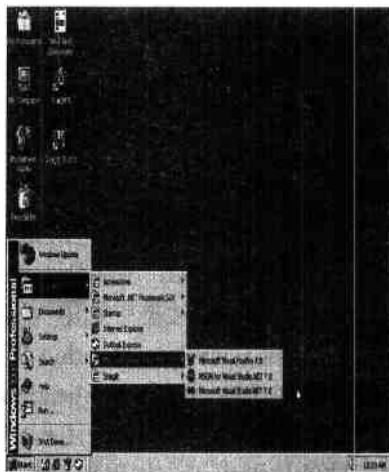


图 1.1 Visual C++ 7.0 的启动

- (2) 在弹出的菜单中单击“Programs”菜单。
 - (3) 将鼠标指针移动到“Microsoft Visual C++.NET 7.0”命令。
 - (4) 在弹出的菜单中单击“Microsoft Visual C++.NET 7.0”命令。
- 如图 1.1 所示, 这时 Microsoft Visual C++.NET 开始启动。

1.2 Visual C++.NET 的用户界面

Microsoft Visual Studio.NET 的界面风格较之 6.0 版本有很大改动, 给人以耳目一新的感觉, 而且其界面是智能化的, 非常宽容并且鼓励开发者去实践和尝试。

1.2.1 Visual C++.NET 起始页

Visual C++.NET 刚启动时呈现的界面称为 Visual Studio 起始页。如图 1.2 所示。Visual Studio 起始页中的各项功能如下:

- Get start on a recent, existing or new project (最近工程名)

如果开发者已经使用 Visual C++.NET 7.0 创建或编辑过工程, 则在 Visual Studio 起始页中会相应的显示出最近打开过的工程名, 其日期也会相应的显示出。如图 1.2 中所示的 Project8、Project7、Project6 和 Project1。其中, Project8 工程是启动 Visual C++.NET 的前一天晚上 10:39 打开过, Project7 工程是启动 Visual Studio.NET 的前一天打开过, 而 Project6 工程在 2001 年 5 月 11 日晚上 7:24 打开过。

- Open Existing Project (打开现存工程)

点击该项将会弹出 Open Project (打开工程) 对话框。如图 1.3 所示。

在图 1.3 中双击其中的任一个工程, 即可打开这个工程。如果将“Add to solution”复选框选中, 则可将此工程加入到一个 Solution 方案 (也可称为工程组中)。

- Creating New Project (创建新工程)

点击该项将会弹出 New Project (新工程) 对话框。如图 1.4 所示。

在图 1.4 中的左边框中选中 Visual C++ Projects 文件夹, 则右边方框中会相应的出现模板类型。下面以 MFC Application 模板为例, 简单介绍一下 MFC 应用程序的生成。在图 1.4 的右面方框中选中 MFC Application 模板, 然后在 Name (名字) 项中为应用程序命名, 在 Location (位置) 项中键入应用程序所要存放的位置, 或者单击“Browse”按钮, 直接选定所要存放的位置也可。如果想要将工程加入到方案中, 需要将“Add to Solution”选项按钮选中。然后单击“OK”按钮, 弹出 MFC 应用程序向导对话框。如图 1.5 所示。

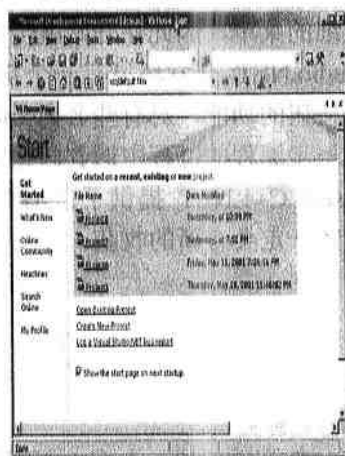


图 1.2 Visual Studio 起始页

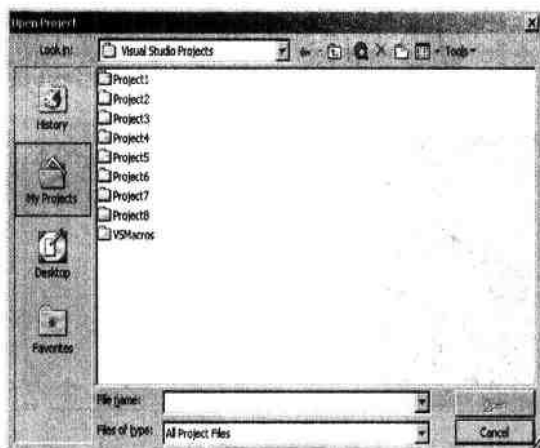


图 1.3 Open Project (打开工程) 对话框

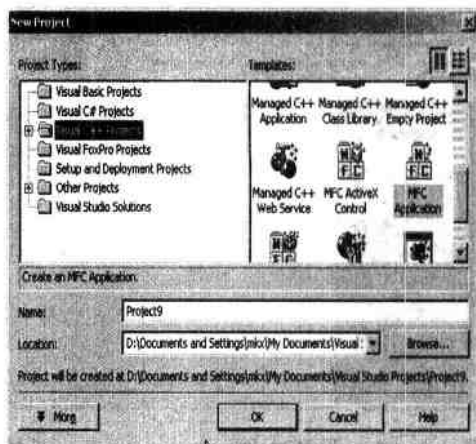


图 1.4 New Project (新工程) 对话框

在图 1.5 所示的对话框左面蓝色的部分有 Overview (综述)、Application Type (应用程序类型)、Compound Document Support (复合文档支持)、Document Template String (文档模板字符串)、Database Support (数据库支持)、User Interface Features (用户接口特性)、Advanced Features (高级特性)、Generated Classes (已创建的类) 一共八个选项。单击这些选项然后可以进行对应用程序的设置。

在 Overview (综述) 项中对即将创建的应用程序作简单的介绍。单击“Application Type (应用程序类型)”选项，对话框如图 1.6 所示。图 1.6 中在 Select application Type (选择程序类型) 条目下，可选择 Single Document (单文档类型)、Multiple Document (多文档类型)，等等类型。在 Select Project Style (选择工程风格) 条目下可对工程风格进行设置。在 Select use of MFC (选择 MFC 程序的使用方式) 条目下，可以选择是将 MFC 应用程序用作动态连接库还是静态库。在 Resource language (语言选择) 条目下，有一个组合列表框，单击此列表框可选择英语或其他语言。

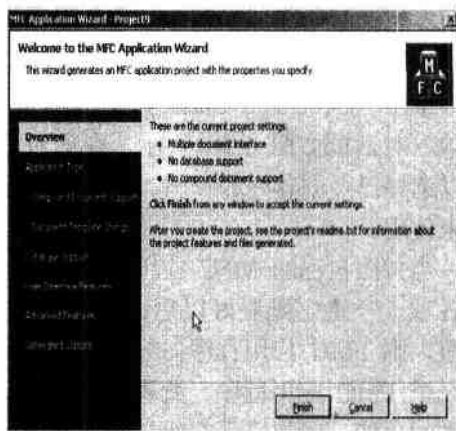


图 1.5 MFC 应用程序向导对话框

单击“Compound Document Support (复合文档支持)”选项，对话框如图 1.7 所示。在此选项中可以指定复合文档支持方式，即可选择 OLE 容器、服务器选项和活动文档。

单击“Database Support (数据库支持)”选项，对话框如图 1.8 所示。在此选项中可以指定包括各种数据源在内的数据库支持类型。

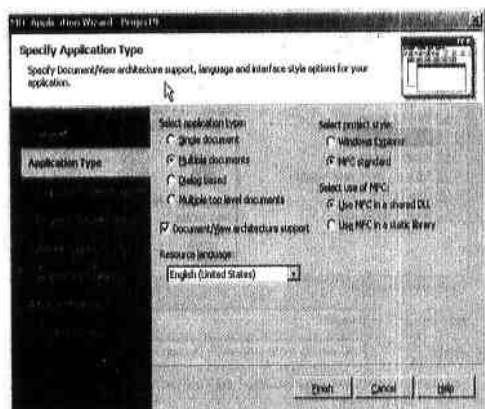


图 1.6 MFC 程序对话框 Application Type 选项

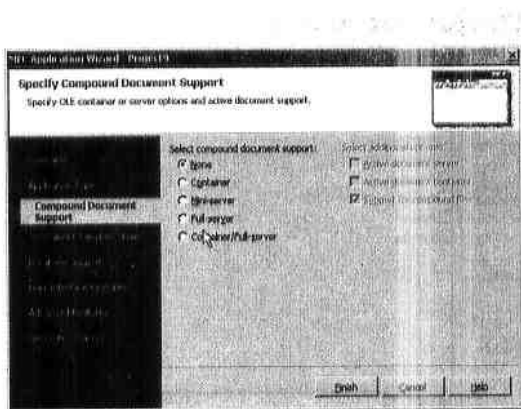


图 1.7 MFC 程序对话框 Compound Document Support 选项

单击“Advanced Features (高级特性)”选项,对话框如图 1.9 所示。在此对话框中可以为 MFC 应用程序设定其他一些高级属性。

单击“Generated Classes (已创建的类)”选项,对话框如图 1.10 所示。在此对话框中可以选择已经创建好的一些类供 MFC 应用程序来使用。

当按照自己的 MFC 应用程序的要求将上述各项一一选定配置好以后,单击“Finish”按钮,则 FMC 应用程序向导将自动为用户创建一个 FMC 应用程序,此程序已经具有一些基本的功能,用户只须在此基础上加以改动或添加一些功能即可。由以上可以看出, Visual C++.NET 强大的应用程序向导功能将为用户节约大量的精力与时间。同时可以从一个方面看出 Visual C++.NET 的无比优越性。

同样,在图 1.4 中,如果想要创建右面方框中的其他类型的程序,利用应用程序向导一样可以创建出优秀的其他应用程序。

● Log a Visual Studio.NET bug report (错误报告日志)

点击该项将会显示 Visual Studio 主页,可以在其中输入该产品的出错信息,Internet 将信息传到 Visual Studio.NET 7.0 的开发人员手中,以进一步改进优化 Visual Studio.NET 7.0,使之更加完善。

● Show the start page on next startup (在下一次启动时显示该页)

如果取消选择该复选框,则下次启动 Visual C++.NET 7.0 时,将不显示 Web 浏览器窗口。

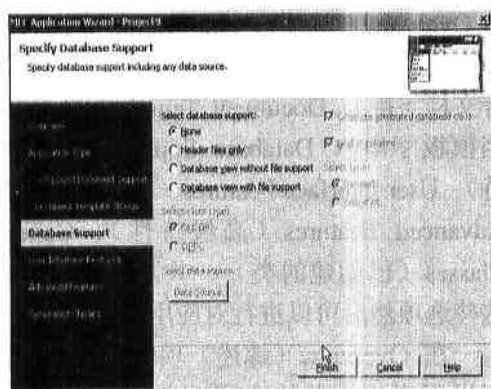


图 1.8 MFC 程序对话框 Database Support (数据库支持) 选项

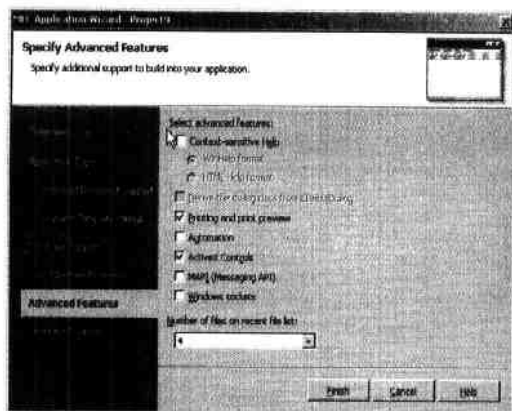


图 1.9 MFC 程序对话框 Advanced Features
(高级特性) 选项

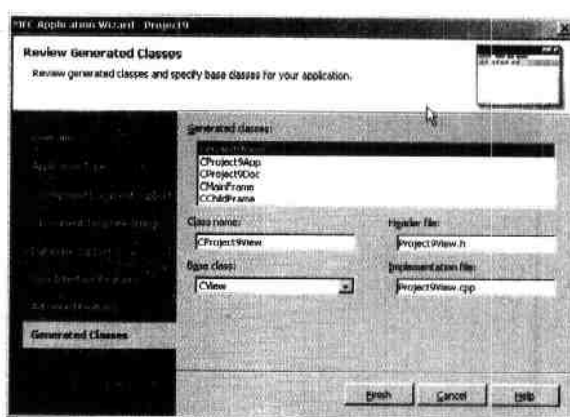


图 1.10 MFC 程序对话框 Generated Classes
(已创建的类) 选项

● What'New、OnLine Community、Headlines 及 Search OnLine

通过这些项能检索和查看关于 Visual C++.NET 7.0 的最新技术资料、事件和 MSDN 在线主题。

单击“What'New”项，将会出现如图 1.11 所示的页面。

在图 1.11 所示的页面中，单击超级连接 What'New in Visual Studio 7、What'New in Visual Studio Tools、What'New in Visual Basic、What'New in Visual C++、What'New in Visual FoxPro 7、Locating Readme Files 和 Automation object model、Visual Studio SDKs、Visual Studio Integrators program 和 Check for the latest Visual Studio updates 等等，将能在线阅读关于这些项的最新最详细最权威的资料。

单击“OnLine Community”项，能使项目开发者和程序员在线讨论一些关于运用 Visual C++.NET 7.0 开发中的一些问题。

单击“HeadLines”项，将会看到 Microsoft 公司的有关产品的 MSDN 的在线帮助。

单击 Search OnLine 项，将会弹出如图 1.12 的对话框。在图 1.12 所示的对话框中，在 MSDN Online Library 方框中输入所要查询的关键词，然后单击“go”超级连接，将会在对对话框中显示出相关内容。

● My Profile (个人风格设置)

单击“My Profile”选项，将会出现如图 1.13 所示的页面，在此页面上可以进行对 Visual Studio 界面风格的修改。可以修改 IDE 行为、键盘类型、窗口的布局等等参数。由于本书将介绍 Visual C++的开发，所以，将这些参数都设置为 Visual C++配置。

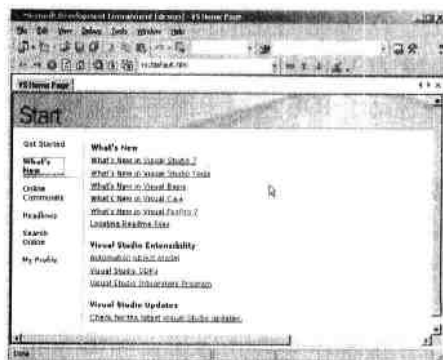


图 1.11 Visual Studio.Net 7.0 Web
浏览页的 What'New 选项



图 1.12 Visual C++.NET 7.0 Web 浏览页
的 Search onLine 选项

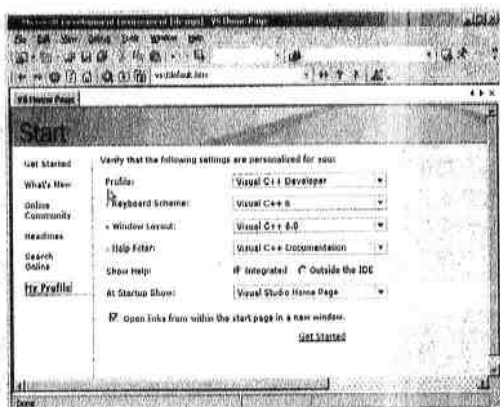


图 1.13 Visual C++.NET 7.0 Web 浏览页
的 My Profile 选项

1.2.2 Visual C++.NET 7.0 各窗口功能

Visual C++.NET 7.0 的窗口管理功能较以前的版本有了较大的改进，使得在屏幕上一次所能浏览的内容变得空前的多。图 1.14 为 Visual C++.NET 7.0 的窗口布局，其中包括了 IDE 环境中的主要窗口。

Visual C++.NET 7.0 的窗口管理功能主要包括有：标签化文档、自动隐藏、前进/后退 (Navigate Backward/Navigate Forward) 等等。

● 标签化文档

标签化文档功能将自动为 IDE 集成环境中的文档窗口设置标签。在 Visual C++.NET 7.0 集成环境中，默认情况下 Solution Explorer (方案管理器)、Class View

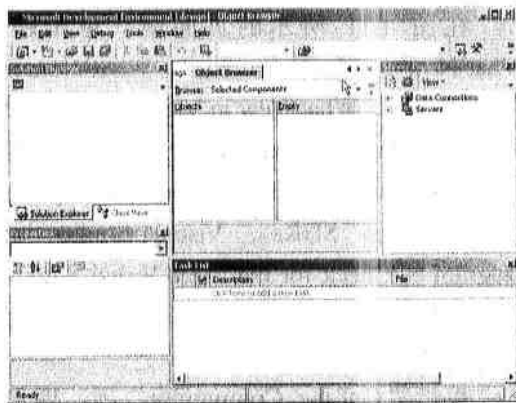


图 1.14 Visual Studio.NET 7.0 的窗口布局

(类窗口)、Resource View (资源窗口)、宏管理器窗口使用同一窗口 (见图 1.13 窗口的左上角)。通过切换底部的标签在此窗口中查看不同视图中的内容。当用户在编辑器或设计器中编辑多个文档时，这些窗口将全部显示于多文档界面 (MDI) 区域中。虽然一次只能显示一个文档中的内容，但是通过顶部与底部的标签可以方便的浏览其他文档中的内容。这较之以前的版本中只能通过 Windows 菜单中的窗口列表来选择，其方便性是显而易见的，尤其是在打开的文档数目较多时，尤为如此。

● 自动隐藏

在窗口标题栏处单击右键，在弹出的快捷键菜单中选择 “Auto Hide (自动隐藏)” 命令，即可实现自动隐藏。自动隐藏允许用户将工具窗口最小化。例如方案管理器和工具框，当执行自动隐藏功能时，将排列在 IDE 集成环境的边缘，从而可以节约宝贵的屏幕空间，提高

编辑器的可视面积。

- 前进/后退导航 (Navigate Backward/Navigate Forward)

此项功能可以使用户在打开的窗口或同一文件中的被选中的文本和光标位置中进行切换,这与 Web 浏览器中的“前进”和“后退”功能极为相似。例如,如果用户在第 16 行中编辑代码后移动到 189 行,这时可以通过 Navigate Backward 按钮来返回光标在上次 16 行中的位置。Navigate Backward 和 Navigate Forward 按钮都带有下拉列表,在其中显示了导航历史。这两个按钮位于 IDE 集成环境的标准工具栏中,同时用户也可以使用 Ctrl+(-) 和 Ctrl+Shift+(-) 来完成导航的功能。

在初步介绍了窗口管理的功能后,下面将逐一介绍各个窗口的功能。

- Solution Explorer (方案管理器窗口)

在 Visual C++.NET 7.0 的 IDE 集成环境的标准菜单栏的“View”(视图)菜单项中选择“Solution Explorer”选项,则会在主窗口的左上角弹出 Solution Explorer 窗口,如图 1.15 所示。

Project (工程) 是 Visual Studio 应用程序的构造块,这和 Visual Studio 6.0 以及以前的版本中的概念是一致的。如果将工程作为应用程序的构造块的话,那 Solution (方案) 就是应用程序的基础。与 Visual Studio 的早期版本一样,用户依然可以单独管理工程。但是,使用方案和方案管理器可以带来更多的方便,对于多工程项目尤为如此。

一个方案中可以包括多个工程,方案管理器可以使用户方便的组织和管理所要开发和设计的工程和文件,并且可以方便的配置应用程序和组件。图 1.16 所示为方案管理器中的工程的层次结构。方案中包括工程以及工程中的条目。此外,还包括两种可选文件—共享方案条目文件和杂项文件。

在方案管理器中创建和复制某个条目的应用将非常的简单,只需将要操作的条目拖动到另外的一个位置同时按下适当地组合键即可。

方案管理器中的拖放操作介绍如下:

- 拖放 其作用: 移动
- 【Shift】+拖放 其作用: 移动
- 【Ctrl+Shift】+拖放 其作用: 引用
- 【Ctrl】+拖放 其作用: 复制
- 【Ctrl+C】 其作用: 复制
- 【Ctrl+X】 其作用: 剪切
- 【Ctrl+V】 其作用: 粘贴

双击方案管理器中的文件,可以打开该文件。使用 Open With 对话框可以修改条目的默认编辑器。在方案管理器中不仅可以选择同一个工程的多个条目,也可以选择不同工程中的多个条目。当选择了多个条目时,属性窗体中将显示这些条目的共同属性,而此时可用的命令也为这些条目的交集。

方案管理器中的启动工程 (startup Project) 以粗体显示。当前焦点无论处于哪个工程或方案条目中,按下【F5】进行编译时,启动工程总是会被编译。如果存在多个启动工程,这必须选择 Build/Build Solution 命令来编译启动工程。