



010101011010101
01010110
10101101101110101
011010101010110
01010101101101101
1101110101011010101
10111010101101
0101101101110101
1011110001100110100101010
101011110100101010
0101011100100101010
0101011110001101001
11101101010111110
01110110101011110101011011
01000111011010101111101001101001
1000111011010101111010110110

计算机专业人员书库

Visual C++ .NET 实践与提高

——深入 Windows 编程

曾玉明 苏贵洋 马颖华 编著

计算机专业人员书库

Visual C++ .NET 实践与提高 ——深入 Windows 编程

曾玉明 苏贵洋 马颖华 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从实用的角度出发,用尽量简洁的方式讲解了 Visual C++ 在 Windows 编程中的各种应用,包含了入门者在 Windows 编程提高过程中所经常遇到的大多数问题,如面向对象的编程思想、Windows 消息机制、对话框与控件的编程。本书还详细和深入地讨论了动态链接库、ActiveX、多线程编程、进程间数据通信、COM 接口和 Shell 编程,最后,讨论了编程时经常遇到的问题:数据组织。在本书的讲解过程中,更多讲述的是一种思路,一些方法和一些常常困惑众人的难点,如此来启发读者的思想。

本书大部分例子的关键代码在书中都有详细的讲解,试图从各个方面给读者提供一些编程思想,所有例子的详细代码都可以在光盘中得到。

本书内容翔实、丰富,可供广大 VC 开发者、软件开发人员和计算机爱好者学习参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual C++ .NET 实践与提高——深入 Windows 编程/曾玉明,苏贵洋,马颖华编著. —北京:电子工业出版社, 2002.4

(计算机专业人员书库)

ISBN 7-5053-7410-9

I . V… II . ①曾… ②苏… ③马… III . C 语言—程序设计 IV . TP312

中国版本图书馆 CIP 数据核字(2001)第 097836 号

责任编辑:朱沐红

印 刷:北京天宇星印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:29.25 字数:740 千字 附光盘 1 张

版 次:2002 年 4 月第 1 版 2002 年 4 月第 1 次印刷

印 数:6 000 册 定价:48.00 元(含光盘)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077

前 言

Visual C++ 在 Windows 下的程序设计中有广阔的应用,但是,Visual C++ 却也一直困扰着很多入门和需要提高的人。这或许是因为其自身的结构复杂,但更多的或许是在实际学习过程中,学习者太过拘泥于学习它的编程细节,忽视了对应用的抽象以及对应用的总体框架的思考和知识的汇总与反思,所以水平也总是停滞不前。作者也曾一步一步地走过这些坎坷和困难,回头再看,深感自己走过的弯路太多。那么如何在基本入门后,迅速提升自己使用 Visual C++ 的编程水平呢?这就是本书想要具体讨论的内容。

本书的组织形式

本书从实用的角度出发,用尽量简洁的方式讲解了 Visual C++ 在 Windows 编程中的各种应用,其中包含了 Windows 编程入门者在提高过程中所经常遇到的大多数问题,例如面向对象的编程思想、Windows 消息机制、对话框与控件的编程。本书还较为详细和深入地讨论了动态链接库、ActiveX、多线程编程、进程间数据通信、COM 接口和 Shell 编程。在本书的最后,讨论了编程时经常遇到的问题:数据组织。在本书的讲解过程中,更多讲述的是一种思路,一些方法和一些常常困惑众人的难点,希望能够如此来启发读者的思想。

由于 Windows 提供的各种资源十分丰富,所以讲述过程中很难面面俱到,正如前文所讲,本书的讲述偏重于思想,希望通过这些讲解能起到举一反三的效果。本书提供了不少具体实例,大部分例子的关键代码在书中都有详细的讲解,这些例子的选择都是经过作者仔细筛选,试图从各个方面给读者提供一些编程思想;另外,例子也试图能对于其他 VC 书籍中未详细讲述到的 Windows 编程的一些技术作一些补充。本书提供的例子都是作者亲自编制、调试的,而且编制这些例子代码的思想也经过了详细的考虑。所有例子的详细代码都可以在光盘中得到。

但是作为技术的潮流和方向,本书对 Visual C++ .NET 做了介绍,并在各个章节中尽量以 Visual C++ .NET 为编程环境来讲解,但这丝毫不影响使用 Visual C++ 6 的读者阅读本书。

本书的开始(第 2 章)就讲述到了面向对象的思想,而且这个思想也贯彻本书的始终,这么设计的目的是希望读者在阅读本书后设计自己的工程应用时,能首先以面向对象的思想对工程项目进行抽象,从而简化和加速应用的构建。本书所提供的各个例子,也尽量贯彻面向对象这一编程思想。事实上,作者学习 Visual C++ 在 Windows 下程序设计时,就因为缺少面向对象的思想对工程应用的深层抽象,从而走了不少的弯路。

需要具备的知识

从本书的组织形式上,读者可以看出,本书是一本针对入门通往中高级程序员的书。所以,阅读本书前,希望读者已经具备一定的 Visual C++ 的基础,并且具有少许的 Windows 环境下应用程序的开发经验。如果对于面向对象思想有一定的认识或面向对象编程有一定的经验,那就更好了。

本书覆盖的内容之所以很广,是因为本书主要针对各个知识面中初学者容易遇到的问题而讲解的。但是,对每个知识面的基础知识,本书并没有太多的探究。读者如果需要对相应知

识进行详细研究,可以参考本书附录 B 推荐的相关书目。

为了达到最好的学习效果,建议读者能够边学边做,来随时验证自己所学的知识。同时,建议读者在阅读本书以及进行实践时,能多思考,并且进行各种实现方法、思想的比较,不要太拘泥于细节,当然也不要太拘泥于本书的一些实现方法。

本书的内容

本书分别进行了如下内容的讲解:

- 第 1 章 简单介绍了 Visual C++ .NET 的编程环境;
- 第 2 章 阐述了作者对面向对象编程思想的理解;
- 第 3 章 剖析 Windows 消息机制;
- 第 4 章 较为详细地探讨了对话框和控件编程;
- 第 5 章 深入动态链接库技术;
- 第 6 章 实际举例讲解 ActiveX;
- 第 7 章 讨论多线程编程;
- 第 8 章 用各种办法实现进程间数据通信;
- 第 9 章 从实例的角度探讨 COM,让 COM 不再那么神秘;
- 第 10 章 讲解 Shell 编程;
- 第 11 章 从编程角度构思如何进行数据组织。




关于 CD

本书的所有例子都可以在配套光盘中找到。考虑到还有很大一部分的读者在使用 Visual C++ 6.0 的编程环境,所以本书的例子整理分成了两类:一类是全部在 Visual C++ 6.0 中编译通过的(使用 DSW 项目文件);另一类是全部在 Visual C++ .NET 中编译通过的代码(使用 SLN 项目文件)。读者可以根据自己的编程环境需要,选择适合自己的代码。

由于环境改变带来的一些变化,第 4 章的例子 TS 和第 10 章的例子 Plus 都只能应用在 VC 6 平台下,VC 7 下不能运行,请读者注意。

本书的一些约定

本书书写过程中,对代码部分,都以阴影标出;对代码中的重点突出部分,以黑体标出;对一些需要注意的细节问题,分别以各种标号标记如下:

-  一些技巧
-  一些方法
-  注意事项

支持

Visual C++ 技术的博大精深,实在是让人叹为观止。作者虽然多年从事编程工作,积累了很多心得体会,但也只是窥见其中点滴。有限的水平,再加上时间紧迫,作者虽然非常努力和勤奋地工作,但是书中必然还存在着一些不足之处。如果您对本书有意见、问题或者看法,请通过以下电子邮件进行联系:vc-net@263.net。

致谢

在本书的编写过程中,得到了很多朋友和同行的帮助,作者在此表示深深的谢意。感谢我们身边的朋友,在写作最困难的时候,是他们给予我们巨大的克服困难的动力和热情。在本书的写作过程中,马艳华、尹中航、王朋、卢君明、牛霜霞等做了许多的文本输入、文字校对、格式排版、例子代码测试等工作,在此作者表示深深的感谢。最后,还要把本书献给我们的家人,他们在各方面提供的支持、帮助和便利,保证了本书的如期完成。

曾玉明 苏贵洋 马颖华

2001年10月

目 录

第 1 章 认识 Visual C++ .NET	(1)
1.1 什么是 .NET	(1)
1.2 Visual C++ .NET 的安装	(2)
1.3 开发界面	(4)
1.3.1 工作区窗口	(5)
1.3.2 项目管理	(6)
1.3.3 类管理	(6)
1.3.4 资源管理	(7)
1.3.5 属性窗口	(8)
1.4 转换旧的项目	(9)
1.5 应用程序向导	(10)
1.6 添加映射	(12)
1.7 调试工具	(13)
1.8 不同的 MSDN	(16)
1.9 其他新鲜事物	(18)
1.10 小结	(19)
第 2 章 面向对象的程序设计	(20)
2.1 根本思想和要解决的根本问题	(20)
2.2 代码重用	(24)
2.2.1 应用程序重用方式	(24)
2.2.2 模块重用方式	(25)
2.2.3 接口重用	(25)
2.2.4 源代码重用方式	(25)
2.2.5 类代码重用	(26)
2.3 接口设计	(31)
2.3.1 类封装	(31)
2.3.2 模板类	(35)
2.3.3 深入接口设计过程	(38)
2.4 小结	(48)
第 3 章 Windows 消息机制	(49)
3.1 概述	(49)
3.2 发送消息的方式	(50)
3.3 MFC 中的消息机制	(56)
3.3.1 MFC 对于消息的 Default 处理	(56)

3.3.2	MFC 中菜单、工具条、状态条的更新问题	(57)
3.4	MFC 中的消息反射机制	(62)
3.4.1	为什么需要消息反射	(62)
3.4.2	消息反射的响应	(62)
3.4.3	消息反射的响应顺序	(64)
3.4.4	消息反射的例子	(65)
3.5	遗留的问题	(68)
3.5.1	消息循环的遗留问题	(68)
3.5.2	处理 MDI 主框架客户区的消息	(73)
3.6	指针与句柄	(79)
3.7	小结	(83)
第 4 章	对话框与控件编程	(84)
4.1	概述	(84)
4.1.1	对话框	(84)
4.1.2	控件	(85)
4.2	定制 dialog 中的控件	(86)
4.3	给对话框添加工具栏、状态栏和菜单	(94)
4.3.1	在对话框中实现工具栏	(94)
4.3.2	在对话框中实现状态栏	(96)
4.3.3	在对话框中实现菜单	(97)
4.4	在 ToolBar 中创建控件	(98)
4.5	动态创建控件	(100)
4.6	启动时隐藏对话框窗口	(101)
4.7	子类化	(105)
4.8	自定义控件	(109)
4.9	控件的 Owner Draw	(116)
4.10	Common Dialog 扩展	(121)
4.11	伸缩对话框	(124)
4.12	小结	(128)
第 5 章	动态链接库	(129)
5.1	概述	(129)
5.1.1	认清 DLL 与 LIB 的真实面目	(129)
5.1.2	DLL 的两种动态链接方法	(130)
5.1.3	DLL 与 EXE 的关系	(132)
5.2	DLL 编程细节	(133)
5.2.1	DLL 编程必须注意的问题	(133)
5.2.2	DEF 文件	(134)
5.2.3	DLL 的载入过程	(139)
5.2.4	DLL 变量	(141)
5.2.5	DLL 的其他特性	(144)

5.3	几种常用的 DLL	(145)
5.3.1	最简单的 DLL	(145)
5.3.2	资源 DLL	(148)
5.4	特殊的 DLL 应用	(152)
5.4.1	控制面板应用	(152)
5.4.2	屏幕保护程序	(154)
5.5	模块的几种函数接口	(166)
5.5.1	回调函数	(167)
5.5.2	模块封装的发展趋势	(170)
5.6	小结	(171)
第 6 章	ActiveX 控件	(172)
6.1	概述	(172)
6.2	ActiveX 控件	(173)
6.2.1	ActiveX 控件与 Windows 控件	(173)
6.2.2	ActiveX 控件与普通 COM 组件	(174)
6.3	在 Visual C++ 中使用 ActiveX 控件	(177)
6.4	使用 MFC 开发 ActiveX 控件	(182)
6.4.1	接口设计	(183)
6.4.2	生成控件框架	(187)
6.5	使用 ActiveX 控件子类化 Windows 控件	(206)
6.6	ActiveX 控件中的消息反射机制	(207)
6.7	小结	(210)
第 7 章	多线程编程	(211)
7.1	概述	(211)
7.2	Visual C++ 中多线程库支持	(213)
7.2.1	MFC 提供的方法	(214)
7.2.2	Win32 提供的 API 函数	(218)
7.2.3	C 运行库提供的函数	(220)
7.3	线程操作	(223)
7.3.1	线程暂停计数器	(223)
7.3.2	线程的标识	(223)
7.3.3	线程的返回值	(224)
7.3.4	进程优先级	(224)
7.3.5	线程优先级	(227)
7.3.6	线程休眠	(230)
7.4	MFC 与多线程	(231)
7.5	多线程编程示例	(232)
7.6	线程同步	(238)
7.6.1	Wait 函数	(238)
7.6.2	Event(信号量)	(244)

7.6.3	Mutex(互斥量)	(246)
7.6.4	CriticalSection(关键代码)	(248)
7.6.5	Semaphore(信标)	(249)
7.7	同一进程创建多个界面线程	(251)
7.8	遗留的问题	(254)
7.9	小结	(261)
第 8 章	进程间数据通信	(263)
8.1	概述	(263)
8.2	使用文件进行通信	(263)
8.3	使用注册表进行通信	(264)
8.4	使用特定的消息进行通信	(266)
8.5	使用文件映射进行通信	(275)
8.6	使用剪贴板进行通信	(286)
8.7	使用读写进程内存方式进行通信	(297)
8.8	使用管道进行通信	(298)
8.9	使用共享数据段进行通信	(309)
8.10	小结	(318)
第 9 章	COM 接口	(319)
9.1	概述	(319)
9.2	COM 组件与 C++ 类对象	(325)
9.3	代理/残根 DLL	(329)
9.4	BSTR	(330)
9.5	几个常用的接口	(333)
9.5.1	IUnknown	(333)
9.5.2	IDispatch	(333)
9.5.3	IObjectWithSite	(335)
9.5.4	IConnectionPoint 与 IConnectionPointContainer	(336)
9.5.5	IoleCommandTarget	(340)
9.5.6	永久化接口	(344)
9.6	几个例子	(345)
9.6.1	浮动按钮例子	(345)
9.6.2	IE 插件例子	(355)
9.7	MFC 中接口的实现	(361)
9.8	小结	(368)
第 10 章	Shell 编程	(369)
10.1	概述	(369)
10.2	几个常用的 Shell 功能函数	(369)
10.2.1	ShellExecute 与 ShellExecuteEx	(369)
10.2.2	SHBrowseForFolder	(372)
10.2.3	SHFileOperation	(379)

10.2.4	SHGetFileInfo	(382)
10.2.5	SHGetSpecialFolderLocation 与 SHGetSpecialFolderPath	(386)
10.2.6	SHGetSettings	(389)
10.2.7	SHAutoComplete	(390)
10.2.8	Shell-NotifyIcon	(391)
10.3	文件拖放支持	(394)
10.4	Shell 常用接口	(403)
10.4.1	Shell 扩充使用的接口	(404)
10.4.2	Shell 名字空间扩展使用的接口	(407)
10.5	小结	(421)
第 11 章	数据组织	(422)
11.1	概述	(422)
11.2	MFC 提供的数据组织手段	(422)
11.2.1	CObject	(423)
11.2.2	MFC 类库提供的数据结构	(427)
11.3	窗口属性	(429)
11.4	线程内数据组织	(431)
11.5	Owner-Draw 的组织数据	(434)
11.6	Block Memory	(438)
11.7	小结	(445)
附录 A	Visual C++ 编程风格	(446)
A.1	变量风格	(446)
A.2	文件风格	(447)
A.2.1	文件生成	(447)
A.2.2	文件头部注释	(448)
A.2.3	空行约定	(448)
A.3	函数风格	(448)
A.3.1	函数注释	(448)
A.3.2	代码缩进	(449)
A.4	注释风格	(449)
A.5	小结	(449)
附录 B	推荐书目	(450)
	C++ 基础书籍及提高书籍	(450)
	MFC 编程书籍	(450)
	Windows 编程技术书籍	(451)
	面向对象编程书籍	(452)
	COM 书籍	(452)
附录 C	光盘使用说明	(453)

第1章 认识 Visual C++ .NET

程序员最关心的就是如何在最短的时间内开发出高效的应用程序，开发界因此盛行着 RAD (Rapid Application Development, 快速应用程序开发) 风潮。用过 Visual Studio, Delphi 等工具的程序员，会对这些开发工具的直接、便利印象深刻。应该承认，微软在这个范围内做的还是很不错的（对于作者本人，一个从 Visual C++ 1.5 一步步用到 Visual C++ .NET 的程序员来说，更是体会深刻）：开发界面越来越便利，功能越来越强大。对开发 Windows 应用程序的程序员来说，大多数人都或多或少对 Visual Studio 有一定的依赖性。

微软推出了 .NET 战略后，Visual C++ .NET 也随之应运而生。很自然的，可以在其中看到不少变化和进步。但是，即使是 Visual Studio 的熟练用户，刚刚接触 .NET 时，也会对其中的很多改变觉得束手无策。

本章将主要讲解这个新的开发环境的特点和一些细节要点。对于 Visual C++ 的老用户来说，掌握这些“新鲜”的步骤和方法是非常简单的；对于刚刚加入 Visual C++ 行列的程序员，本章也能使之快速入门，学会如何使用这样一个工具来进行应用程序的开发。

在 Visual Studio .NET 中，Visual C++，Visual BASIC 以及 C# 都集成在同一个开发环境下。掌握了其中任何一种语言的开发环境，都可以轻易地在同一环境下进行另外一门语言的开发。鉴于本书讨论的范围，本章的介绍将以 Visual C++ 为例进行。

不管开发环境如何便利，对应用程序的开发，更多的还是需要扎实的基本功和综合的知识。开发环境归根结底只是一个环境，除了节省用户的时间，提高效率以外，如何开发出优秀的应用程序，还是要靠用户自己的能力。不过不要着急，这些内容都是本书后面所希望带给读者的。第一步，还是先来认识一下 Visual C++ .NET。



本章并不准备对 Visual C++ .NET IDE 环境中的每一个菜单、按钮和其他小工具进行一一讲解，因为本书不想对用户手册进行翻译。下面的陈述，除了对 Visual C++ .NET 最基本的应用讲解之外，将力图突出讲解 .NET 环境下相对于 Visual Studio 之 VC++ 6 的改进和“新鲜”的地方，讲解开发环境中最令人感兴趣和最有用的地方。

1.1 什么是 .NET

微软为什么会推出 .NET 战略？要回答这个问题，先来看一下微软执行主席史蒂夫·鲍尔默在 Microsoft .NET 开发商大会上的演讲：“从微软的角度来讲，我们就是要建设一个平台来创建并且支持新一代的应用。……现有的基础设施很薄弱，基础设施需要以平台形式加以生存，这个平台应由多个软件开发商共享。每一个网站的开发商所建造的自己的平台是远远不够的。我们必须有一套通用系统服务来支持这样的操作。这种观点就说明，我们还有下一个层次的发展，也就是说因特网下一步的发展，它将使因特网的作用远远超越展

现一个网站。这一切使我们创建了 .NET。……需要有一个平台，也就是说在不同的环境下，创建一个统一的、持续的平台……”。

那么 .NET 到底能干什么呢？继续看他的解释：“我们提供的第一个主要产品可以展示很多这方面的技术，叫 Visual Studio .NET。这个产品还包括一些应用的框架，这些框架可以支持下一代 XML 的信息的编程和信息处理……”。

什么是 .NET？“我想从概念和具体内容两方面加以说明。 .NET 代表了一个集合、一个环境、一个可以作为平台支持下一代互联网的可编程结构。它是一个支持环境。就像比尔所作的类比那样， .NET 与 Windows 一样，也是一个位于客户机上的用户环境和基础用户服务集合，在服务器和网络中，它们还与可编程模型一致并且由可编程模型构建。因此，它既是一个用户体验，也是一个开发商体验的集合。”这就是鲍尔默对 .NET 的描述。

看了上面这么多，简而言之，可以归纳为 .NET 有三个优势：

- 与语言无关
- 与操作系统紧密结合
- 体系完备，易于使用

因此，从软件到服务的转变，这才是微软 .NET 计划的核心。作为 Microsoft .NET 战略开发工具的 Visual Studio .NET，自然也就被赋予了不同一般的重任。

1.2 Visual C++ .NET 的安装

并不是所有机器都适宜安装 Visual C++ .NET，微软公布的最低配置如下。

- CPU：PII450 以上机器（建议使用 PIII733 以上）
- 内存：96MB（建议 128MB）
- 空闲硬盘空间：不少于 3GB
- 显示：800×600，256 色（建议使用 16 位以上真彩色）
- 操作系统：Windows 2000，Windows NT 4，Windows Me，Windows 98

为了更加顺畅地运行 Visual C++ .NET，用户恐怕至少要使用 PIII600 以上的机器，256MB 内存好像也是必须的。如果还没有达到这个配置，就先去升级机器（尤其是内存）。不然听着哗啦哗啦作响的硬盘声音，像老牛一样运行的 Visual C++ .NET 也就失去了升级的意义。

在安装 Visual C++ .NET 之前，首先需要使用组件升级光盘 Windows Component Update（Windows 组件升级）。这个升级程序会首先检测操作系统是否符合 Visual C++ .NET 的安装要求，如果有什么必须的项目没有安装的话，该程序会给出列表和提示，然后再依次进行安装，如图 1-1 所示。升级过程中，可能要多次重启机器。对于使用 Windows NT 和 Windows 2000 的用户，如果对每次重启后的登录感到比较厌烦，则可以设置让安装程序自动记住登录密码，这样每次重新启动后，升级程序会自动登录系统继续进行安装，如图 1-2 所示。



升级程序包中并没有包含 IIS，但是安装 Visual C++ .NET 会建议用户安装。升级程序还会给出提示以及相关帮助，告诉用户如何安装 IIS 以及对服务器扩展进行配置。不过，用户还是可以跳过 IIS 的安装直接进行后面的升级。

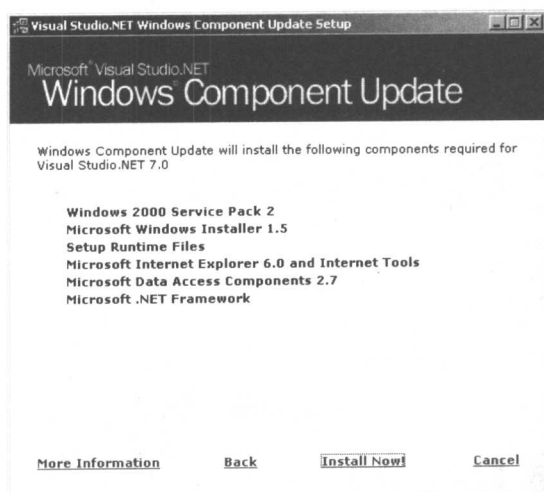


图 1-1 需要升级的组件列表

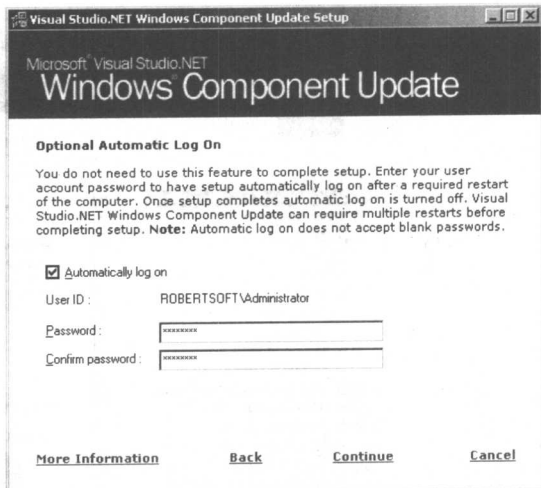


图 1-2 允许系统进行自动登录

如果用户使用的是微软的最新操作系统 Windows XP，则它和 Visual Studio .NET 的兼容性大大增加了。在安装时，用户只需要进行 3 个组件的更新安装，如图 1-3 所示，而且整个安装过程不需要重启。

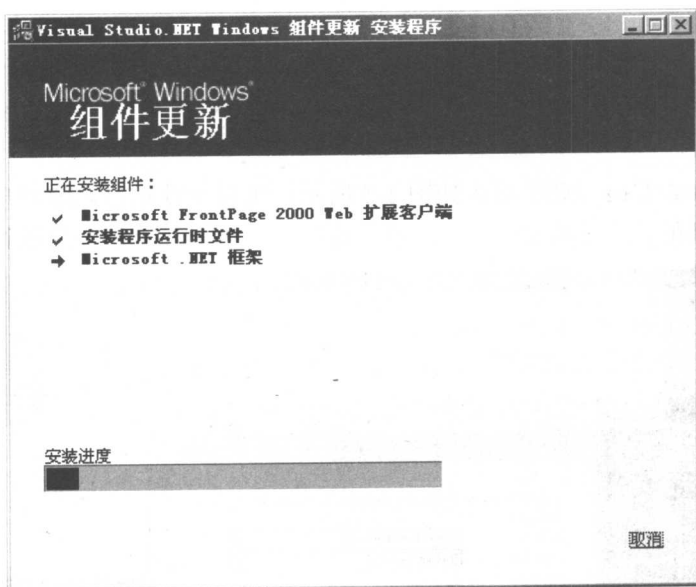


图 1-3 在 Windows XP 下需要更新的组件

升级全部完成后，才能开始真正的安装。因为 Visual Studio .NET 在同一 IDE 下集成了 Visual C++、Visual BASIC、Visual C# 三种开发语言，所以用户在安装时可以按照自己的喜好进行定制安装，如图 1-4 所示。

选择好需要的组件，经过一段漫长的拷贝，Visual C++ .NET 就在机器上安家了。

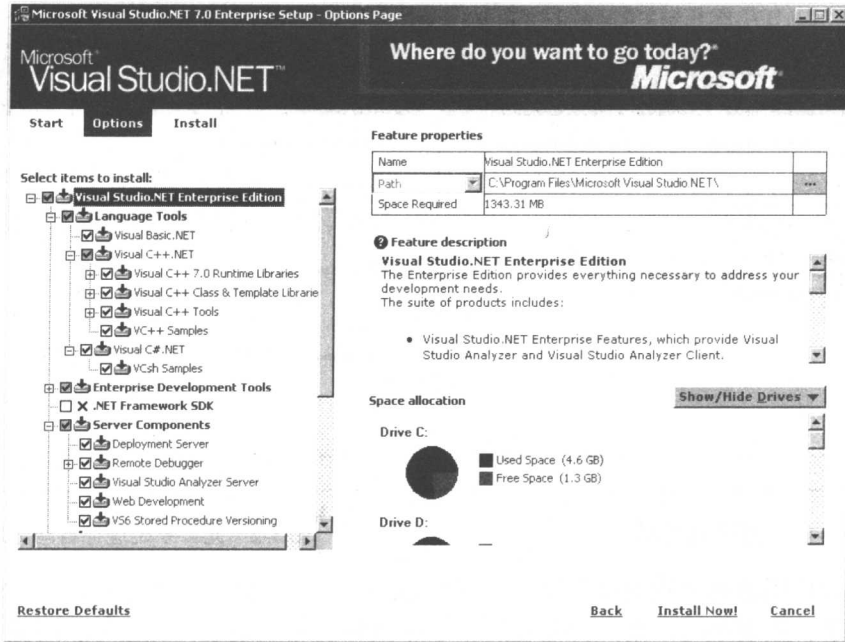


图 1-4 选择安装组件

1.3 开发界面

首次进入 Visual C++ .NET 后，如图 1-5 所示，可以个性化自己的用户界面，在这里修改键盘布局、窗口布局、帮助过滤等。当然，这些工作可以在以后随时进行配置修改。

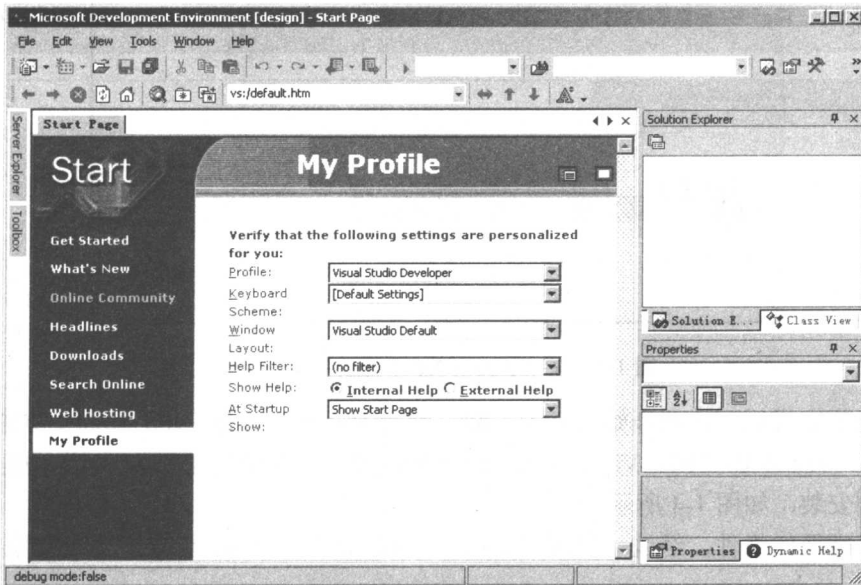


图 1-5 个性化自己的 Visual C++ .NET

1.3.1 工作区窗口

进入 Visual C++ .NET，让人不得不承认微软是一个界面高手：整个界面都非常淡雅清新，布局更加紧凑方便。用户可以灵活地定制工作区的各个子窗口，如图 1-6 所示。

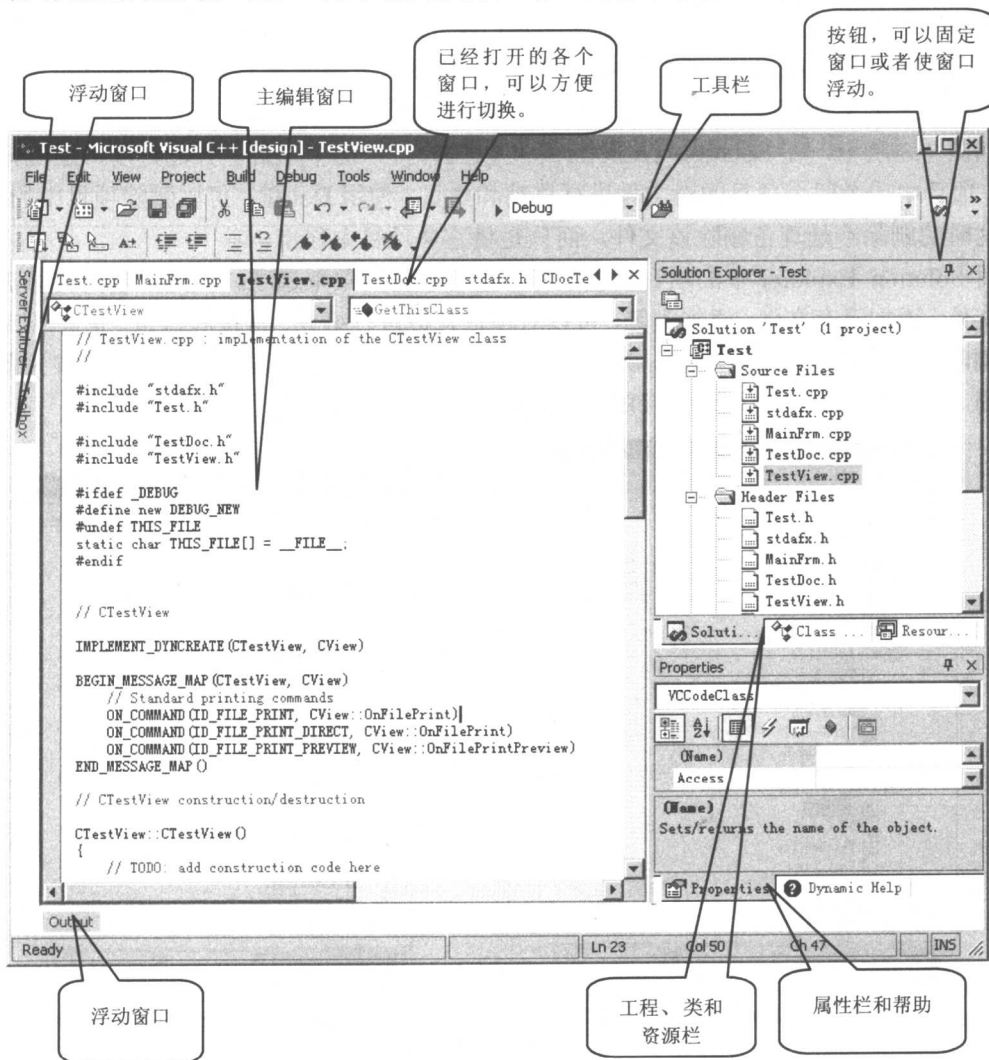


图 1-6 Visual C++ .NET IDE 的界面

每个子窗口都有一个“图钉”一样的按钮，使用它可以把子窗口“固定”或者“松开”，可以使得该子窗口固定在某一位置或者处于浮动状态。浮动时的窗口，在鼠标移过时，会自动恢复窗口原始大小；而鼠标移走或在平时状态下，则自动最小化依附在主窗口的边框上。浮动窗口带来的最大好处就是在编辑代码时，用户可以获得最大的编辑空间。

对每个子窗口，用户还可以拖住，然后移动到自己希望停放的位置，从而更加符合用户的个性化界面设置。

同样对于工具栏，通过右击鼠标，用户可以选择显示什么、不显示什么；也可以通过

菜单“Tools”中的“Customize”拖拽来自定义工具栏上的每一项。

1.3.2 项目管理

在 Visual C++ .NET 中，项目文件不再是以前的以 dsw 为后缀的文件，后缀名改为了 sln（solution，解决方案）。在主窗口的各个子窗口中，就有一个“Solution Explorer”，如图 1-7 所示。它很像以前 Visual Studio C++ 版本的 FileView（文件浏览），在这里用户不仅可以看到项目中的全部文件列表，而且可以对文件和项目进行强大的管理。

如图 1-8 所示，在 Solution Explorer 中可以同时打开两个项目，并允许用户通过拖拽文件，在项目或者两个项目间对文件进行移动和拷贝。使用 Del 键，可以删除项目中的文件。但是这时的删除不是真正删除该文件，而只是将它从项目中移出。

对 Solution Explorer 中的项目右击鼠标，会有丰富的菜单选项。例如，它支持添加新的项目文件，添加已有文件，添加类等，可以对项目直接进行编译、链接和调试。选择“Clean”，可以删除项目编译链接时产生的.pch，.pdb，.obj 等临时文件，这样往往可以节省大量的磁盘空间。不过，如果再次编译链接时，将需要花费更长的时间。

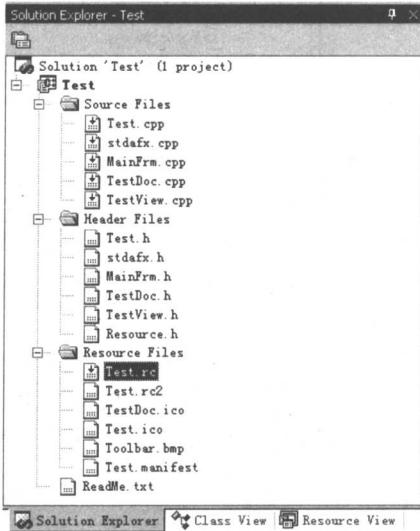


图 1-7 Solution Explorer 子窗口



图 1-8 在 Solution Explorer 中同时打开两个项目

1.3.3 类管理

在类管理 Class View 里，用户可以用树形结构来了解类的组织，包括成员函数和成员变量。对于公有成员（无标记）、保护成员（钥匙）和私有成员（小锁），将分别以不同的符号表示。

和以往 Visual C++ 版本不同的是，在 Class View 的每一个类中，都有其基类信息（Bases and Interfaces）和消息映射（Message Maps）。对于整个 Project，还在 Class View 里显示了宏和常量（Macros and Constants）。如图 1-9 所示。

在 Class View 里，通过双击相应类名、变量名等，可以迅速定位到相应函数体、变量定义的位置；通过右击菜单，可以为项目添加类和资源，为类添加函数和成员变量。