



Microsoft®



计算机教育图书研究室
Computer Education Books

总策划

Visual Studio.net

Visual C++

Completely Windows tutorial

全方位教程

.NET



黄承安 张长春 主编

● 本书内容

- Visual C++.NET 概述
- C++ 语言简介
- VC++ 与面向对象程序设计
- Windows 程序设计
- MFC 程序设计
- 文档/视图模型
- 多线程编程
- 动态链接库编程
- 图形图像编程

航空工业出版社

Visual C++ .NET全方位教程

主 编 黄承安 张长春
编 委 吴宝山 任立功
 王文增 董金波



航空工业出版社

155/02/06

内 容 提 要

本书深入浅出地介绍了 Visual C++ .NET 程序开发的基础知识和应用技术。第一章首先介绍了微软的 .NET 技术和 Visual Studio .NET 开发环境的使用以及定制技巧。从第二章开始介绍了 Visual C++ .NET 的语言基础编程知识, 其中还涉及用托管代码开发 .NET 框架应用程序的知识。接着介绍了 C++ 语言在面向对象程序设计中的应用。从第四章开始介绍了如何使用 Visual C++ .NET 编写 Windows 应用程序, 包括: Visual C++ .NET 的库支持——MFC, 最基本的程序结构——文档视图结构, 用户界面的最新支持——各种类型的控件, GDI+ 和图像处理、多线程、数据库、动态链接库的内容, 以及用 MFC、ATL 和 ATL Server 开发 COM+ (包括 ActiveX) 的方法。

本书内容丰富、思路清晰、图表丰富、内容全面、方法实用, 适合于具有一定编程基础的开发人员学习使用或作为教学培训的参考用书使用。

图书在版编目 (CIP) 数据

Visual C++ .NET 全方位教程 / 黄承安等主编.

—北京: 航空工业出版社, 2003.11

ISBN 7-80183-221-3

I. V… II. 黄… III. C 语言—程序设计

IV. TP321

中国版本图书馆 CIP 数据核字 (2003) 第 083239 号

航空工业出版社出版发行

(北京市安定门外小关东里 14 号 100029)

北京市燕山印刷厂印刷

全国各地新华书店经售

2003 年 11 月第 1 版

2003 年 11 月第 1 次印刷

开本: 787×1092 1/16

印张: 25.25

字数: 408 千字

印数: 1-6000

定价: 32.00 元

本社图书如有缺页、倒页、脱页、残页等情况, 请与本社发行部联系调换。联系电话: 010-65934239 或 84917422

前 言

Visual Studio.NET 系统的推出是这两年来最激动人心的事情之一。Microsoft 公司对 Visual Studio 工具进行了深层次的开发，推出了全新的开发平台 Visual Studio.NET，其中包括了 Visual C++、Visual Basic 和 C# 等多种开发工具。作为微软系列开发工具的老大，Visual C++.NET 也有了脱胎换骨般的变化。

“精通 C++程序员才是真正的程序员”是软件业中最流行的一句话。C++作为广大程序员公认的最美妙的语言，既具有高级语言简便、易用的特性，又具有低级语言功能、强大的特性。而 VC++无疑是 C++各种版本语言中最优秀的，它是微软为 Windows 系统量身定做的超级开发语言。在强大的 Windows 系列操作系统的支持下，VC++已经成为用户最多、功能最强的程序开发语言。学会了 VC++，你就能窥见 Windows 操作系统的内部结构；你就能够欣赏精致的 Windows 消息循环机制；你就能灵活运用 Windows 提供的各种各样的强大工具。到那时，你才能拍着胸脯说“我学会了 Windows 编程”。所以学习、掌握甚至精通 VC++是每个 Windows 程序员的目标。

对于程序爱好者和程序开发人员来讲，如何学习和掌握 Visual C++.NET 成了一件相当迫切的事情。本书的目的就是帮助广大读者尽快掌握 VC++.NET 的程序设计方法。

总的来说，本书具有以下一些特点：

第一、内容平实，能全面反映 Visual C++.NET 的最新特性，又不艰深晦涩。本书主要介绍最基本的概念和操作，而对过于深入的内容则没有浪费过多的笔墨。

第二、内容全面，难度适中，使读者在阅读过程中不会感觉突兀，能够自然地了解并掌握 VC++.NET 的各种知识。

第三、实例丰富、实用。本书的实例步骤清晰、代码明确，具有很重要的实用价值，对实际开发应用有很大的参考作用。书中注重程序的简洁性、实用性和通用性，摒弃了以往一些书籍过于注重所谓“技巧新颖”而编制特殊程序的做法。因为这样的程序通用性、可读性都很差。

第四、语言简洁明了，全书没有晦涩的字句。内容全面，列表、示意图、代码丰富，便于理解和查阅。

本书以通俗易懂的语言对 VC++的本质和原理进行了阐述，既针对向初、中级用户，又兼顾高级用户，完全满足了初、中级用户的学习应用，以及高级用户进阶的需要，可以说是深入浅出、内容全面、知识新颖、难度适中、方法实用，适合于具有 C/C++知识的开发人员学习使用或作为教学培训参考资料使用。本书由黄承安、张长春主编，同时参与本书编写和制作的还有：吴宝山、任立功、王文增、董金波等。由于编者水平有限，加之时间仓促，书中难免有疏漏和不足之处，恳请广大读者批评指正。

<http://www.china-ebooks.com>

编 者

2003年8月



目 录

第 1 章 Visual C++ .NET 概述 1

1.1 微软的 .NET 战略 1

1.1.1 .NET 的核心组件 1

1.1.2 .NET 的重大意义 1

1.1.3 .NET 与 Web 服务 2

1.1.4 .NET 体验 2

1.1.5 Visual Studio .NET 3

1.2 Visual C++ .NET 的新特性 4

1.2.1 C++ 属性 5

1.2.2 通用语言运行时 5

1.2.3 受控代码 (C++ 托管扩展) 5

1.2.4 基础类库与命名空间 6

1.2.5 .NET 框架类库 6

1.2.6 ATL Server 6

1.2.7 类库的升级 6

1.2.8 编辑器特性的改进 7

1.2.9 调试器特性的改进 7

1.2.10 编译器的新特性 8

1.3 Visual Studio 的集成开发环境 9

1.3.1 Visual Studio 起始页 10

1.3.2 Visual Studio .NET 的配置 11

1.3.3 创建新项目 12

1.3.4 Visual Studio 集成开发环境
(IDE) 主要窗口功能介绍 13

1.3.5 窗口模式与界面模式 19

1.3.6 文本编辑器的使用 20

1.3.7 控件编辑 23

本章小结 25

第 2 章 C++ 语言简介 26

2.1 C++ 语言概述 26

2.1.1 C++ 的发展历程 26

2.1.2 C++ 与面向对象程序设计 26

2.1.3 C++ 的程序结构 28

2.1.4 在 Visual Studio .NET 中设计
C++ 程序 28

2.2 C++ 的数据类型 30

2.2.1 常量及其定义 30

2.2.2 变量 31

2.2.3 保留字 31

2.2.4 指针 31

2.2.5 基本数据类型 32

2.2.6 复合数据类型 33

2.3 C++ 的流程控制 36

2.3.1 条件语句 36

2.3.2 循环语句 38

2.3.3 转移语句 39

2.4 C++ 的函数 40

2.5 .NET 对 C++ 语言的扩展 40

本章小结 42

第 3 章 VC++ 与面向对象 程序设计 43

3.1 面向对象概述 43

3.2 类 44

3.2.1 什么是类 44

3.2.2 类的定义格式 44

3.3 对 象 46

3.3.1 对象的定义格式 46

3.3.2 对象成员的表示方法 46

3.3.3 指向对象的指针 47

3.4 类与对象的使用 47

3.4.1 对象初始化 47

3.4.2 成员函数的特性 50

3.4.3 常类型 (const) 51

3.4.4 静态成员 (static) 51

3.4.5 类的作用域 52

3.4.6	友元 (friend)	52
3.4.7	new 和 delete 的应用	53
3.4.8	局部类和嵌套类	54
3.4.9	对象的生存周期	55
3.4.10	对象指针和对象引用	55
3.4.11	类型转换	57
3.5	继承与派生	57
3.5.1	基类和派生类	58
3.5.2	单继承	59
3.5.3	多继承	60
3.5.4	虚基类	61
3.6	多态和虚函数	62
3.6.1	重载	62
3.6.2	运算符重载	64
3.6.3	虚函数	64
3.6.4	纯虚函数和抽象类	65
3.6.5	虚析构函数	66
	本章小结	66

第 4 章 Windows 程序设计

4.1	Windows 基础	67
4.1.1	Windows 的介绍	67
4.1.2	Windows 用户界面的构件	67
4.2	Windows 程序设计	69
4.2.1	Windows 程序设计基础	69
4.2.2	事件驱动编程方法	71
4.2.3	Windows 窗口编程	71
4.2.4	匈牙利表示法	73
4.2.5	Windows 程序流程分析	74
4.3	消息响应机制	75
4.3.1	事件驱动与消息	75
4.3.2	消息的处理机制	77
4.3.3	消息的种类	77
4.3.4	消息的发送和接收	81
4.3.5	消息映射	83
4.3.6	自定义消息	83
4.4	VC++.NET 与 Windows 程序设计	84
4.4.1	WinMain 函数	84

4.4.2	应用程序类	86
4.4.3	Run 成员函数	87
4.4.4	关闭应用程序	87
4.4.5	理解 VC 工程	88
4.4.6	一个简单实例	89

本章小结

第 5 章 MFC 程序设计

5.1	微软基础类 (MFC) 库	94
5.2	MFC 中的常用类	95
5.2.1	基本类	95
5.2.2	应用类	96
5.2.3	其他用户界面类	98
5.2.4	绘图类	99
5.2.5	其他 MFC 类	100
5.3	类的消息机制	102
5.4	MFC 的程序流程	104
5.5	MFC 的程序实例	104
	本章小结	111

第 6 章 文档/视图模型

6.1	文档视图模型概念	112
6.1.1	文档/视图结构	112
6.1.2	SDI 与 MDI 模型	113
6.1.3	文档视图体系的核心类	114
6.1.4	文档视图相互作用	124
6.1.5	使用文档/视图结构的意义	125
6.1.6	文档/视图结构中主要类的作用	126
6.1.7	访问当前活动视图和文档	130
6.2	文档/视图结构程序实例	131
6.2.1	创建应用程序的骨架	131
6.2.2	声明文档的数据对象	131
6.2.3	初始化文档类的数据成员	131
6.2.4	清理文档类的数据成员	132
6.2.5	读写文档-串行化	133
6.2.6	文档显示的实现	136
6.2.7	文档编辑功能的实现	136
6.3	多文档界面窗口	140
6.3.1	MDI 与 SDI 的区别	141

6.3.2 MDI 应用的文档的创建与显示	141	7.6 在非对话框窗口中使用控件	207
6.3.3 多文档视图程序的架构	141	7.6.1 在表单视图中使用控件	207
本章小结	145	7.6.2 在工具条和状态栏中使用控件	207
第 7 章 对话框和控件	146	7.6.3 在非对话框窗口中使用控件 的实例	208
7.1 对话框的基本概念	146	7.7 控件的运用实例	213
7.1.1 模态对话框与非模态对话框	146	7.7.1 控件面板的创建	213
7.1.2 MFC 对对话框的封装	147	7.7.2 成员变量的设定	214
7.1.3 CDialog 对话框类	148	7.7.3 程序功能的实现	215
7.1.4 CWnd 和 CWinApp 类对于 对话框的管理	151	7.8 对话框与控件运用的 综合实例	218
7.1.5 .NET 对对话框属性的处理	152	7.8.1 对话框创建和显示的一般方法	218
7.2 系统预制对话框与标签式 对话框	153	7.8.2 创建对话框类及处理消息	219
7.2.1 系统预制对话框	153	7.8.3 对话框模板的设计	219
7.2.2 标签式对话框	158	7.8.4 对话框类的设计	221
7.3 Windows 标准控件	161	7.8.5 对话框的初始化	223
7.3.1 静态控件	162	7.8.6 对话框的数据交换机制	223
7.3.2 编辑控件	164	7.8.7 对话框的运行机制	224
7.3.3 按钮控件	169	7.8.8 处理控件通知消息	225
7.3.4 列表框控件	172	7.8.9 非模态对话框的运用	228
7.3.5 组合框控件	177	本章小结	230
7.3.6 滚动条控件	181	第 8 章 多线程编程	231
7.4 高级控件	183	8.1 线程概述	231
7.4.1 高级控件的通知消息	184	8.1.1 进程与线程	231
7.4.2 旋转按钮控件	186	8.1.2 深入理解线程	231
7.4.3 滑尺控件	189	8.2 线程的编程技术	232
7.4.4 进度条控件	191	8.2.1 编写线程函数	232
7.4.5 树形视图控件	192	8.2.2 创建一个线程	233
7.4.6 列表视图控件	197	8.2.3 终止线程	233
7.5 控件通用技术	203	8.2.4 线程的挂起与恢复	234
7.5.1 所有的控件都是窗口	203	8.2.5 设定优先级	235
7.5.2 控件的创建方法	203	8.2.6 线程的同步	235
7.5.3 访问控件的方法	204	8.2.7 线程间通信	238
7.5.4 控件及控件对象的删除	205	8.2.8 线程对共享资源的访问	240
7.5.5 控件通知消息	205	8.3 基于 MFC 的多线程编程	241
7.5.6 管理对话框控件的 CWnd 成员函数	205	8.3.1 工作者线程	242
7.5.7 标准控件的控件通知消息	206	8.3.2 用户接口线程编程	242
		8.3.3 同步对象的使用	243



8.4 多线程实例.....	243	10.1.3 不同类别的动态链接库.....	296
8.4.1 程序框架的创建.....	244	10.1.4 动态链接库的选用.....	296
8.4.2 创建 CUIThread 类.....	244	10.2 普通的 DLL 程序.....	296
8.4.3 添加线程对话框.....	245	10.2.1 动态链接库(DLL)结构.....	297
8.4.4 创建工作程序线程.....	248	10.2.2 编写 DLL 的主要步骤.....	298
8.4.5 测试线程应用程序.....	251	10.2.3 DLL 中函数的导出方法.....	298
本章小结.....	252	10.2.4 使用模块定义文件导出} DLL 中的函数.....	299
第 9 章 组件技术与 ATL	253	10.2.5 使用_declspec(dllexport)导出 DLL 中的函数.....	301
9.1 组件技术概述.....	253	10.2.6 DLL 的加载与调用方式.....	302
9.2 ActiveX 组件技术基本概念.....	254	10.2.7 使用隐含链接.....	303
9.2.1 ActiveX 的由来.....	254	10.2.8 使用显式链接.....	305
9.2.2 组件的标准化.....	255	10.3 使用 MFC 扩展动态链接库.....	307
9.2.3 开发 ActiveX 控件的方法.....	255	10.3.1 DLL 与 MFC 扩展 DLL 的 不同之处.....	307
9.2.4 ActiveX 的组件类型.....	256	10.3.2 MFC 扩展 DLL 的示例程序.....	307
9.3 ActiveX 组件程序.....	256	10.3.3 DLL 调用时的注意事项.....	310
9.3.1 容器应用程序的创建.....	256	本章小结.....	311
9.3.2 ActiveX 控件的创建.....	261	第 11 章 图形图像编程	312
9.3.3 添加 ActiveX 控件的方法.....	262	11.1 图形图像的基本概念.....	312
9.3.4 服务器应用程序的建立.....	263	11.2 图形设备接口.....	313
9.3.5 利用 ATL 来创建 ActiveX 控件.....	268	11.2.1 设备描述表.....	313
9.3.6 ActiveX 控件的注册.....	270	11.2.2 窗口视和视口视.....	315
9.4 COM 系列技术简介.....	271	11.2.3 坐标系统与映射模式.....	316
9.4.1 Object Viewer 工具.....	272	11.2.4 绘图工具.....	317
9.4.2 DCOM.....	272	11.2.5 绘图函数.....	317
9.4.3 COM+.....	273	11.3 Windows 位图.....	318
9.4.4 .NET 与 COM/COM+的关系.....	275	11.3.1 设备无关位图.....	318
9.4.5 .NET 对 COM+的支持.....	276	11.3.2 设备相关位图.....	320
9.5 ATL 技术.....	277	11.3.3 抖动和非抖动颜色.....	324
9.5.1 什么是 ATL.....	277	11.3.4 位图编程实例.....	325
9.5.2 ATL 基本技术.....	278	11.4 基于 .NET 框架的 GDI+ 编程.....	331
9.5.3 使用 ATL 编写组件程序.....	279	11.4.1 GDI+.....	332
9.5.4 ATL Server.....	289	11.4.2 GDI+的新功能.....	332
本章小结.....	293	11.4.3 GDI+命名空间.....	332
第 10 章 动态链接库编程	294	11.4.4 图形类.....	335
10.1 动态链接库概述.....	294		
10.1.1 动态链接库的作用.....	294		
10.1.2 动态链接库的优点.....	295		



11.4.5 常见图形对象	335	12.2.4 CRecordset 类	361
11.4.6 为命名空间加引用	338	12.2.5 CRecordView 类	367
11.4.7 GDI+图形编程实例	338	12.3 数据库应用的实例	368
11.5 图像编程综合实例	340	12.3.1 注册数据源	368
11.5.1 编写 DIB 类	340	12.3.2 创建应用程序	370
11.5.2 建构应用程序	345	12.3.3 添加资源和代码	371
本章小结	353	12.4 DAO 编程技术	377
第 12 章 数据库编程	355	12.4.1 什么是 DAO	377
12.1 数据库基本概念	355	12.4.2 DAO 与 ODBC 的异同	377
12.1.1 数据库、DBMS 和 SQL	355	12.4.3 DAO 技术的特点	378
12.1.2 数据库访问技术	356	12.4.4 使用 DAO 设计数据库程序	379
12.2 ODBC 编程技术	357	12.5 自动注册 DSN 和创建表	381
12.2.1 ODBC 基本概念	357	12.5.1 自动注册 DSN	381
12.2.2 MFC 的主要 ODBC 类概述	359	12.5.2 用 ODBC 创建表	382
12.2.3 CDatabase 类	360	12.5.3 用 DAO 创建表	383
		本章小结	384

第 1 章 Visual C++.NET 概述

什么是.NET? 简单地说, Microsoft.NET 是微软以服务的方式递交软件的一种策略。

本章将详细介绍 Visual C++.NET 给程序设计带来的重大意义和它的一些新特性以及它独特的开发环境。

1.1 微软的.NET 战略

Microsoft.NET 是微软公司的新战略, 所有微软产品都将围绕这个战略开发。Microsoft.NET 能使用户通过 Web 与众多的智能设备交互, 同时确保用户而不是应用程序控制这个交互。Microsoft.NET 能使得用户对应用程序、服务、个性化设备的体验简单、一致而安全。

Microsoft.NET 的策略是将互联网本身作为构建新一代操作系统的基础, 对互联网和操作系统的设计思想进行合理的延伸。这样, 开发人员必将创建出摆脱设备硬件束缚的应用程序, 以便轻松实现互联网连接。Microsoft.NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑。

1.1.1 .NET 的核心组件

.NET 的核心组件包括:

- 一组用于创建互联网操作系统的组件块。
- 构建和管理新一代服务的基本结构和工具, 包括: Visual Studio.NET、.NET 企业服务器、.NET 框架和 Windows.NET。
- 能够启用新型智能互联网设备的.NET 设备软件。
- .NET 用户体验。

1.1.2 .NET 的重大意义

.NET 对最终用户来说非常重要, 因为计算机的功能将会得到大幅度提升, 同时计算机操作也会变得非常简单。特别是用户将完全摆脱人为的硬件束缚: 用户可以自由地冲浪于互联网的多维时空, 而不是束缚在电脑的方寸空间。用户可以通过任何桌面系统、任何便携式电脑、任何移动电话或 PDA 进行访问, 并可对其进行跨应用程序的集成。

.NET 可使用户轻松地进行互联网的连接, 并轻松完成那些在当今看来十分费时、费力的事。通过将多项安全数据流合并到单一的用户界面, .NET 架构将用户从充斥于当今网页中的数据陷阱的束缚中解脱出来。用户可以自由访问、自由查看、自由使用他们的数据。

.NET 将会改变开发人员开发应用程序的方式, 使得开发人员能创建出全新的各种应用程序。新的开发方式的核心将是 Web 服务。Web 服务是基于一种简单对象访问协议(SOAP)

的，它能在互联网上展现其功能性的、公开的服务。

同时，.NET 作为微软目前的核心战略，其意义并非只限于技术范畴。微软在当初面临有可能被拆分的局面时，.NET 就是应对之策，它将担负起用技术形式将拆分后的若干小微软捆绑成一个整体的重任。无论是应用软件还是开发工具，都将基于 .NET 体系实现，通过 .NET 与操作系统的紧密结合就把应用软件和开发工具的核心与操作系统捆绑在一起了。这样每一个领域里的竞争对手将要面对的并不是一个小微软，而是由若干小微软组成的混合舰队。.NET 的核心竞争力就是与操作系统的紧密结合，而 .NET 战略的主要目标就是把微软的核心竞争力由操作系统转移到 .NET 上，这也就意味着 .NET 完全可能被移植到其他操作系统上，通过这种方式，微软的软件可以覆盖到软件业的所有方向。

1.1.3 .NET 与 Web 服务

在过去，开发人员通过集成本地系统服务来构建应用程序。在这种模型下，开发人员可以访问丰富的开发资源并能严格控制应用程序的行为。

如今，开发人员已在很大程度上摆脱了这种模型的束缚，致力于构建具有复杂结构的多层化系统，这种系统能将网络上众多的应用程序一并进行集成，大大提升了应用程序的价值。这样，开发人员便可把精力集中在充分挖掘软件的商业价值上，而不是构建基本结构上。可喜的局面将应运而生：软件投放市场的时间将大大缩短、开发人员的编程效率明显提高。

利用 XML 能够创建出可供任何人从任何地方使用的、功能非常强大的应用程序。它极大的拓展了应用程序的功能，并实现了软件的动态提供。在这种情况下，软件已不完全指那些从光盘进行安装的程序，而是演变成了一种服务，它类似于按收看次数进行收费的电视，人们可通过通信媒体订购服务。

Web 服务是一种应用程序，它可以通过编程并使用标准的 Internet 协议将功能展示在互联网和企业内部网上。还可将 Web 服务视为 Web 上的组件编程。从理论上讲，开发人员可通过调用 Web 应用编程接口将 Web 服务集成到应用程序中。其调用方法与调用本地服务类似，不同的是 Web API 调用可通过互联网发送给位于远程系统中的某一服务。例如，Microsoft Passport 服务使得开发人员能够对应用程序进行认证。通过对 Passport 服务编程，开发人员可以充分利用 Passport 的基本结构，通过运行 Passport 来维护用户数据库，以确保其正常运行、定期备份等。

.NET 正是根据这种 Web 服务原则而创建的，微软目前正在着手提供这个基本结构，以便通过 .NET 平台的每一部分来实现这种新型的 Web 服务。而 Visual Studio.NET、.NET 框架、Windows.NET 和 .NET 企业服务器正是为了进行基于 Web 服务模型的应用程序开发而量身定做的新一代开发工具和基本架构。即将到来的 .NET 用户体验将人们彻底攻克这一难题划上一个圆满的句号，使人们能够充分利用 Web 服务模型如愿以偿的开发出新一代的应用程序。

1.1.4 .NET 体验

.NET 体验是最终用户与 XML、Web 服务交互的一种手段。一次 .NET 体验类似于用户

与现行单个的应用程序交互。假设你想打开空调，而你正在车上，这时你就可以进行一次.NET 体验，即通过电子设备控制你的家用电器。在这个例子中.NET 的独特性在于：

- 每一个设备都为.NET 体验产生一个不同的用户界面。掌上电脑可能会显示一个地形图，智能手机会给你语音提示，多媒体信息亭可能会有一个触摸屏。你可以用.NET 体验完成同一个任务，但是你看到的是适合于不同设备的截然不同的界面。

- .NET 体验是与位置无关的。利用定位安装和基于 Web 应用程序两者各自的优势，.NET 体验可同时在在线和脱机状态下进行。

- .NET 体验是个性化的。.NET 体验读取存储在低层基础中的用户参数和配置信息，并自动将用户信息和参数结合在一起显示出来。.NET 体验根据你的登录 ID 已经知道你住在哪里，你不需要输入地址。

1.1.5 Visual Studio.NET

为了配合微软的.NET 战略，微软在经过四年的开发和两年的市场宣传以后终于发布了 Visual Studio.NET。“增加产出率，帮助企业更有效地适应快速变化且具有竞争性的市场”，这就是 Visual Studio.NET 的主要目标。通过提供端对端的 Web 开发能力和可升级、可重用的服务器端组件，Visual Studio.NET 可以容易、快速地创造出强大、可靠的企业级应用程序解决方案。

Visual Studio.NET 的新特性使开发人员可以实现以下功能：

- 利用高产出的开发工具提高开发速度：高度集成、统一的可视化环境简化了开发应用程序的过程，减少了学习上的弯路。

- 快速设计、功能全面的 Web 应用程序：借助于 Web 表单，开发人员使用原来用于构建基于表单的桌面应用程序的工具就可创建跨平台、跨浏览器可编程的 Web 应用程序。Web 表单在 Web 服务器上运行，这样就提供了快速的运行性能，建立了可运行在任何浏览器与 HTML3.2 兼容的文档。

- 使用 XML Web 服务可以很容易地简化分布式计算：借助于 XML Web 服务，开发人员就可以使用标准的互联网协议通过线路调用企业逻辑。XML Web 服务将 HTTP 作为优先使用的传输机制。XML 用于格式化请求的输入和输出参数，这样请求就不会绑定到任何特定组件或对象的调用约定上。最终，XML Web 服务就可以被任何语言访问，可以被任何组件模型使用，能运行在任何操作系统中。

- 快速构建中间层企业组件：Visual Studio.NET 的一个主要目标是为服务器的开发提供快速应用程序开发（RAD）工具。Visual Studio.NET 为创建中间层企业逻辑带来的产出率与 Visual Basic 带给基于表单的开发产出率具有相同的等级。所创建的组件非常强大并具有升级性。

- 构建可升级的有效解决方案：Visual Studio.NET 可以很容易地建立可靠的应用程序和组件，而且默认是可升级的。例如，在通过 Microsoft Visual C++ 构建应用程序时，新的 ATL 服务器提供了重要的性能推动。

Visual Studio.NET 提供了一套完整而强大的开发环境，是微软.NET 战略中的重要一环。它是微软用来将它的 Windows 垄断地位转换成日益增长的 Web 服务技术市场中的关键开发工具。任何原来 VC、VB 的程序员都要受到.NET 的冲击和洗礼。正像著名的评论

员 David Chappell 在 Develop Mentor 年会上所说的那样,“如果你不转变,就离开软件产业”。不过不必害怕,对于原来的 VC 程序开发人员来说,.NET 将使编程的工作变得更加简单。

1.2 Visual C++.NET 的新特性

Visual C++.NET 是使用 C++ 开发语言创建基于 Microsoft Windows 和 Microsoft .NET 的应用程序、动态 Web 应用程序和 XML Web 服务的综合工具集。这一强劲的开发环境包括工业标准活动模板库(ATL)和 Microsoft 基础类(MFC)库、高级语言扩展以及允许开发人员有效编辑和调试源代码的强大集成开发环境(IDE)功能。Visual C++.NET 中的新功能包括 C++ 属性、托管扩展、ATL Server, 以及大量的新库、IDE 和编译器优化与增强功能。

Visual C++.NET 为开发人员提供了许多专业级功能,能够为 Windows 和 Web 创建功能非常强大的应用程序和组件。自始至终,该工具都提供了帮助简化 C++ 软件开发过程的功能。

应用程序向导为快速启动项目提供了强大的模板。其中包括的 MFC 和 ATL 应用程序框架库是最强劲、最有效和最流行的用于开发基于 Windows 应用程序的库,现在又针对 Windows XP 得到了增强。这些功能与专业级编译器和高级调试器集成在一起,共同提供了一致和有效的开发体验。

Visual C++.NET 为开发人员创建强大和关注性能的应用程序提供了业已证实的面向对象的语言。通过高级模板功能、低级平台访问和一个优化编译器,Visual C++.NET 为生成强劲的应用程序和组件提供了出色的功能。该产品让开发人员能够创建广泛的解决方案,包括 Web 应用程序、基于智能客户端的基于 Microsoft Windows 应用程序,以及用于瘦客户端和智能客户端移动设备的解决方案。C++ 是全世界最流行的系统级语言,而 Visual C++.NET 则为开发人员提供了创建软件的世界级工具。通过 Visual C++.NET,开发人员能够享用世界级的 C++ 开发工具。C++ 是一种可互操作的基于标准的语言,而 C++ 技能也可以在多个计算环境和社区间利用。

Visual C++.NET 为核心 C++ 开发人员提供了高性能和精细控制开发功能。Visual C++.NET 允许开发人员在操作系统上创建本机运行的应用程序。开发人员还可利用 C++ 托管扩展,面向 .NET 框架和公共语言运行库创建应用程序。由于 Visual C++.NET 开发人员可以直接对操作系统进行写操作,因此可以得到最大的控制权;同时由于他们还可以对 .NET 框架进行写操作,因此又拥有最大的灵活性。跟 Visual Basic 和 Visual C# 一样,Visual C++.NET 还允许开发人员创建强大的 XML Web 服务。



在与 .NET 兼容的众多语言中, Visual C++.NET 是惟一的一种既支持 .NET 框架提供的托管代码模型,又支持基于 Windows 的非托管本机的语言。通过同时支持两种编程模型, Visual C++.NET 既保留和增强了代码和技能方面的现有投资,又为开发人员和组织提供了尽可能多的选择。

这里首先讲一下系统机制方面的改进,其次是关于集成开发环境特性的内容。

1.2.1 C++属性

C++属性允许公布式编程，它们可以通过尽可能减少重复样板源代码来简化开发工作。开发人员可以通过应用简单的属性标记，以 XML Web 服务形式公开新的和现有的 ATL COM 组件。C++属性标记的引入还减少了独立接口定义语言（IDL）模块的需要。

1.2.2 通用语言运行时

通用语言运行时，即 Common Language Runtime。Microsoft .NET 给开发人员带来了一种全新的开发框架，而通用语言运行时则处于这个框架的最低层，是这个框架的基础。开发人员对于所谓的 C 运行时、VB 运行时、Java 虚拟机这些概念已经非常熟悉了，而通用语言运行时则为多种语言提供了一种统一的运行环境。另外它还提供了更多的功能和特性，比如统一和简化的编程模型，用户不必迷惑于 Win32 API 和 COM；避免了 DLL 的版本和更新问题，从而大大简化了应用程序的发布和升级；多种语言之间的交互，例如，我们甚至可以在 VB 中使用 C++编写的类；自动的内存和资源管理等。Microsoft .NET 正是基于通用语言运行时，实现了这些开发人员梦寐以求的功能。

1.2.3 受控代码（C++托管扩展）

基于通用语言运行时开发的代码称为受控代码（托管代码），它的运行步骤大体如下：首先使用一种通用语言运行时支持的编程语言编写源代码，然后使用针对通用语言运行时的编译器生成独立于机器的微软中间语言（Microsoft Intermediate Language），同时产生运行所需的元数据，在代码运行时再使用即时编译器（Just In Time Compiler）生成相应的机器代码来执行。

托管代码在 .NET 运行时环境下运行。使用托管代码不是必需的，但使用它有许多优点。例如，使用 C++托管扩展用托管代码编写的程序可以运行公共语言运行库来提供内存管理、跨语言集成、代码访问安全和对象的自动生存期控制等服务。Visual C++托管扩展扩展了标准的 C++语言，便于它向新的和现有的应用程序添加对 .NET 框架的支持。

使用 C++托管扩展，开发人员可以利用 Microsoft .NET 合并下一代技术（包括资源管理、统一类型和组件远程处理）。使用 .NET，开发人员可以获得用于无缝垃圾回收和减少程序复杂性的出色的内存管理技术。通过在进行有效远程过程调用的同时使用 .NET 框架通用类型系统，开发人员可以利用以 20 多种支持 .NET 的语言中的任意一种写成的代码。

Visual C++ .NET 的非受控代码（Unmanaged Code）是 VC6.0 的升级，可直接生成二进制代码，支持 MFC；可用 ATL Server 开发底层的 XML Web services。

Visual C++的托管扩展（Managed Extensions）是 Visual C++的 .NET 版本，仍然是 Visual C++，但是对 ANSI 标准 C 语言做了扩展，支持所有 VC 的特性，还可以完成所有 C#能做到的，可以将现有的代码直接移植到 .NET，并且完全支持 .NET Framework。C++托管扩展通过使用强大的新语言关键字确定 Microsoft .NET 平台目标，并使用基于 .NET 的功能（如垃圾回收和基于 .NET 的组件技术）实现新的和现有的基于 Windows 的本机应用程序。



1.2.4 基础类库与命名空间

当然对于开发者而言，他们除了关心通用语言运行时提供那么多新特性外，还想知道它究竟给开发者提供了什么样的编程接口，这就是基础类库（Base Class Library）。这组基础类库包括了从输入输出到数据访问等各方面，提供了一个统一的面向对象的、层次化的、可扩展的编程接口。它使用一种点号分隔的方法，使得查找和使用类库非常容易。例如，基础类库中的根，它的命名空间是 System，提供数据访问的类库的命名空间是 System.Data。在使用时，开发者只需在自己的应用中添加所需的基础类库的引用，然后就可以使用这个类库中的所有方法、属性等。跟传统的 Windows 编程相比，使用和扩展基础类库都非常容易，这使得开发者能够高效、快速地构建基于下一代互联网的网络应用。

1.2.5 .NET 框架类库

Visual C++ .NET 开发人员还可以使用 Microsoft .NET 框架类库并利用强大的内置功能：包括一组丰富的集合类、网络支持、多线程支持、字符串和正则表达式类，以及对 XML、XML 架构、XML 命名空间、XSLT、Xpath 和 SOAP 的广泛支持。

1.2.6 ATL Server

当开发人员选择用 C++ 编写部分 Web 应用程序时，他们通常会考虑性能或控制原因。ATL Server 在一组简单和可扩展的 ATL 类中封装了开发高性能 C++ Web 应用程序的最佳做法供开发人员重用。ATL Server 包括让 Web 应用程序和 XML Web 服务扩展到一个 Web 服务器群集内的多个 CPU 的功能。

增强的库合并了最新和最先进的功能。使用 ATL Server 创建快速动态的 Web 站点和 XML Web 服务，通过对统一描述、发现和集成（UDDI）Web 服务目录的完全集成的支持，可以在应用程序中方便地查找和使用公用的 XML Web 服务。

1.2.7 类库的升级

Visual C++ .NET 合并了最强劲、最有效和最流行的可用于 Windows 工业标准的 Microsoft 基础类（MFC）库和活动模板库（ATL）的应用程序框架库，并加入了针对 Microsoft Windows XP 的增强功能。

下面列出了 Visual C++ .NET 中新增的或经过更改的库。

- 活动模板库（ATL）；
- ATL Server；
- C 运行时库；
- Microsoft 基础类（MFC）库；
- OLE DB 模板；
- 共享类；
- 标准 C++ 库；

- 旧 iostream 库。

改进后的 MFC 主要体现在以下几个方面：

- 使用新的 MFC DLL (MFC70.DLL)，不再兼容 MFC42.DLL，但你的源程序依然保持兼容性（消息映射类型更加安全，所以可能导致某些代码违例）。
- MFC 和 ATL 结合更加紧密了，一些通用的类如 CString 可以同时被两个库使用。
- 头文件与新的 Platform SDK 同步，支持 Windows 2000/XP 中的新 UI 特性，如主题 (themes)，manifest resources，Active Accessibility 和新的通用对话框。
- 增加了不少 UI 的类，包括支持 DHTML 的对话框和支持扩展 BITMAP 的 CImage。
- 新的工具类可以在 MFC 和 ATL 中使用，比如正规表达式，计数器和安全。
- MFC 应用程序有力地支持 Web Services ATL 服务器类编写 Web Services 和应用程序。
- 使用新的 OLE DB 特性使得高效率地访问数据库从来就没有像现在这样容易。
- STL 更新。

1.2.8 编辑器特性的改进

通过出色的 IDE，Visual C++ .NET 为程序员提供了终极开发人员环境。“起始页”为开发人员提供了一个门户，由此只需一次单击即可访问有关最近使用的项目、首选项、更新程序和 MSDN Online 社区的信息。Microsoft IntelliSense®功能、“工具箱”和“任务列表”大大提高了工作效率，而自动隐藏窗口和多监视器支持功能可帮助程序员尽可能多地利用屏幕并自定义开发环境，强劲的开发环境使 Visual C++ 成为专业应用程序开发人员的理想选择。

Visual Studio 为 VS.NET 所支持的所有语言提供了一个统一的代码编辑器，统一的集成环境，没有 VB IDE，也没有 VC，也没有 VI。所有的开发都在同一个环境下。其工具条、任务窗口、调试环境都是相同的，而对每种语言又支持特定的特性。编辑器有了很大改进，如字提示、递增搜索、代码大纲、重叠文本、行号、分色显示和快捷键。编辑器还提供了许多特定于语言的特性，如它能在你输入时完成原型和函数调用。且采用了智能感知技术，包括对象自动弹出所有方法和成员；自动显示所有重载方法的参数；大、小括号等自动匹配标示等。

除了编程语言，编辑器还支持 HTML 文档，层叠样式表单，甚至是 XML 的开发。而且，编辑器提供了源视图和数据视图。



对于 VC6.0 的使用者来说，一个要注意的地方就是 ClassWizard 已被几个新向导所取代，这几个向导分别对添加成员变量、消息处理程序、方法、属性和事件提供了更多的控制。

1.2.9 调试器特性的改进

Visual C++ .NET 可以快速有效地调试应用程序，在方便的停靠窗口中跟踪应用程序调用堆栈、变量、线程、内存、模块和注册表使用情况。在文件、代码行、内存地址和变量上设置断点。通过使用“编辑并继续”在调试期间更改代码，Visual C++ .NET 具有跟踪

MFC、ATL 和标准模板库 (STL) 支持库 (随完全源代码提供) 的能力。

Visual Studio .NET 为所有 Visual Studio 语言提供了单个集成调试器。新的、统一的接口集成了旧的 Visual C++ 和 Visual Basic 调试器的功能, 并增加了很多新功能。对于调试, 主要的增强功能包括:

- Visual Basic .NET、Visual C++ .NET、Visual C# .NET、C++ 的托管扩展、脚本和 SQL 的跨语言调试。
- 为 Microsoft .NET 框架公共语言运行库而编写的应用程序和 Win32 本机应用程序的调试。
 - 附加到主机上或远程计算机上的正在运行的程序。调试时自动附加到服务器进程 (对于 ATL Server 应用程序、ASP.NET Web 应用程序和 ASP.NET Web 服务)。
 - 通过在单个 Visual Studio 解决方案中启动多个程序或者通过附加到已经在运行的其他程序来调试多个程序。
 - 在 Visual C++ 中使用运行时错误检查来捕获常见运行时的错误, 例如, 堆栈指针损坏、局部数组超限、堆栈溢出、未初始化的局部变量上的依赖项和向较短变量进行赋值时的数据损失。
 - 用 /GS 选项检查 Visual C++ 应用程序的缓冲区安全性。可以使用该选项来检测缓冲区溢出 (它将改写返回地址)。有关更多信息, 请参阅 /GS (缓冲区安全检查)。
 - 在尚未加载的 DLL 中设置断点。在加载 DLL 时, 断点将自动成为活动的。不必在“其他 DLL”对话框中指定要调试的 DLL (而在 Visual C++ 调试器中需要这样做)。
 - 用客户端与服务器应用程序之间的集成调用堆栈一步步地从 XML Web services 客户端走向 XML Web services 服务器端。
 - 无模式窗口已经取代了用于断点和线程的模式对话框, 从而可以更加灵活地控制和访问工作区中的功能。

1.2.10 编译器的新特性

Visual Studio.NET 里的 C++ 可以从两个方面来介绍: 标准的 C++ 和 Managed Extensions for C++。

对标准 C++ 的扩充主要是为了保持兼容性; 也就是说, 为了保持国际标准化组织 (ISO) 所规定的 C++ 语言特性。而 Managed Extensions for C++ 则是为了把 C++ 纳入到 .NET 框架之中。

Visual Studio.NET 下的标准 C++ 为了保持对 ISO 标准的兼容性, 做了如下改进:

- (1) 虚拟函数的返回值现在支持共变 (covariant) 类型了, 这是类层次体系上的一个重大改进。
- (2) 支持对静态整数类型常量 (static const integer) 成员的显示初始化。
- (3) 主函数的返回值默认为 0。

Visual Studio .NET 的受管 C++ (Managed Extensions for C++) 提供三种基本应用策略:

- (1) 为原有的 API 提供 .NET “包装”, 把 C++ 类统统移植到 .NET 平台中。
- (2) 混用 C++ 类与微软 .NET 的三种框架类:
 - 核心语言支持, 例如, 收集类和系统输入 / 输出;