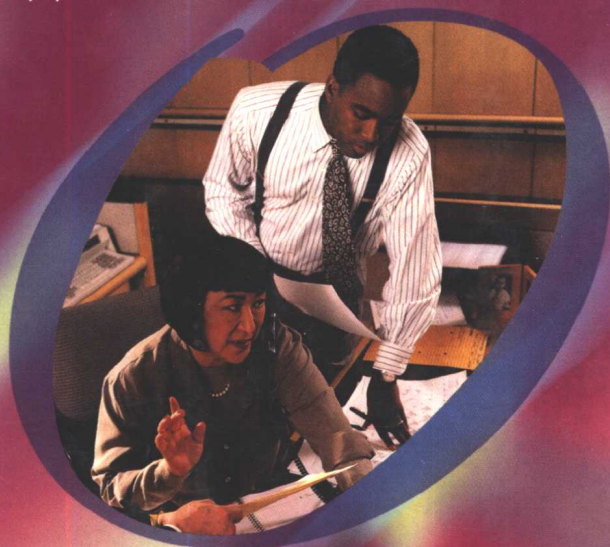


“十五”国家重点电子出版物规划项目 计算机知识普及和软件开发系列  
体验办公新工具丛书 (4)



# Master programming with Visual C#.NET

深入  
编程



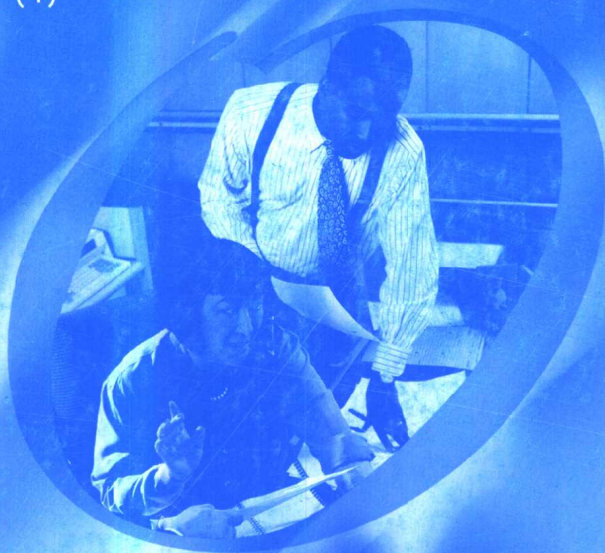
北京希望电子出版社 总策划  
徐可 熊伟 袁和金 编写



北京希望电子出版社  
Beijing Hope Electronic Press  
[www.bhp.com.cn](http://www.bhp.com.cn)



“十五”国家重点电子出版物规划项目 计算机知识普及和软件开发系列  
体验办公新工具丛书 (4)



Master programming with  
**Visual C#.NET**



**深入  
编程**

北京希望电子出版社 总策划  
徐可 熊伟 袁和金 编写



北京希望电子出版社  
Beijing Hope Electronic Press  
[www.bhp.com.cn](http://www.bhp.com.cn)

## 内 容 简 介

本书是关于 VC#.NET 的应用开发指导书,目的在于用丰富的范例教会读者掌握 VC#.NET 的编程方法和技术。全书由 12 章组成,内容涵盖 C# 的语言介绍、集成开发环境介绍、控件编程、MDI 程序、数据库开发、多线程开发、网络编程以及混合语言组件开发等。

本书内容丰富、全面详实,结构合理,重点突出,深入浅出,实用性强、指导性强,融入了作者全新的编程理念和技巧。所介绍的所有实例程序均实际调试通过,并具有相当的代表性。

本版教程面向初中级用户,同时对高级用户也有重要参考价值。

本版 CD 内容为实例源代码,并附赠《Visual c++ 6.0 编程高手》多媒体教学软件。

- 盘书系列名 : “十五”国家电子出版物规划项目 计算机知识普及和软件开发系列  
体验办公新工具丛书(4)
- 盘 书 名 : Visual C#.NET深入编程
- 总 策 划 : 北京希望电子出版社
- 文 本 著 者 : 徐可 熊伟 袁和金
- 责 任 编 辑 : 蒋湘群
- C D 制 作 者 : 希望多媒体开发中心
- C D 测 试 者 : 希望多媒体测试部
- 出 版、发 行 者 : 北京希望电子出版社
- 地 址 : 北京中关村大街 26 号, 100080  
网址: www.bhp.com.cn E-mail: lwm@hope.com.cn  
电话: 010-62562329,62541992,62637101,62637102,62633308,62633309 (发行部)  
010-62613322-215 (门市) 010-62547735 (编辑部)
- 经 销 : 各地新华书店、软件连锁店
- 排 版 : 希望图书输出中心 全卫
- CD 生 产 者 : 北京中新联光盘有限责任公司
- 文 本 印 刷 者 : 北京双青印刷厂
- 开 本 / 规 格 : 787 毫米×1092 毫米 16 开本 32.25 印张 579.6 千字
- 版 次 / 印 次 : 2001 年 12 月第 1 版 2001 年 12 月第 1 次印刷
- 印 数 : 1-5000 册
- 本 版 号 : ISBN 7-980008-33-2
- 定 价 : 52.00 元(本版 CD)

说明: 凡我社售出的产品如有残缺,可执相关凭证与我社调换。

# 前 言

Visual Studio.NET是Microsoft为实现.NET技术而开发的一整套工具组件,具有许多令人兴奋的特征和一些全新的内容。如:新的独一无二的集成开发环境、称作“C#”的编程语言等。本书将对Visual Studio.NET平台上如何利用C#进行软件开发进行介绍。同VC++.NET, VB.NET一样, VC#.NET也是一种具有可视化编程界面的开发工具。事实上, .NET 框架与编程语言无关,任何语言都可以支持.NET 框架。目前可以用许多语言来创建.NET 程序,包括: C++, Visual Basic, Java Script 和 Microsoft 的最新语言C#,它们使用的都是相同的.NET类库。

C#是从C 和C++ 演变而来的一种简单的、面向对象的和类型安全的新型程序设计语言。然而,仅仅使用C#编程语言是无法完成复杂的开发工作的,只有依赖于.NET环境下的强大的类库支持以及其各种可视化编程工具,成为功能强大的开发工具——VC#.NET,才能够研发应用程序。

本书将对在Visual Studio.NET平台上如何利用C#进行软件开发进行介绍。

本书对.NET 开发环境、C#语言、.NET 常用类库和 C#的各种编程方法进行了详细介绍,全书利用丰富的源程序进行讲解,对 VC#.NET 学习者来说,将会是一本难得的教程,具有很强的实用价值。

## 本书结构安排

第一章为 Visual Studio.NET 简介,介绍了 Visual Studio. net 的特点和基本知识。

第二章为 Visual Studio. net 的安装,介绍了 Visual Studio. net 安装的软件、硬件需求、安装的过程和经常出现的问题的解决方法,使读者能够了解 VS.NET 的安装过程。

第三章为 C#语言基础,介绍了 C#语言的结构,语法等,全章用大量的代码详细介绍了 C#语言,使读者能够迅速熟悉 C#这个全新的语言。

第四章为.NET 集成开发环境,介绍了.NET 集成开发环境的使用,各个窗口、菜单、工具栏的作用,以及一些设置。这部分将会使读者迅速熟悉.NET 开发环境。

第五章为 VC#快速上手,这部分从设计几个简单的应用程序开始,使读者很快了解 VC#的基本程序开发方法。对常用的 Windows 控件的介绍,可帮助读者很快编写实用的 Windows 程序。

第六章为图形程序开发,介绍了 Windows 画图的一般方法,讨论了.NET 提供的画图常用类。该章的最后部分将会和读者一起,一步步完成一个画图的程序。

第七章为文件操作,介绍了文件的一般使用方法,包括文件的打开、读取、写入、关闭等等。在此基础上,介绍了有关文件操作的其他类和方法,大量的代码和一个完整的实例会使读者迅速熟悉文件的操作方法。

第八章为 MDI 程序开发及键盘消息处理,介绍了单文档和多文档程序,并附带介绍了

73P00/10-06

键盘消息处理的一般方法，用一个实例程序简单明了地说明了该类程序的开发方法。

第九章为数据库开发，介绍了 ADO.NET 的开发方法，利用所介绍的类和方法，通过一个实例，读者一定会掌握数据库的基本开发方法。

第十章为多线程开发，介绍了多线程应用程序的开发方法，并在实例中演示了多线程的使用方法。

第十一章为网络开发，介绍了网络应用程序的开发，侧重于 Socket 程序的编写，将会和读者一起，完成一个实用的网络通讯小程序。

第十二章利用多种语言开发组件，介绍了如何利用不同的语言开发和调用组件。本章利用一个实际的例子使读者能够轻松掌握这个技术。

本书由徐可、熊伟、袁和金主编，其他参加编写的人员主要有胡卓颖、徐进、朱宏涛、周凌、钟伟峰、李冰、王健、戴斌华、刘绍辉等。

由于 Visual Studio.NET 问世不久，值得借鉴的中文资料相当稀少，加之本书编者水平有限，书中难免有缺点和错误，一些术语的使用可能也不尽如人意，恳请广大读者批评指正。

读者有任何技术问题需要咨询，请和我们联系。

Email: [xxroom@sina.com](mailto:xxroom@sina.com)。

编者

# 目 录

<b>第一章 Visual Studio.NET 简介</b> .....1	<b>第三章 C#语言基础</b> ..... 26
本章要点.....1	本章要点.....26
1.1 Visual Studio.NET 简介.....1	3.1 绪论.....26
1.1.1 设计应用程序和共享 最佳方法.....1	3.1.1 Hello, world.....27
1.1.2 企业框架和模板.....2	3.1.2 自动内存管理.....28
1.1.3 满足业务需求.....2	3.1.3 类型.....30
1.1.4 通过软件建模表达体系结构....3	3.1.4 预定义类型.....32
1.1.5 测试 XML Web 服务和 应用程序.....3	3.1.5 数组类型.....34
1.2 Visual Studio.NET 安装要点.....3	3.1.6 统一系统类型.....36
1.2.1 解决方法.....4	3.1.7 语句.....37
1.3 解答 .NET.....5	3.1.8 类.....39
1.4 本章小结.....8	3.1.9 结构类型.....39
本章习题.....8	3.1.10 接口.....40
<b>第二章 安装 Visual Studio.NET 7.0</b> .....9	3.1.11 代理(delegates).....41
本章要点.....9	3.1.12 枚举.....41
2.1 安装要求.....9	3.1.13 命名空间.....41
2.1.1 硬件、系统需求.....9	3.1.14 属性.....42
2.1.2 安装 Microsoft Windows Component Update.....9	3.1.15 索引器.....43
2.1.3 组件的单独安装.....13	3.1.16 事件.....44
2.2 在 Windows 2000 操作系统下安装 Visual Studio.NET 7.0.....18	3.1.17 Attribute.....45
2.3 启动 Visual Studio.NET.....23	3.2 语法结构.....47
2.4 在其它操作系统上安装 Microsoft Visual Studio.NET.....24	3.2.1 预处理.....47
2.4.1 在 Windows® NT 4.0 操作系 统下安装 Visual Studio.NET ..24	3.2.2 标识符.....51
2.4.2 在 Windows Me 或 Windows 98 操作系统下 安装 Visual Studio.NET.....25	3.2.3 关键字.....52
2.5 本章小结.....25	3.2.4 数据类型.....52
本章练习.....25	3.2.5 操作符和标点符号.....56
	3.3 基本概念.....57
	3.3.1 声明.....57
	3.3.2 成员.....57
	3.3.3 成员可见性 、 (member-access).....58
	3.3.4 存取控制.....61
	3.4 类型.....61
	3.4.1 值类型.....62

3.4.2	缺省值 .....	63	3.8.6	表达式语句 (expression-statement) .....	111
3.4.3	简单类型 .....	64	3.8.7	条件语句 (selection-statement) .....	112
3.4.4	引用型 .....	66	3.8.8	循环语句 (iteration-statement) .....	116
3.5	变量 .....	67	3.8.9	转子语句(Jump) .....	119
3.5.1	变量的分类 .....	67	3.8.10	checked 和 unchecked 语句 .....	123
3.5.2	变量缺省值 .....	69	3.8.11	Lock 语句 .....	123
3.5.3	变量明确赋值 .....	69	3.9	命名空间 .....	124
3.5.4	变量引用 (variable-reference) .....	72	3.9.1	编译单元 .....	124
3.6	类型转换 .....	72	3.9.2	命名空间声明 (namespace-declaration) ...	124
3.6.1	隐式转换 .....	72	3.9.3	Using 命令 .....	125
3.6.2	显式转换 .....	73	3.9.4	命名空间的应用 .....	127
3.7	表达式 .....	75	3.10	类 .....	132
3.7.1	表达式的分类 .....	75	3.10.1	类声明 .....	133
3.7.2	运算符 .....	76	3.10.2	类成员 .....	136
3.7.3	加括号的表达式 .....	78	3.10.3	常量(constant) .....	139
3.7.4	表达式调用 .....	78	3.10.4	域 .....	141
3.7.5	元素访问(element-access) .....	79	3.10.5	方法 .....	147
3.7.6	this 访问 .....	81	3.10.6	属性 .....	161
3.7.7	Base 访问 .....	84	3.10.7	事件 .....	171
3.7.8	递增和递减操作符 .....	86	3.10.8	索引器 .....	175
3.7.9	new 操作符 .....	88	3.10.9	操作符 .....	178
3.7.10	typeof 操作符 .....	93	3.10.10	构造函数 .....	181
3.7.11	sizeof 操作符 .....	94	3.10.11	析构函数 .....	187
3.7.12	checked 和 unchecked 操作符 .....	95	3.10.12	静态构造函数 .....	187
3.7.13	一元表达式 .....	101	3.11	结构 .....	190
3.7.14	二元表达式 .....	102	3.11.1	结构声明 .....	190
3.8	语句 .....	107	3.11.2	结构成员 .....	191
3.8.1	终点和可达 .....	107	3.11.3	结构例子 .....	191
3.8.2	分程序 .....	108	3.12	接口、代理和枚举 .....	196
3.8.3	空语句 (empty-statement) .....	108	3.12.1	接口 .....	196
3.8.4	标号语句 (labeled-statement) .....	109	3.12.2	代理(delegates) .....	197
3.8.5	声明语句 (declaration-statement) .....	109	3.12.3	枚举 .....	198

3.13	版本.....	199	5.3.1	控件的基本概念.....	264
3.14	本章小结.....	202	5.3.2	我们从哪里得到控件.....	265
	本章习题.....	202	5.3.3	控件的使用方法简介.....	265
<b>第四章</b>	<b>Visual Studio.NET 集成开发环境.....</b>	<b>203</b>	5.3.4	菜单、工具栏、状态栏、 和图标的使用.....	270
	本章要点.....	203	5.4	本章小结.....	275
4.1	Visual Studio .NET 集成开发 环境介绍.....	203		本章习题.....	275
4.1.1	VISUAL STUDIO .NET 开发环境.....	203	<b>第六章</b>	<b>图形程序开发.....</b>	<b>276</b>
4.1.2	Visual Studio .NET 集成 开发环境的新内容.....	204	6.1	Vcsharp 画图基本概念.....	276
4.1.3	Visual Stdio .NET 常用 窗口对象简介.....	208	6.1.1	上下文设备, 句柄和 Graphics 对象.....	276
4.2	Visual Studio .NET 菜单介绍.....	222	6.1.2	画一条直线.....	277
4.2.1	File 菜单组.....	223	6.1.3	画图和填充.....	279
4.2.2	Edit 菜单组.....	227	6.1.4	构造 Regions.....	280
4.2.3	View 菜单组.....	229	6.1.5	Ellipses 和 Arcs.....	280
4.2.4	Project 菜单组.....	232	6.1.6	Drawing, Positioning, 和 Cloning Images.....	281
4.2.5	Build 菜单组.....	235	6.1.7	Polygons.....	283
4.2.6	Debug 菜单组.....	237	6.1.8	Pens, Lines, and Rectangles 高级使用.....	284
4.2.7	Tools 菜单组.....	238	6.1.9	Regions 高级使用.....	285
4.2.8	Window 菜单组.....	242	6.1.10	Paths.....	286
4.2.9	.NET 开发环境中的工具条.....	244	6.2	Graphics 类.....	288
4.3	.NET 下开发程序的一般步骤.....	244	6.2.1	Graphics 概述.....	288
4.4	本章小结.....	247	6.2.2	Rectangle, Size 和 Point.....	288
	本章习题.....	247	6.2.3	Graphics 画图.....	289
<b>第五章</b>	<b>VC#快速上手.....</b>	<b>248</b>	6.3	Pen,Font,Brush,Bitmap 和 Color 简介.....	293
5.1	新的开端.....	248	6.3.1	Brush.....	293
5.1.1	“hello, world!”一个老土 的例子.....	248	6.3.2	Bitmap.....	294
5.2	Windows 下的编程, 基于 Form 的应用程序.....	254	6.3.3	Font 和 Color.....	294
5.2.1	Windows 下编程要注意的 几个问题.....	254	6.3.4	Pen.....	294
5.2.2	开发一个 WINDOWS 下的 应用程序.....	256	6.4	图形编程.....	295
5.3	常用的 form 控件.....	264	6.4.1	创建应用程序框架.....	295
			6.4.2	界面设计.....	297
			6.4.3	代码设计.....	305
			6.4.4	程序运行结果.....	310



6.5	本章小结.....	313	9.1.1	数据库简介.....	361
	本章练习.....	313	9.1.2	数据库的体系结构.....	361
<b>第七章</b>	<b>文件操作.....</b>	<b>314</b>	9.1.3	数据库的操作过程.....	362
7.1	VC#的文件常用类简介.....	314	9.2	数据库世界的世界语--SQL.....	364
7.2	文件.....	315	9.2.1	数据定义语言 (DDL).....	366
7.2.1	打开文件.....	316	9.2.2	数据控制语言 (DCL).....	370
7.2.2	文件存取.....	317	9.2.3	数据处理语言 (DML).....	371
7.2.3	文件读写操作.....	319	9.3	从头学起, 开始 ADO 编程.....	380
7.2.4	关闭文件.....	323	9.3.1	基本的 ADO 编程模型.....	380
7.3	Stream.....	323	9.3.2	使用 ADO 的对象实现 ADO 的编程模型.....	382
7.3.1	Stream 类及其派生类.....	323	9.4	ADO 数据库编程开发应用实例.....	399
7.3.2	用 Stream 进行同步 I/O 和异步 I/O 操作.....	326	9.4.1	设计应用程序.....	399
7.3.3	在网络上使用 Stream.....	328	9.4.2	编译执行.....	425
7.3.4	Directory.....	329	9.5	本章小结.....	427
7.4	文件编程.....	329		本章习题.....	427
7.4.1	基本思想.....	329	<b>第十章</b>	<b>多线程开发.....</b>	<b>428</b>
7.4.2	界面设计.....	329	10.1	线程基本知识.....	428
7.4.3	代码设计.....	331	10.1.1	线程与进程.....	428
7.4.4	程序执行.....	335	10.1.2	线程是如何工作的.....	429
7.5	本章小结.....	338	10.1.3	线程的优点.....	429
	本章练习.....	339	10.2	工作线程.....	430
<b>第八章</b>	<b>MDI 程序开发及键盘消息处理.....</b>	<b>340</b>	10.2.1	Windows 的线程实现函数.....	430
8.1	MDI 程序简介.....	340	10.2.2	启动线程.....	430
8.1.1	单文档与多文档应用程序.....	340	10.3	多线程编程的常用类.....	434
8.1.2	文档类.....	340	10.3.1	线程同步.....	434
8.1.3	视类.....	340	10.3.2	Thread 类.....	435
8.1.4	框架.....	340	10.3.3	Mutex 类.....	440
8.2	创建应用程序的步骤.....	341	10.3.4	AutoResetEvent 和 ManualResetEvent.....	441
8.3	编写 MDI 程序实例.....	342	10.3.5	管道 (Monitor).....	443
8.3.1	应用程序设计.....	342	10.3.6	Interlocked.....	443
8.4	键盘消息响应.....	354	10.3.7	ThreadPool.....	444
8.5	程序运行结果.....	358	10.3.8	ReaderWriterLock 类.....	444
8.6	本章小结.....	360	10.4	多线程编程.....	445
	本章练习.....	360	10.4.1	界面设计.....	445
<b>第九章</b>	<b>数据库的开发.....</b>	<b>361</b>	10.4.2	设计几个线程函数.....	448
9.1	数据库, 并不像听起来那么难.....	361			

10.4.3	增加 Mutex 类 .....	451		SendTo ( ) .....	470	
10.4.4	增加 Thread 类 .....	451		11.3.6	接收数据 Receive ( ) 和 ReceiveFrom ( ) .....	472
10.4.5	启动线程 .....	452		11.3.7	关闭 Close ( ) .....	475
10.4.6	关闭线程 .....	453		11.3.8	Select ( ) .....	475
10.4.7	线程清除工作 .....	454		11.4	在网络上的数据表示 .....	476
10.4.8	程序运行 .....	455		11.5	Socket 编程实例 .....	482
10.5	本章小结 .....	456		11.5.1	创建新项目 .....	482
	本章练习 .....	456		11.5.2	添加命名空间 .....	483
<b>第十一章</b>	<b>网络开发</b> .....	<b>457</b>		11.5.3	设计几个辅助函数 .....	483
11.1	网络基本知识 .....	457		11.5.4	添加控件 .....	488
11.1.1	网络发展的历史 .....	457		11.5.5	接收和发送数据 .....	491
11.1.2	网络参考模型 .....	458		11.5.6	退出程序 .....	493
11.1.3	Client / Server 编程模式 .....	461		11.5.7	执行程序 .....	493
11.2	VS.NET 的 Socket .....	462		11.6	本章小结 .....	495
11.2.1	建立 Socket .....	462			本章练习 .....	495
11.2.2	连接一个 Socket .....	463		<b>第十二章</b>	<b>用多种语言开发自己的组件</b> .....	<b>496</b>
11.2.3	使用 Socket 类 .....	464		12.1	设计组件 .....	496
11.2.4	Socket 调用过程 .....	466		12.1.1	设计一个用 VC#编写的 组件 .....	496
11.3	Socket 类 .....	467		12.1.2	设计一个用 VB 写的组件 .....	499
11.3.1	Socket 构造函数 .....	467		12.2	组件的调用 .....	500
11.3.2	绑定地址 Bind ( ) .....	468		12.3	本章小结 .....	503
11.3.3	建立连接 Connect ( ) .....	469			本章练习 .....	504
11.3.4	监听服务请求 Listen ( ) .....	469				
11.3.5	发送数据 Send ( ) 和 SendTo ( ) .....	470				

# 第一章 Visual Studio.NET 简介

## 本章要点

本章简要介绍了 Visual Studio.NET 的基本概念，分析了它的特性和优势，并就 Visual Studio.NET 安装过程中可能出现的问题进行了说明，给出了相应的解决方法，最后针对 Visual Studio.NET 的一些常见的问题给出了解答。

## 1.1 Visual Studio.NET 简介

Visual Studio.NET 是 Microsoft 为实现 .NET 技术而开发的一整套工具组件。应用现有的可升级的、稳定可靠的工具快速开发 Web 应用程序是不容易的事情。Visual Studio.NET 很好地解决了这一难题。通过提供能极大提高终端对终端 Web 开发效率和可升级、可重用的服务器端组件，Visual Studio.NET 将更有效地帮助企业在充满竞争的市场中满足其需求。

Visual Studio.NET 具有许多令人兴奋的特征，其中某些是对早期 Visual Studio 版本的增强，而有些是全新的内容。例如，新的独一无二的集成开发环境，称作“C#”的编程语言，Visual Studio.NET 新的面向对象的特征以及生命周期开发工具。Visual Studio.NET 反映了以下方面的主要特点，它的生存期功能主要包括：

- 领先于业界的数据库设计工具，可以构建概念、逻辑和物理方面的模型。
- 对业界标准软件模型的全面支持。
- 一种新的基于 XML 的语言，用于说明体系结构策略和指导方针。
- XML Web 服务和应用程序的功能测试工具。
- XML Web 服务和应用程序的集成式负载测试和性能测试工具。
- 识别应用程序瓶颈的直观分析工具。
- 自动执行的回归测试功能。
- 可重复使用的框架组件。
- 集成的源代码控制。

### 1.1.1 设计应用程序和共享最佳方法

一个组织要获得成功，其中最重要的方法之一就是在开始编码之前正确设计应用程序。通常，一个组织总是希望最有经验的人员来负责设计阶段的工作。但是，大多数组织中经验丰富的高级程序员人数有限，因此很难做到为所有正在进行的开发项目合理分配设计资源。很多公司曾尝试采用如下方式分配设计资源：首先制定体系结构指南或策略（通常以

大量文档的形式),然后将它们分发给各个项目的开发小组。通常,这种方式的效果不是很好。开发人员认为这些策略文档内容陈旧、重点不突出,最终放弃使用它们。

为了帮助客户解决这一问题,Microsoft 引入了企业框架和模板,这是一组极具创新性的服务和功能,允许各个组织在 Visual Studio.NET IDE 下提供体系结构指南。此外,Visual Studio.NET 还提供了一组端到端的建模功能,可使一个组织内的所有资源最大限度地参与设计过程。

### 1.1.2 企业框架和模板

企业框架和模板提供了为开发组织提供体系结构指南所需的关键基础结构和内容。

- 模板提供一些用于定义和交流体系结构和开发策略的基础结构服务,客户和第三方可以对模板进行完全扩展和自定义。
- 框架为常见的开发任务提供可重复使用的组件,同时提供详细的实际参考应用程序。这些应用程序采用多种 Microsoft 技术介绍了体系结构概念和决策。

在 Visual Studio.NET 中,模板是围绕新的工程类型构建的,Visual Studio.NET 中包含用于描述开发策略的基于 XML 的语言(模板说明语言)。借助模板,客户可以为开发人员提供最初的应用程序起始点以及可供选择的合适技术,在高级开发和体系结构人员定义的开发策略中概括了这方面的信息。这样,开发人员会发现根据特定指南构建应用程序较为容易。

框架包含一些可重复使用的通用组件,开发人员可利用这些组件简化复杂功能的实现过程。框架将由 Microsoft 开发,并定期在 Web 上发布,因此开发人员无需等待 Visual Studio.NET 的主版本发布就可以提前使用新组件构建应用程序。Microsoft 最初提供的框架组件主要用于简化应用程序的管理和操作,将开发人员的要求减少为一个简单的应用程序编程接口(API)调用。

### 1.1.3 满足业务需求

有一种方法可以确保应用程序满足业务需求,这就是安排最了解业务流程的人参与设计过程。在过去,业务专家的参与仅仅局限在提出一系列的高级需求,然后再由开发人员将这些需求转换为应用程序。

通过 Visual Studio.NET,业务专家可以直接参与设计过程,这样可以提高整个应用程序生存期的工作效率。借助一系列概念建模功能,只需简单的事实陈述(如“一个员工有一部电话”)就可以开始数据库的设计。这种使业务专家参与设计过程的能力可以使应用程序更精确地反映业务需求。

### 1.1.4 通过软件建模表达体系结构

在应用程序生存期内，众多不同技术背景的人需要交流和沟通，才能成功创建新的应用程序。有经验的交流者都会意识到如何通过多种方式和多种媒体展示信息，使其更易于理解。对于应用程序体系结构的交流也是如此。除了提供设计文档以及企业框架和模板功能之外，Visual Studio.NET 还提供另一种说明和表达体系结构的机制：符合业界标准的软件建模。

软件建模使架构工程师或业务流程分析者能够直观展示应用程序的体系结构、工作流和方案。通过向文档和其它信息载体中添加图表，开发小组可以更容易地理解复杂应用程序的各部分是如何组合在一起的，从而提高开发效率。

### 1.1.5 测试 XML Web 服务和应用程序

许多研究表明，当在应用程序开发后期发现错误时，纠正错误的成本将大幅度增加。由于这个原因，也由于应用程序的复杂程度和业务需求在不断增加，客户迫切希望在整个应用程序生存期内都能支持测试功能。

如果构建的应用程序无法满足可扩展性、性能或功能方面的需求，最终将不得不花费巨大的人力物力去纠正错误。Visual Studio.NET 提供了一些帮助客户尽早发现重大错误的功能，其中包括一套功能强大的集成式测试工具，可以进行性能测试、可扩展性测试和功能测试。

为了简化 Web 服务测试，Visual Studio.NET 自动生成测试存根页，该页可以非常方便地与性能和功能测试工具一起使用。此外，用于识别分布式应用程序瓶颈的直观分析工具可以实现 XML Web 服务和应用程序的端到端测试。借助这些工具，客户可以在项目开发阶段的早期发现可扩展性、性能和功能方面的关键问题，从而减少这些问题对整个项目的不良影响。

## 1.2 Visual Studio.NET 安装要点

如果你以前安装过 NGWS SDK Preview 版，并且在此基础上安装了 Visual Studio.NET 的 Alpha 版本，那么如果你在此基础上安装 Microsoft .NET Beta2 版，将会发现你的机器速度变得非常慢，仔细检查一下，会发现 CPU 一直被占用到 100%，其中占用最大的是 Mdm.exe 进程，一般高达 95%。同时，在此时 VS.NET 的功能已经基本丧失了，只要一调试，就会告诉你出现“系统找不到指定文件”的错误。



## 1.2.1 解决方法

### 1. 关于 MDM.ExE

MDM.exe 是 VS.NET 附带的一个服务程序，全名是 Mache Debug Manage,其描述为：“Manage Local and Remote Visual Studio Debug”,在控制面板下管理工具的服务中你可以找到它，你可以尝试将其关闭，但系统会报错，关闭不了，唯一的解决办法是先将其设置为禁用状态，然后再重启一下机器，这就一切 OK 了。

### 2. 如何卸载 VS.NET

也许你和我们一样持续尝试使用添加/删除程序或者原安装程序去卸载，却发现只用了几秒钟，程序就会报道你已经删除完毕，但是当你查找时，却哭笑不得的发现，程序仍然还在那儿，似乎这个程序删除不了，呵呵，那么你就跟我来吧，这是我摸索了一个小时才找出来的法宝。

打开开始菜单，在运行中输入下面一句话：“msiexec.exe /x {33334F07-0C6B-11D3-A51D-00A0C9231BA1} NOVSI=1”，然后回车，系统则会替你自动删除。我在写这篇文章的时候，系统正在删除着呢，十分痛快。

### 3. Config.Web 哪里去了

Config.Web 不再像原来一样，装在 win2000 的安装路径下的 complus 下面，而是找了一个新路径，即 [win2000path]microsoft.NET\framework\[version.name] 下面，其中 win2000path 即你安装 win2000 的路径，如我的为 C:\winnt,Version.name 是一个 V 打头的字符串，当前的版本号。

你可以在这儿配置你的 Config.Web

### 4. 关于常量 Null

VB 及 C#原来一直用 Null 来表示空值，但在 beta1 里面已经不使用了，开始用 DBNull 来代表。豆腐的关于跨站读别人网页的程序当中有几个错误，这是其中一条，另外，我感觉用 readline()方法似乎不好，不如改为 Readtoend(),一次全部读完，也不用我们自己判断哪儿结束了，另外，豆腐的文章当中涉及的编码问题有误，如果你用它的程序去读一个中文网页，得到的全是乱码，无论你是否改正了 Config.Web，都会出错。其实你只要把它程序中的那个 ASCII 改为 Default 就可以了，用网页的默认编码。呵呵，很简单吧？

本书的第三章将详细介绍 VS.NET 安装过程。

## 1.3 解答 .NET

### 1. 什么是 .NET 框架

.NET 框架是创建、部署和运行 Web 服务及其他应用程序的一个环境。它包括三个主要部分：公共语言运行时、框架类和 ASP.NET。

.NET 框架是否只适用于创建 Web 站点的用户？

.NET 框架使您可以创建优秀的 Web 应用程序。但它也可以帮助您创建现在的各种应用程序。与当前创建应用程序的方式相比，.NET 在编写任何 Windows 软件（使用 ATL/COM、MFC、Microsoft® Visual Basic® 或标准 Microsoft® Win32®）方面都具有更大的优势。当然，如果您是开发 Web 站点，那么从 ASP.NET 开始，您就会感受到 .NET 框架的强大吸引力。

### 2. .NET 框架可以运行于哪些平台

Beta 1 版可以运行在 Microsoft® Windows® 2000、Windows 95/98/ME 和 Windows NT® 4.0 上。另外，还有一个称为 .NET 精简框架的 .NET 框架版本。它用于使蜂窝电话和增强型电视等设备也具有 .NET 框架的某些功能。.NET 精简框架将运行在 Windows CE 和其他嵌入式操作系统上。

### 3. .NET 框架支持哪些编程语言

.NET 框架与编程语言无关。事实上任何语言都可以支持 .NET 框架。目前，您可以使用许多语言来创建 .NET 程序，包括：C++、Microsoft® Visual Basic.NET、JScript® 和 Microsoft 的最新语言——C#。以后，也会有大量的第三方语言可以用来创建 .NET 框架应用程序，包括 COBOL、Eiffel、Perl、Python、Smalltalk 等等。

### 4. .NET 框架和 COM+ 服务之间是什么关系

在 .NET 框架中，不仅可以完全访问 COM+ 服务，而且也更容易创建服务组件。

.NET 框架组件可被添加至 COM+ 应用程序中。在 COM+ 应用程序中，.NET 框架组件可以利用自动组件服务，例如：事务、对象池、排队组件、事件等等。

### 5. .NET 框架和 DCOM 之间是什么关系

DCOM 是用于进程间通讯的 COM 基础结构。.NET 框架支持大量用于进程间通讯的可插入通道和格式化程序。在托管代码和非托管代码之间进行转换时，.NET 框架使用了 COM 基础结构，尤其是 DCOM。所有使用 COM+ 服务的方案都使用了托管到非托管转换，因此默认使用 DCOM。对于注重互操作性的进程间通讯，.NET 框架也支持 SOAP（简单对象访问协议）。

## 6. .NET 框架是否仅仅是 Windows DNA 的新名称

不。Windows DNA 是用于创建紧耦合的分布式 Web 应用程序的一种体系结构。由于分布式应用程序变得需要更多的松耦合原理，因此 Microsoft 在 .NET 中发展了该体系结构。.NET 框架是 .NET 体系结构的一部分。

## 7. 什么是托管代码和托管数据

托管代码是编写为支持公共语言运行时服务的代码（请参阅“什么是公共语言运行时？”）。为了支持这些服务，代码必须向运行时提供最小级别的信息（元数据）。默认情况下，所有 C#、Visual Basic.NET 和 JScript.NET 代码都是托管代码。Visual Studio.NET C++ 代码在默认情况下不是托管代码，但通过指定命令行开关 (/CLR)，编译器也可以生成托管代码。

与托管代码密切相关的是托管数据。托管数据是由公共语言运行时的垃圾回收器进行分配和释放的数据。默认情况下，C#、Visual Basic 和 JScript.NET 数据是托管数据。不过，通过使用特殊的关键字，C# 数据可以被标记为非托管数据。Visual Studio.NET C++ 数据在默认情况下是非托管数据（即使在使用 /CLR 开关时），但是在使用 C++ 的托管扩展时，可以使用“\_\_gc”关键字将类标记为托管类。就象该名称所显示的那样，它表示类实例的内存由垃圾回收器管理。另外，该类也完全成为 .NET 框架的成员，同时具备它所带来的好处和限制。好处是：它可以与其他语言编写的类正确地进行互操作（如托管的 C++ 类可以从 Visual Basic 类继承）；限制是：托管类只能从一个基类继承。

## 8. 名称空间与程序集名称之间有什么区别

名称空间是类型的一种逻辑命名方案，其中简单类型名称（如 MyType）前面带有用点分隔的层次结构名称。这样的命名方案完全在开发人员的控制之下。例如，键入 MyCompany.FileAccess.A 和 MyCompany.FileAccess.B 在逻辑上将会具有与文件访问相关的功能。.NET 框架使用一种层次结构命名方案，用于将类型按相关功能的逻辑类别进行分组，例如，ASP.NET 应用程序框架或远程处理功能。设计工具可以利用名称空间使开发人员更容易在代码中浏览和引用类型。名称空间的概念与程序集的概念之间没有任何联系。一个程序集可以包含其层次结构名称具有不同名称空间根的类型，而一个逻辑名称空间根可以跨越多个程序集。在 .NET 框架中，名称空间是在设计时进行逻辑命名的便捷方式，而程序集在运行时为类型建立名称作用域。

## 9. 什么是垃圾回收

垃圾回收是使计算机能检测何时不再能够访问某个对象的一种机制。它将自动释放由该对象使用的内存（也调用用户编写的称为“结束者”的清理例程）。一些垃圾回收器（如

由 .NET 使用的) 会压缩内存, 并因此减少程序的工作集。

#### 10. 非确定性垃圾回收是如何影响代码的

对于大多数编程人员而言, 拥有一个垃圾回收器 (并且使用可作为垃圾回收的对象) 意味着永远不需要操心释放内存或引用计数对象, 即使您使用了复杂的数据结构。但如果您通常在同一个用于释放对象内存的代码块中释放系统资源 (文件句柄、锁定等等), 那么在编码样式方面需要做一些修改。使用可作为垃圾回收的对象时, 您应该提供一种方法, 来明确释放系统资源 (也就是说, 由您的程序控制), 同时允许垃圾回收器在压缩工作集时释放内存。

#### 11. 是否能够避免使用可作为垃圾回收的堆

所有支持运行时的语言都允许您从可作为垃圾回收的堆中分配类对象。这在快速分配方面带来了好处, 并且使编程人员无需自己来计算何时应该显式“free”每个对象。

CLR 还提供了 ValueTypes 对象——它们与类相似, 但 ValueType 对象是在运行时堆栈 (不是堆) 中分配的, 因此当您的代码退出定义这些对象的过程时, 将自动回收它们。这就是 C# 中“struct”的操作方式。

#### 12. 如何使代码与安全系统协调工作

通常, 这不成问题——大多数应用程序能安全地运行, 不会受恶意攻击的干扰。通过简单地使用标准类库来访问资源 (如文件) 或执行受保护的操作 (例如反转类型的私有成员), 安全性由这些库来实施。应用程序开发者需要完成的一项简单工作是包括权限请求 (一种公开的安全性), 将代码可能接收的权限限制在它所需要的权限范围内。这也确保了如果代码被允许运行, 它在运行时将具有所需的所有权限。

仅当开发人员需要编写提供新型资源的新基类库时, 他们才需要直接处理安全系统。在这种情况下, 并非所有的代码都有潜在的安全性问题, 代码访问安全机制将其限制在替代了安全系统的那部分代码上。

#### 13. 为什么在网络共享驱动器中运行代码时会发生安全异常

默认安全策略仅给来自本地 Intranet 区域的代码授予有限的权限。这个区域是由 Internet Explorer 安全设置定义的, 它们应该配置为与企业内部的本地网络相匹配。由于由 UNC 或映射驱动器 (例如使用 NET USE 命令) 命名的文件都需要在本地网络上发送, 因此它们也在本地 Intranet 区域中。

默认值是为不安全的 Intranet 这种最坏情况而设置的。如果您的 Intranet 比较安全, 您可以修改安全策略 (用 CASPol 工具), 给本地 Intranet 或其一部分 (例如特定的计算机共享名) 授予更多的权限。