

# Visual Basic .NET

## 编程指南

飞思科技产品研发中心 编著

Visual Basic.NET语言的新特性

基本的数据结构字符串和集合的使用

Visual Basic.NET语言的各种应用技巧

Visual Basic 6.0到Visual Basic.NET的完美升级

随书附赠光  
盘包含书中  
精彩范例的  
源代码



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

开发专家

之MS.NET

# Visual Basic .NET

## 编程指南

TP312BA

163

飞思科技产品研发中心 编著

本书配有光盘，需要的读者请到 <http://210.34.51.1/tractate/index.asp>  
网页上申请，或到“网络与光盘检索实验室”联系。

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书首先介绍了.NET及.NET框架下的Visual Basic.NET语言的新特性及其语法要点,接下来介绍了几种基本的数据结构字符串和集合的使用。然后介绍用Visual Basic.NET语言编写各类应用的一般步骤和若干技巧,包括控制台应用、类库、Windows窗体应用、Windows服务、Windows控件、Web窗体应用、Web服务器控件、Web服务、COM+应用、基于远程处理的分布式应用等。这部分穿插讲解了几个高级编程技术,如流和反射等。最后介绍了如何把Visual Basic 6.0应用的升级到Visual Basic.NET。随书附赠的光盘收录了书中经典实例的源代码。

本书内容新颖,实践性强,面向所有的Visual Basic.NET程序设计人员,也可供对.NET感兴趣的读者参考,是一本适用于Visual Basic老用户和初学者的经典图书。

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 Basic.NET 编程指南 / 飞思科技产品研发中心编著. —北京: 电子工业出版社, 2003.1  
(开发专家之 MS.NET)  
ISBN 7-5053-8109-1

I.V... II.飞... III.BASIC 语言—程序设计—指南 IV.TP312-62

中国版本图书馆 CIP 数据核字 (2002) 第 087079 号

责任编辑: 王树伟 王 蒙

印刷者: 北京民族印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 33.25 字数: 851.2 千字

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数: 4000 册 定价: 49.00 元 (含光盘)

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话 68279077

# 出版说明

“开发专家”是电子工业出版社计算机研发部长期以来精心培育的计算机科学技术类本品牌。这个品牌是由多个专题系列组成的横向大系列，涵盖了计算机技术的各个方面，特别是一直受到极大关注的程序开发类系列，例如《开发专家之数据库》、《开发专家之网络编程》、《开发专家之 Delphi》、《开发专家之 Sun ONE》，以及《开发专家之 MS.NET》等。这些专题系列基于各自的角度，从纵向上包含了该专题的所有内容。因此，整个“开发专家”的品牌架构纵横交错，囊括了所有的计算机技术和所有的技术层面，海纳百川而又极具可扩展性。

“开发专家”的作者队伍主要依托于“飞思科技产品研发中心”。“飞思科技产品研发中心”是由专业的策划人员、权威的技术专家和资深的作者队伍共同构成。在图书的出版上，形成了以研发为基础、以出版为中心、以服务为支持的专业化出版框架和流程。通过深入的市场调查和技术跟踪，在综合了技术需求和读者焦点等因素的基础上，形成各系列丛书的写作重点和大纲，然后聘请业界的最前沿学者进行写作。同时，策划工作全程介入写作进程，严格控制写作质量，用最专业的技术背景、最深刻的理论基础、最具代表性的案例、最能为专业读者接受的形式，为读者提供品质最佳的图书产品。这体现了出版者和著作者的完美结合。

多年来，计算机研发部始终把创造社会效益摆在首位，秉承一切为国内计算机技术专业读者服务的精神，为推动国内 IT 技术发展、为体现国内技术的原创水平，穷尽所有的创意与努力，将出版者的命运与读者的支持紧紧地连在了一起。

在此，我们临出版之残酷竞争而不惧，旌旗猎猎而异军突起，这与广大读者的支持是分不开的。为了使我们的脚步更坚实、使我们的队伍永葆活力和创造力，我们期待着您能为我们的前进贡献出您的意见和建议。同时，我们也在等待着您的加入。

我们的联系方式：

电 话：(010) 68134545    68131648

电子邮件：support@fecit.com.cn

飞思在线：<http://www.fecit.com.cn>    <http://www.fecit.net>

勘误网址：<http://www.fecit.com.cn/question.htm>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

电子工业出版社计算机研发部

# 前言

## 关于本丛书

对开发人员而言，.NET 是一个完美的开发平台。它提供了一套公共的运行库，并制定了一套公共语言规范，所有符合该规范的语言都可以无缝使用这套运行库。在.NET 平台下，除了语法上的区别，各种编程语言没有本质的不同。它们共享公共类库，具有类似的编程模型和相差无几的功能。开发人员可以自由选择自己喜爱的语言开发程序。.NET 平台提供大量的服务，包括垃圾自动收集、面向对象的多线程、基于程序集的部署、异常处理、特性编程、远程处理、ASP.NET 网页框架、互操作、安全性等，使开发人员可以快速构建各种应用，从传统的桌面应用到面向 Web 的大型分布式应用，应有尽有。.NET 将彻底改变软件的开发方式、使用方式和发行方式，.NET 将是一场软件革命。

Microsoft .NET 是一个用于构建、运行和体验下一代分布式应用程序的平台。它是跨客户端、跨服务器的开发人员工具，它由以下几部分组成：

- .NET 框架编程模型，它使开发人员能够构建 Web 应用程序、智能客户端应用程序和可扩展标记语言 (XML) Web 服务应用程序，它们利用诸如 SOAP、可扩展标记语言 (XML) 和 HTTP 等标准协议以编程方式通过网络开放其功能。
- 开发工具，比如 Visual Studio.NET，该工具为用 .NET 框架进行编程提供了一个迅速集成开发应用程序的环境。
- 一组服务器，包括 Windows 2000、SQL Server 和 BizTalkServer，它可集成、运行、操作和管理 XML Web 服务和应用程序。
- 客户端软件，比如 Windows XP、Windows CE 和 Office XP，它有助于开发人员跨多种设备和现有产品，为用户提供深切、引人注目的使用感受。

《开发专家之 MS.NET》系列丛书针对 Microsoft .NET 整个开发体系及其特色组织者进行了编写，它涵盖了 MS.NET 中的方方面面，为那些希望学习 MS.NET 技术的开发人员展示了非常完整的知识体系。在利用 MS.NET 进行项目开发的过程中，本丛书也可以为各种不同的需求提供完美的解答。

## 关于本书

Visual Basic.NET 是 Visual Basic 的最新版本。跟以往任何一个版本都不同，现在的 Visual Basic 是纯粹面向对象的编程语言，具有同 Visual C++ 及针对公共语言运行库的其他任何语言相同的变量类型、数组、用户定义的类型、类和接口。用它编写的代码可以与用其他语言（如 C# 或 C++）编写的代码充分互用，并使 Visual Basic.NET 开发人员能够只需借助 .NET 框架的功能，而不必求助于编程方法（而传统上要使 Windows API 工作，则必须使用编程方法）。这为 Visual Basic 赋予了全新的活力，Visual Basic 将继续成为最有效的 Windows 应用程序开发工具之一，而且现已成为创建下一代 Web 站点的最佳工具。

## 本书读者对象

本书面向所有的 Visual Basic.NET 程序设计人员，也可供对 .NET 感兴趣的读者参考。

## 本书内容

- .NET 及 .NET 框架下的 Visual Basic.NET 语言的新特性及语法要点。
- 几种基本的数据结构字符串和集合的使用。
- Visual Basic.NET 软件开发过程的大体步骤和经典经验技巧。
- 控制台应用、类库、Windows 窗体应用、Windows 服务、Windows 控件、Web 窗体应用、Web 服务器控件、Web 服务、COM+应用、基于远程处理的分布式应用等各种 Visual Basic.NET 开发的详细知识。
- 典型的高级编程技术，如流和反射等。

## 本书特点

跟同类书相比，本书有如下特色：

### 内容新颖

全书完全剖析了 Visual Basic.NET 的新特性。无论对 Visual Basic 的新手和老手，Visual Basic.NET 几乎都是新的。因为现在的 Visual Basic.NET 语言直接面向 .NET 平台。本书详细讨论了 Visual Basic.NET 中的各种新语法，如何用它来开发各种新型应用——包括控制台应用、Web 窗体应用、Web 服务、基于远程处理的分布式应用等。

### 实践性强

本书的作者都是第一批 Visual Basic.NET 的研究者和使用者。书中所述，均为有感而发。为方便读者学习，他们精选了有代表性的例子来说明每个难点，很多例子都来源于自己的实际工作。每个读者都将在本书中找到能在实践中需要的内容。

### 新老皆宜

虽然有 Visual Basic 经验的老手在阅读时可能体会更深，但本书仍然适用那些毫无 Visual Basic 编程经验的新手。只要熟悉了本书第 2 章的内容，新手也就完全具备了用 Visual Basic.NET 开发应用的实力。而本书最后一章则专门为老手而写，以帮助他们决定是否把应用升级到 Visual Basic.NET，并给出了升级的建议。

写作过程中参考了微软公司的相关资料，在此特做说明。

本书由飞思科技产品研发中心策划并组织编写，同时参与写作的还有张宏伟、黄涛、金莲、周叶、江黎明、何宏艳、何鹏、何彬、向蓉、曾冰、王国华、王守宏、田第员、江春明、刘强、刘力、王爱国、杨艳、胡瑛、周朝阳、邓守坤、田黎参与了材料收集工作。刘晓华、吴大雷参与了程序调试。杨学慧、李江、刘伟、潘志远参与了后期的技术审定工作。由于编者水平有限，再加上时间仓促，虽然经过再三勘误，但仍难免有纰漏之处，欢迎广大读者予以批评和指正。

我们的联系方式：

电 话：(010) 68134545 68131648

电子邮件：support@fecit.com.cn

飞思在线：<http://www.fecit.com.cn> <http://www.fecit.net>

答疑网址：<http://www.fecit.com.cn/question.htm>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

飞思科技产品研发中心

# 目 录

第 1 章 .NET 框架下的	
Visual Basic.NET.....	1
1.1 什么是.NET.....	1
1.2 什么是.NET 框架.....	4
1.3 Visual Basic.NET 编程语言	
的新特性 .....	4
1.3.1 继承 .....	5
1.3.2 异常处理 .....	6
1.3.3 重载 .....	7
1.3.4 构造函数和析构函数 .....	8
1.3.5 数据类型 .....	10
1.3.6 接口 .....	11
1.3.7 委托 .....	13
1.3.8 共享成员 .....	14
1.3.9 命名空间 .....	15
1.3.10 程序集 .....	16
1.3.11 引用 .....	17
1.3.12 特性 .....	18
1.3.13 多线程编程 .....	19
1.3.14 控制台应用 .....	20
1.4 Visual Basic.NET 的语言	
更新 .....	21
1.5 Visual Studio.NET 集成	
开发环境 .....	27
1.6 命令行编译器 .....	29
1.6.1 编译选项 .....	29
1.6.2 示例 .....	31
1.7 Visual Basic.NET 学习建议 .....	32
1.8 小结 .....	32
第 2 章 Visual Basic.NET 语言.....	33
2.1 选用 Visual Basic.NET 的	
理由 .....	33
2.2 Hello World.....	34
2.3 Visual Basic.NET 中的类型 .....	40
2.3.1 预定义类型 .....	43
2.3.2 转换 .....	45
2.3.3 数组类型 .....	48
2.3.4 通用类型 Object.....	51
2.3.5 模块.....	52
2.4 变量和参数.....	54
2.5 自动内存管理 .....	60
2.6 运算符和表达式.....	62
2.7 语句.....	65
2.7.1 选择语句.....	66
2.7.2 循环语句.....	69
2.7.3 跳转语句.....	74
2.7.4 On Error 语句 .....	79
2.7.5 With...End With 语句.....	81
2.7.6 SyncLock 语句.....	82
2.8 类.....	82
2.8.1 常数.....	85
2.8.2 字段.....	85
2.8.3 方法.....	87
2.8.4 属性.....	89
2.8.5 事件.....	94
2.8.6 实例构造函数.....	96
2.8.7 静态构造函数.....	100
2.8.8 继承.....	100
2.9 结构.....	103
2.10 接口.....	105
2.11 委托.....	106
2.12 枚举.....	108
2.14 特性.....	110
2.15 小结 .....	111
第 3 章 字符串.....	113
3.1 .NET 框架类库概述 .....	113
3.1.1 基本功能.....	113
3.1.2 程序集.....	114
3.2 命名空间.....	114
3.3 操作字符串 .....	116
3.3.1 创建新字符串.....	116
3.3.2 剪裁和移除字符.....	118
3.3.3 填充字符串 .....	120
3.3.4 比较字符串 .....	120

3.3.5	更改大小写 .....	123	6.3	序列化 .....	178
3.3.6	使用 StringBuilder 类 ..	124	6.4	在网络编程时使用流 .....	186
3.4	编码和解码字符 .....	126	6.5	小结 .....	194
3.5	小结 .....	127	<b>第 7 章</b>	<b>创建和使用类库 .....</b>	<b>195</b>
<b>第 4 章</b>	<b>集合 .....</b>	<b>129</b>	7.1	理解程序集 .....	195
4.1	基本概念 .....	129	7.1.1	基本功能 .....	195
4.1.1	.NET 框架类库中的集 合类提供的功能 .....	129	7.1.2	几种创建程序集的 方法 .....	196
4.1.2	.NET 框架类库中的 集合的类型 .....	130	7.1.3	程序集的优点 .....	196
4.2	运行库提供的集合类 .....	133	7.1.4	程序集的内容 .....	197
4.3	利用 .NET 框架类库中的 集合类操作集合 .....	135	7.1.5	程序集的清单 .....	199
4.3.1	选择一个集合类 .....	135	7.2	创建 .....	200
4.3.2	通过枚举数遍历集合 ..	136	7.2.1	Visual Studio.NET 开发 类库应用 .....	201
4.3.3	集合和同步 .....	139	7.2.2	设定类库的相关属性 ..	205
4.3.4	比较和排序 .....	142	7.3	使用类库 .....	206
4.4	集合拷贝 .....	144	7.4	创建和使用强名类库 .....	213
4.5	自定义集合 .....	146	7.4.1	强名称的基本概念 .....	214
4.5.1	Item 类 .....	147	7.4.2	创建强名类库 .....	214
4.5.2	IMyList .....	148	7.4.3	延迟签名 .....	219
4.5.3	ItemCollection .....	148	7.4.4	引用强名类库 .....	221
4.5.4	IEnumerator .....	150	7.5	共享类库 .....	226
4.5.5	测试函数 .....	151	7.6	利用共享程序集实现 透明引用 .....	229
4.6	小结 .....	152	7.7	小结 .....	232
<b>第 5 章</b>	<b>控制台应用程序 .....</b>	<b>153</b>	<b>第 8 章</b>	<b>开发窗体应用 .....</b>	<b>233</b>
5.1	概述 .....	153	8.1	第一个窗体应用 .....	233
5.2	开发范例 .....	154	8.2	设计窗体 .....	239
5.2.1	重定向 .....	155	8.2.1	设置窗体布局 .....	239
5.2.2	其他签名形式的 入口函数 .....	157	8.2.2	窗体前置 .....	241
5.2.3	添加窗体支持 .....	159	8.2.3	设置透明度 .....	241
5.2.4	在窗体应用中 应用控制台 .....	163	8.2.4	设置边框 .....	243
5.2.5	集成其他功能 .....	164	8.2.5	设置大小 .....	245
5.3	小结 .....	166	8.2.6	设置屏幕位置 .....	246
<b>第 6 章</b>	<b>流及其应用 .....</b>	<b>167</b>	8.2.7	设计窗体模式 .....	247
6.1	概述 .....	167	8.3	处理事件 .....	249
6.2	文件操作 .....	170	8.3.1	窗体中的事件 .....	250
			8.3.2	委托及其在事件处 理中的功能 .....	250



8.3.3	窗体事件的处理程序 ..	251	8.8.4	排列 MDI 子窗体 .....	283
8.3.4	在设计时创建事件 处理程序 .....	251	8.9	支持打印 .....	283
8.3.5	动态挂钩事件处理 .....	252	8.9.1	创建打印作业 .....	284
8.4	基于对话框的窗体应用 .....	254	8.9.2	在运行时更改 Windows 窗体打印选项 .....	284
8.4.1	创建对话框 .....	254	8.9.3	设置打印输出 .....	285
8.4.2	显示 .....	255	8.9.4	提示打印结束 .....	286
8.4.3	关闭对话框 .....	255	8.9.5	使用打印预览 .....	286
8.4.4	检索对话框的结果 .....	256	8.10	数据绑定 .....	287
8.4.5	使用 MessageBox .....	257	8.11	小结 .....	289
8.4.6	获取父窗体的信息 .....	258	<b>第 9 章 窗体高级编程 .....</b>	<b>291</b>	
8.4.7	使用标准对话框 .....	258	9.1	分割窗口 .....	291
8.4.8	通过可视化继承 创建对话框 .....	262	9.2	添加帮助 .....	292
8.5	使用菜单 .....	266	9.3	引发和处理周期性事件 .....	299
8.5.1	向窗体中添加主菜单 ..	266	9.4	绘制二维图形 .....	301
8.5.2	添加上下文菜单 .....	267	9.5	图像处理 .....	302
8.5.3	添加菜单项 .....	268	9.5.1	加载和显示图像 .....	302
8.5.4	添加子菜单 .....	270	9.5.2	克隆图像 .....	303
8.5.5	添加默认菜单项 .....	270	9.5.3	裁切和缩放图像 .....	304
8.5.6	增加检查标志 .....	271	9.6	本地化窗口 .....	305
8.5.7	克隆菜单项 .....	272	9.7	小结 .....	310
8.5.8	使用上下文菜单 .....	273	<b>第 10 章 窗体控件开发 .....</b>	<b>311</b>	
8.5.9	菜单合并 .....	273	10.1	概述 .....	311
8.6	拖放 .....	276	10.1.1	基本概念 .....	311
8.6.1	拖动数据 .....	277	10.1.2	开发窗体组件的 一般步骤 .....	313
8.6.2	放置数据 .....	277	10.2	开发用户控件 .....	320
8.6.3	在应用程序之间执 行拖放操作 .....	279	10.3	创建支持授权的控件 .....	324
8.7	操作剪贴板 .....	279	10.4	小结 .....	325
8.7.1	用 SetDataObject 方法 放置数据 .....	280	<b>第 11 章 开发窗体服务 .....</b>	<b>327</b>	
8.7.2	用 GetDataObject 方法 获取数据 .....	280	11.1	概述 .....	327
8.8	多文档界面应用程序 .....	280	11.2	创建和使用窗体服务 .....	330
8.8.1	创建 MDI 父窗体 .....	281	11.2.1	创建并配置服务 .....	330
8.8.2	创建 MDI 子窗体 .....	281	11.2.2	添加日志功能 .....	331
8.8.3	确定活动的 MDI 子窗口 .....	282	11.2.3	响应服务的启动 事件 .....	333
			11.2.4	处理服务停止事件 .....	333
			11.2.5	定义服务的其他 操作 .....	333

11.2.6	为服务创建安装程序 .....	334
11.2.7	生成服务项目 .....	335
11.2.8	为窗体服务创建安装项目 .....	335
11.2.9	利用 InstallUtil.exe 安装和卸载服务 .....	340
11.2.10	利用 ServiceController 组件控制服务 .....	341
11.2.11	调试窗体服务应用 .....	343
11.3	小结 .....	345
<b>第 12 章</b>	<b>ASP.NET Web 应用开发 .....</b>	<b>347</b>
12.1	ASP.NET 简介 .....	347
12.1.1	ASP.NET 页框架和 Web 窗体页 .....	347
12.1.2	ASP.NET 的功能 .....	348
12.2	Web 窗体页编程模型 .....	349
12.2.1	ASP.NET Page 类 .....	349
12.2.2	从 Page 类派生 .....	350
12.2.3	单文件 Web 窗体页 .....	351
12.3	Web 窗体页的生命周期 .....	352
12.3.1	往返行程 .....	352
12.3.2	重新创建页 (视图状态和状态管理) .....	353
12.3.3	事件驱动模型与线性处理模型的优点 .....	353
12.3.4	Web 窗体处理中的各个阶段 .....	354
12.4	创建简单的 Web 窗体应用 .....	355
12.4.1	创建项目和窗体 .....	356
12.4.2	添加控件和文本 .....	357
12.4.3	创建事件处理程序 .....	360
12.4.4	生成并运行 Web 窗体页 .....	360
12.5	处理服务器端的事件 .....	360
12.6	状态管理 .....	365
12.6.1	Web 窗体状态管理介绍 .....	365

12.6.2	基于客户端的状态管理 .....	365
12.6.3	基于服务器的状态管理选项 .....	368
12.6.4	使用视图状态保存 Web 窗体页值 .....	370
12.7	数据访问 .....	371
12.7.1	在 Web 窗体页中显示数据 .....	371
12.7.2	查询与更新 .....	374
12.8	小结 .....	380
<b>第 13 章</b>	<b>Web 窗体高级编程 .....</b>	<b>381</b>
13.1	缓存 .....	381
13.1.1	ASP.NET 页缓存 .....	381
13.1.2	片断缓存 .....	383
13.1.3	数据缓存 .....	385
13.2	跟踪 .....	386
13.2.1	启用页跟踪 .....	387
13.2.2	编写跟踪消息 .....	388
13.3	增强 ASP.NET Web 应用的安全性 .....	389
13.3.1	将 IIS 身份验证用于 ASP.NET 模拟 .....	390
13.3.2	使用窗体身份验证 .....	391
13.4	自定义 HTTP 处理器和处理模块 .....	395
13.5	小结 .....	400
<b>第 14 章</b>	<b>开发 Web 服务器控件 .....</b>	<b>401</b>
14.1	基本概念 .....	401
14.2	生命周期 .....	402
14.3	处理事件 .....	403
14.3.1	处理继承的事件 .....	403
14.3.2	捕获回发事件 .....	404
14.4	处理回发数据 .....	405
14.5	简单服务器控件开发实例 .....	406
14.6	定义复合属性 .....	410
14.7	检索内部文本 .....	412
14.8	开发复合控件 .....	413

14.8.1	简单的例子 .....	414	16.2.3	创建代理类的实例 .....	469
14.8.2	处理子控件的事件 .....	415	16.2.4	使用代理访问 Web 服务 .....	469
14.8.3	引发自定义事件 .....	417	16.3	状态管理 .....	471
14.9	状态维护 .....	419	16.3.1	使用 Session 对象 管理状态 .....	471
14.10	用客户端脚本引发回送 .....	420	16.3.2	使用 Application 对象管理状态 .....	473
14.11	模板控件开发 .....	423	16.3.3	利用数据库 .....	474
14.12	数据绑定控件 .....	426	16.4	提高 Web 服务性能 .....	476
14.13	小结 .....	430	16.4.1	少次多量代替多次 少量 .....	477
<b>第 15 章</b>	<b>远程处理和分布式应用 .....</b>	<b>431</b>	16.4.2	使用缓存 .....	479
15.1	理解分布式应用和远程 处理 .....	431	16.5	小结 .....	480
15.1.1	分布式应用简介 .....	431	<b>第 17 章</b>	<b>反射 .....</b>	<b>481</b>
15.1.2	.NET 的远程处理 框架 .....	433	17.1	基本概念 .....	481
15.1.3	基于远程处理的分布 式应用开发模型 .....	434	17.2	运行时查看类型信息 .....	482
15.2	一个基于远程处理的分布 式应用实例 .....	437	17.2.1	System.Type 和 ConstructorInfo .....	482
15.2.1	实现远程处理对象 .....	437	17.2.2	MemberInfo、 MethodInfo、FieldInfo 和 PropertyInfo .....	483
15.2.2	实现远程对象服务 器 .....	438	17.3	动态调用 .....	487
15.2.3	远程处理客户端 实现 .....	440	17.4	小结 .....	489
15.2.4	测试 .....	442	<b>第 18 章</b>	<b>开发 COM+ 应用 .....</b>	<b>491</b>
15.3	以 IIS 发布远程对象 .....	443	18.1	基本概念 .....	491
15.3.1	实例 .....	443	18.1.1	COM+ .....	491
15.4	分布式应用中的数据访问 .....	449	18.1.2	COM+ 服务 .....	491
15.5	小结 .....	452	18.1.3	COM+ 组件 .....	492
<b>第 16 章</b>	<b>Web 服务 .....</b>	<b>453</b>	18.2	COM+ 应用编程模型 .....	493
16.1	理解 Web 服务 .....	453	18.3	开发实例 .....	495
16.1.1	Web 服务的概念 .....	453	18.4	小结 .....	499
16.1.2	Web 服务代码模型 .....	454	<b>第 19 章</b>	<b>应用升级 .....</b>	<b>501</b>
16.1.3	Web 服务的部署 .....	460	19.1	决定是否升级 .....	501
16.1.4	Web 服务的发现 .....	463	19.1.1	不支持的功能 .....	502
16.2	访问 Web 服务的代码 模型 .....	466	19.1.2	确定需要的改 编量 .....	502
16.2.1	定位 Web 服务并 添加 Web 引用 .....	466	19.2	升级准备工作 .....	503
16.2.2	引用代理类 .....	468	19.2.1	一般性建议 .....	504

19.2.2	窗体和控件的 注意事项 .....	504	19.3.6	使用常量而非值 .....	509
19.2.3	数据建议 .....	505	19.3.7	在用户定义类型中避免 使用数组和定长串 .....	510
19.2.4	Web 结构建议 .....	505	19.3.8	避免使用旧式 关键字 .....	510
19.3	有关升级的语言建议 .....	505	19.3.9	调整用于 Win32 API 的数据类型 .....	511
19.3.1	使用早期绑定和显 式转换 .....	505	19.4	升级向导 .....	513
19.3.2	使用 Date 数据类 型存储日期 .....	507	19.4.1	用升级向导升级的 步骤 .....	513
19.3.3	解析无参数默认 属性 .....	507	19.4.2	升级实例 .....	514
19.3.4	避免空传播 .....	508	19.5	小结 .....	520
19.3.5	使用零界限数组 .....	509			

# 第 1 章 .NET 框架下的 Visual Basic.NET

跟以往的版本升级不同，Visual Basic 的最新版本——随同 Visual Studio.NET 一同发布的 Visual Basic.NET 跟前期的 Visual Basic 6.0 相比，有了本质的不同。对于传统的 Visual Basic 程序设计师而言，认识到这一点非常重要。而之所以会发生这些变化，其本质原因就是 Visual Basic.NET 语言是针对 .NET 平台设计的，它符合 CLS (Common Language Specification, 公共语言规范)，并充分利用了 CLR (Common Language Runtime, 公共语言运行时间) 提供的各种服务。

通过本章的学习，读者在 .NET 的整体框架中，了解 Visual Basic.NET 的新功能。下面是本章重点讲述的内容：

- .NET 组成及优势
- .NET 框架的构成
- Visual Basic.NET 语言的新特性
- Visual Basic.NET 语言的变化
- Visual Basic.NET 应用的集成开发环境
- Visual Basic.NET 的命令行编译器

## 1.1 什么是 .NET

Visual Basic.NET 是 .NET 平台下的开发语言，首先介绍什么是 .NET。如果说 Windows 系统是微软过去的标志的话，那么 .NET 就是微软未来的标志。对微软来说，.NET 的未来就是微软的未来。什么样的 .NET，让微软如此倾心呢？

按照微软的说法，.NET 是为简化在第三代 Internet 的高分布式环境下的应用程序开发，基于开放互联网标准和协议之上，实现不同语言和不同平台高度交互性，而构建的新一代计算和通信平台，也是微软以服务的方式递交软件的一种策略。

那么 .NET 到底有何优势呢？微软对此做出了下面的回答：

- 跨语言。.NET 支持多种语言的互操作，即在一种语言下开发的组件，可在另一组件下通过面向对象的继承而得以重用。
- 跨平台。.NET 通过将各语言先编译成中间语言 (IL)，然后在执行时用即时 (Just In Time) 编译器将之编译成本地平台代码来实现异构平台下对象的互操作。目前 .NET 的支持平台有 Windows, Linux 和 UNIX 的支持正在开发中。不仅如此，将来甚至还会出现各种支持 .NET 的智能终端。
- 安全。.NET 通过公共语言运行库来实现资源对象和类型的安全。
- 对开放互联网标准和协议的支持。.NET 通过对 HTTP、XML、SOAP、WSDL 等 Internet 标准的强劲支持提供在异构网络环境下获取远程服务、连接远程设备、

交互远程应用的编程界面。

了解了.NET 的定义及其优势后，下面讲述.NET 的组成。大致说来，.NET 主要由如下三部分构成：

- .NET 框架
- Web 服务
- .NET 企业服务器

这些部件一起提供了按照用户的需要创建 Web 的方法。图 1-1 展示了.NET 的部件构成。

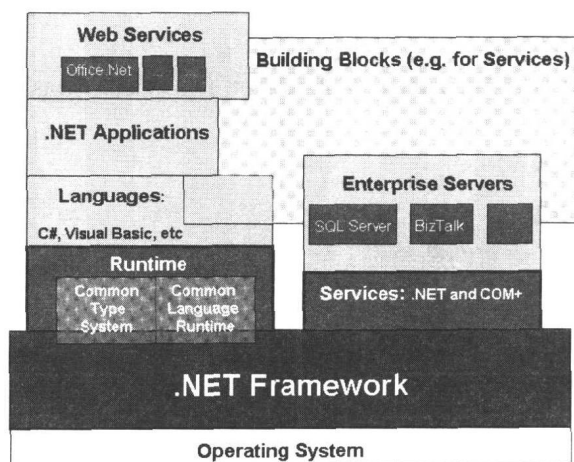


图 1-1 .NET 框架

接下来对图 1-1 做具体说明，以帮助大家进一步了解.NET 的结构。

## 1. 操作系统

微软宣称.NET 平台将独立于操作系统。按照微软的规划，除了 Windows 系列支持.NET 平台，未来的 Linux、UNIX 也将支持.NET 平台。

为了支持.NET 平台，只要操作系统能够识别微软中间代码（即 Microsoft Intermediate Language，简记为 MSIL）即可。因为.NET 所有语言编写的应用将编译为微软中间代码，然后在面向具体平台执行的时候，才会由一个 Just In Time（JIT）编译器转换成机器代码。

为了配合.NET 战略，微软即将推出 Windows.NET 操作系统。目前其测试版已经发布，估计明年中文正式版就会发布了。

## 2. .NET 框架

位于操作系统之上的.NET 是最重要的基础构架。它提供了创建、部署和运行.NET 应用的环境，其中的 Runtime 和 Services 实质上是.NET 框架的一部分。

如果有 Java 编程经验，那么可以把.NET 框架类比为 Java 虚拟机（JVM）。

## 3. 编程语言

.NET 平台的推出初始，让人兴奋的是微软号称无数的编程语言将得到支持，如 Visual

Basic、C#、Visual C++，甚至包括 Java、Pascal。但实际上.NET 仅支持一种叫做 MSIL 的语言。所谓支持一种语言，不过是相应地提供一个编译器将该语言撰写的源代码编译为 MSIL 代码。MSIL 代码是.NET 平台下的汇编语言，它包含一组与 CPU 无关的指令集，如加载、存储、初始化和调用对象方法等指令。

在跨语言编程中，MSIL 扮演了关键的角色。事实上，跨语言集成就是集成不同语言编写的源代码生成的 MSIL 代码。在微软的文档中，读者会看到诸如“跨语言派生”、“跨语言异常处理”等字眼，这时应该清楚，其实质就是从 MSIL 定义类派生，处理其他 MSIL 代码中引发的异常。

### 4. 企业服务

随着.NET 战略的推出，微软的大多数服务器产品都被打上了.NET 标记：

- BizTalk Server 2000
- SQL Server 2000
- Exchange 2000 Server
- Host Integration Server 2000
- Internet Security and Acceleration Server 2000
- Application Center 2000
- Commerce Server 2000
- Mobile Information 2001 Server
- SharePoint™ Portal Server 2001

### 5. 构造块服务 (Building Blocks Service)

它们是一些以用户为中心的 XML Web 服务集，它把对用户数据的控制权从应用程序转移给了用户。比较典型的构造块服务包括 Passport (用于用户标志) 和用于消息传递、文件存储、用户首选项管理、日历管理的服务，以及其他功能。

### 6. .NET 应用

.NET 平台下，开发人员可以开发多种.NET 应用。

- 控制台应用程序
- 脚本和宿主应用程序
- Windows Forms 应用程序 (Windows 桌面 GUI 应用程序)
- ASP.NET 应用程序
- Web Services 应用程序
- Windows 服务程序

### 7. Web Services

构建在.NET 平台上交付最终用户使用的产品如 Office.NET 等深刻体现了微软“软件就是服务”的思想，这样的 Web Services 又叫.NET 体验。

## 1.2 什么是.NET 框架

对.NET 应用开发者而言,处于核心位置的是.NET 框架。Visual Basic.NET 的一系列新的强大功能都是从.NET 框架中自动获得的。

在整个.NET 平台中,.NET 框架位于操作系统之上,为位于它之上的其他部件提供服务,从而实现下面的功能:

- (1) 提供一个一致的面向对象的编程环境,而无论对象代码是在本地存储和执行,还是在本地执行但在 Internet 上分布,或者是在远程执行的。
- (2) 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- (3) 提供一个保证代码(包括由未知的或不完全受信任的第三方创建的代码)安全执行的代码执行环境。
- (4) 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- (5) 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。
- (6) 按照工业标准生成所有通信,以确保基于 .NET 框架的代码可与任何其他代码集成。

了解了.NET 框架的重要作用,再来看看.NET 框架本身的结构。.NET 框架具有两个主要组件:公共语言运行库和 .NET 框架类库。公共语言运行库是 .NET 框架的基础。可以将运行库看做一个在执行时管理代码的代理,它提供核心服务(如内存管理、线程管理和远程处理),而且还强制实施严格的类型安全以及可确保安全性和可靠性的其他形式的代码准确性。事实上,代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码,而不以运行库为目标的代码称为非托管代码。

.NET 框架的另一个主要组件是基础类库,它是一个综合性的面向对象的 reusable 类型集合,利用它可以开发包含从传统的命令行或图形用户界面(GUI)应用程序到基于 ASP.NET 所提供的最新的应用程序(如 Web 窗体和 XML Web Services)在内的应用程序。

## 1.3 Visual Basic.NET 编程语言的新特性

本节是为传统的 Visual Basic 程序员而写的,如果读者没有 Visual Basic 编程经验,则可以直接跳过本节。对传统的 Visual Basic 程序员,一定要认识到 Visual Basic.NET 不是 Visual Basic 6.0 的简单升级,虽然它保留了部分 Visual Basic 的语法,但更主要的,它是面向.NET 平台的一种全新的编程语言。它位于.NET 运行库之上,并且直接利用.NET 的框架类库作为自己的运行库,因此,读者在.NET 框架内才能理解 Visual Basic.NET 的新特性。下面我们采取逐条列举的方式,对 Visual Basic.NET 的新特性做一说明。



在说明每个特性的时候,会给出具体的实例。为了实践这些例子,请读者用 Visual Studio.NET 集成开发环境建立一个 Visual Basic.NET 的控制台应用项目,然后用示例代码替换默认生成的代码,再编译运行。



### 1.3.1 继承

Visual Basic .NET 通过允许定义作为派生类基础的类支持继承。派生类继承并可以扩展基类的属性和方法，它们还可以用新实现重写继承方法。默认情况下，用 Visual Basic .NET 创建的所有类都是可继承的。由于设计的窗体实际上是类，因此可以使用继承在现有窗体的基础上定义新窗体。

下面的示例 1-1 是一个使用继承的简单例子。

示例 1-1

```
Class Class1
    Sub Method1()
        Console.WriteLine("基类的一个方法: Method1")
    End Sub
    'Overridable 关键字定义一个虚方法
    Overridable Sub Method2()
        Console.WriteLine("基类的另一个方法: Method2")
    End Sub
End Class

Class Class2
    Inherits Class1
    Public Field2 As Integer
    Overrides Sub Method2()
        Console.WriteLine("派生来重载的方法:Method2")
    End Sub
End Class

Class Test
    Shared Sub main()
        Dim C1 As New Class1()
        Dim C2 As New Class2()
        C1.Method1()
        C1.Method2()
        C2.Method1()
    End Sub
End Class
```

程序执行结果如图 1-2 所示。



```
基类的一个方法: Method1
基类的另一个方法: Method2
基类的一个方法: Method1
```

图 1-2 执行结果