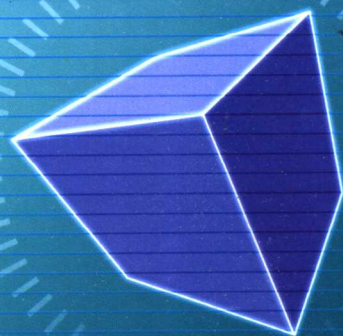
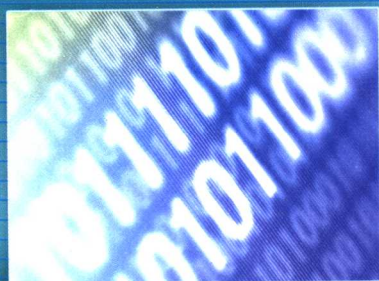




高等院校计算机技术系列教材

Visual Basic.NET 程序设计

喻德旷 杨 谊 李光明 谭小丹 编著



冶金工业出版社

高等院校计算机技术系列教材

Visual Basic.NET 程序设计

喻德旷 杨谊 李光明 谭小丹 编著

北 京

冶金工业出版社

内 容 简 介

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

Visual Basic.NET 以其易学易用、功能强大等特点而深受广大程序员的青睐。本书全面介绍了 Visual Basic.NET 编程的知识,从 Visual Basic.NET 的安装到详细的 Visual Basic.NET 的编程语法,从基本控件的使用到高级控件的使用,从菜单的建立到 Visual Basic.NET 面向对象的编程,从绘图到数据库编程,从 Web 应用程序的开发到 Web 服务器程序的开发,最后详细介绍了给 Visual Basic.NET 应用程序制作帮助以及部署发行。本书涉及的内容广泛,涵盖了 Visual Basic.NET 的基础编程和高级编程技术,是一本循序渐进掌握 Visual Basic.NET 编程的理想书籍。

本书主要面向 Visual Basic.NET 的初学者,可作为高等院校计算机相关专业的教材,也可供程序员从 Visual Basic 6.0 快速升级为 Visual Basic.NET 的学习参考用书。

图书在版编目 (CIP) 数据

Visual Basic.NET 程序设计 / 喻德旷等编著. —北京:
冶金工业出版社, 2005.12
ISBN 7-5024-3877-7

I. V... II. 喻... III. BASIC 语言—程序设计—高等
学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 139384 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

佛山市新粤中印刷有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2006 年 1 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 20.25 印张; 598 千字; 316 页

30.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于本书

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

目前,全国各地本科院校普遍扩招,本科生人数迅速增长,这给他们的就业带来了巨大的压力。而当前本科生的就业情况还不如专科学生,究其原因,所用教材与实际应用脱轨是一主要因素。针对现有教材质量较差、品种单一、版本陈旧、实用性和可操作性不强等原因,肩负着应用型人才培养的高等本科院校急需一系列符合当前教学改革需要的教材。

随着网络经济时代的到来,Microsoft 公司希望广大用户能随时随地都可以获得网络上的信息,并享受网络通信所带来的快乐。.NET 战略就是为了实现这一目标而设立的。

Visual Basic.NET 是 Microsoft 公司推出的 Visual Basic 语言新一代产品,它不是 Visual Basic 6.0 的简单升级,因为它具备了面向对象(OOP)编程语言的所有特征,包括类、接口、封装、继承和多态性等。另外还有一些其他特征,如结构化异常和多线程,使其成为一种强大的面向对象的程序设计语言。同时它基于 .NET Framework,是 Microsoft 公司综合提高安全性,更有效地进行内存管理和便捷地进行程序开发而推出的力作。

二、本书内容结构

全书共分 11 章,各章内容如下:

第 1 章: Visual Basic.NET 概述。主要介绍了 Visual Basic.NET 的新增功能、.NET 框架、Visual Basic.NET 的安装和卸载,以及其集成开发环境等,还介绍了 Visual Basic.NET 应用程序的基本开发步骤。

第 2 章: Visual Basic.NET 编程语法。主要介绍了数据类型、运算符和表达式、语句、流程控制、过程与常用内部函数等。

第 3 章: 标准控件和组件。主要介绍了在 Windows 窗体上放置控件、排列 Windows 窗体上的控件、系统标准控件等。

第 4 章: 菜单。主要介绍了下拉式菜单和弹出式菜单的制作和动态控制方法。

第 5 章: 对话框。主要介绍了能够显示系统的提示信息以及接收用户的输入信息的对话框,详细介绍了三类对话框的定义、创建方法和使用方法。

第 6 章: 面向对象编程。主要介绍了对象的基本概念、类的基本概念以及 .NET Framework 的相关知识。

第 7 章: 图形图像。主要介绍了绘图坐标系、颜色、绘图方法、绘图属性、如何绘制简单几何图形、图像的显示等,最后介绍了在 Visual Basic.NET 中如何进行 OpenGL 绘图。

第 8 章: 数据库编程。主要介绍了数据库基本概念、可视化数据库工具、ADO.NET、数据绑定、数据集更新、并发处理以及报表设计等。

第 9 章: Web 开发。主要介绍了在 Visual Basic.NET 中如何进行 Web 程序开发,包括: ASP.NET 基础知识、Web 窗体、Web 窗体中的数据访问以及 Web 服务,最后给出了一个集成了 Web 服务功能的订单查询系统实例。

第 10 章: 创建自定义控件。主要介绍了在 Visual Basic.NET 中如何创建自定义控件。

第 11 章: 帮助和部署。主要介绍了在开发完 Visual Basic.NET 程序后,如何为应用程序制作帮助系统以及如何将 Visual Basic.NET 应用程序部署到客户计算机中。

三、本书特点

本书系统、全面地研究和借鉴了国外相关教材先进的教学方法，结合国内院校教学实际和先进的教学成果，根据教育部“十一五”国家级规划教材应用型本科教育的指导思想编写，具有实用性和可操作性，与时俱进，与当前就业市场结合得更加紧密。

(1) 组织合理，内容全面。本书全面介绍了 Visual Basic.NET 中文版的编程知识，既介绍了 Visual Basic.NET 的安装、编程语法、基本控件、菜单、绘图等基本内容，也介绍了高级控件、数据库编程、Web 应用程序、Web 服务、自定义控件以及 Visual Basic.NET 应用程序制作帮助以及部署发行。内容全面，即使原来没有 Visual Basic 基础的读者阅读完本书也可全面掌握 Visual Basic.NET 的编程要领，也可让熟悉 Visual Basic 6.0 的程序员领略到 Visual Basic.NET 的编程精髓。

(2) 实例丰富，通俗易懂。本书在介绍 Visual Basic.NET 的编程知识点时，配备了丰富的实例，相关章还有综合实例演练，方便读者理解其内容。本书作为教材使用时，相关章的综合实例演练可作为学生的实验内容。

四、本书适用对象

本书主要面向 Visual Basic.NET 的初学者，可作为高等院校计算机相关专业的教材，也可供程序员从 Visual Basic 6.0 快速升级为 Visual Basic.NET 的学习参考用书。

本书主要由喻德旷、李光明、杨谊编写，谭小丹也为本书的写作付出了辛勤的劳动。由于编者水平有限，书中难免存在疏漏和不足之处，敬请读者批评指正，联系方法如下：

电子邮箱：service@cnbook.net

网址：www.cnbook.net

本书的电子教案、源代码及习题参考答案可在本网站下载中心免费下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者
2005 年 11 月

目 录

第 1 章 Visual Basic.NET 概述	1	2.2.5 逻辑运算符	39
1.1 Visual Studio.NET 中文版新增功能	1	2.2.6 表达式	39
1.2 .NET 框架	2	2.3 语句	40
1.2.1 .NET 的起源	2	2.4 流程控制	41
1.2.2 .NET Framework 简介	2	2.4.1 分支结构	41
1.3 Visual Basic.NET 中文版的安装和卸载	5	2.4.2 循环结构	43
1.3.1 Visual Basic.NET 中文版的安装	5	2.5 过程	45
1.3.2 Visual Basic.NET 中文版的卸载	7	2.5.1 Sub 过程	45
1.4 集成开发环境	7	2.5.2 Function 过程	47
1.4.1 启动和关闭 Visual Basic.NET 中文版 ...	7	2.5.3 Property 过程	48
1.4.2 项目类型	8	2.5.4 参数传递	48
1.4.3 主窗口及其组成	9	2.6 常用内部函数	49
1.4.4 解决方案资源管理器	11	2.6.1 数学函数	49
1.4.5 属性浏览器	13	2.6.2 转换函数	50
1.4.6 类视图	13	2.6.3 时间函数	53
1.4.7 Visual Basic.NET 集成环境的 新特色	14	2.6.4 字符处理函数	53
1.4.8 定制开发环境	15	小结	53
1.5 创建 Windows 应用程序	17	综合练习二	54
1.5.1 Windows 窗体和 Web 窗体	17	一、选择题	54
1.5.2 Windows 应用程序的开发步骤	18	二、填空题	54
小结	22	三、问答题	55
综合练习一	22	第 3 章 标准控件和组件	56
一、选择题	22	3.1 在 Windows 窗体上放置控件	56
二、填空题	23	3.1.1 向 Windows 窗体添加控件	56
三、问答题	23	3.1.2 向 Windows 窗体添加组件	57
第 2 章 Visual Basic.NET 编程语法	24	3.1.3 向 Windows 窗体添加 ActiveX 控件 ...	57
2.1 数据类型	24	3.2 排列 Windows 窗体上的控件	57
2.1.1 变量	25	3.2.1 在 Windows 窗体上停靠控件	58
2.1.2 常量和常数	27	3.2.2 在 Windows 窗体上对齐多个控件	58
2.1.3 复合数据类型	30	3.2.3 在 Windows 窗体之间复制控件	59
2.1.4 隐式和显式声明	35	3.2.4 在 Windows 窗体上锚定控件	59
2.2 运算符和表达式	35	3.2.5 在 Windows 窗体上定位控件	59
2.2.1 算术运算符	36	3.2.6 调整 Windows 窗体上控件的大小	60
2.2.2 赋值运算符	37	3.2.7 调整窗体上多个控件的大小	61
2.2.3 比较运算符	38	3.2.8 在 Windows 窗体上设置 Tab 键顺序 ...	61
2.2.4 串联运算符	38	3.3 系统标准控件	62
		3.3.1 Button 控件	62
		3.3.2 ToolBar 控件	63

3.3.3 TextBox 控件	65	4.3 菜单的有关操作	100
3.3.4 RichTextBox 控件	66	4.3.1 编辑 Windows 窗体的菜单项的 名称	100
3.3.5 Label 控件	68	4.3.2 移动 Windows 窗体菜单上的项	101
3.3.6 LinkLabel 控件	69	4.3.3 复制 Windows 窗体菜单上的项	101
3.3.7 StatusBar 控件	70	4.3.4 禁用 Windows 窗体菜单上的项	102
3.3.8 ListBox 控件、ComboBox 控件、 CheckedListBox 控件	70	4.3.5 隐藏 Windows 窗体上的菜单项	103
3.3.9 ListView 控件	72	4.3.6 删除 Windows 窗体菜单上的项	103
3.3.10 TreeView 控件	75	4.3.7 为 MDI 窗体创建“窗口”列表	104
3.3.11 PictureBox 控件	76	4.3.8 以编程方式合并菜单项	105
3.3.12 ImageList 组件	78	4.3.9 向 Windows 窗体添加菜单 增强功能	105
3.3.13 DomainUpDown 控件	79	4.4 Windows 窗体上的上下文菜单	107
3.3.14 NumericUpDown 控件	80	4.4.1 向 Windows 窗体添加上下文菜单	107
3.3.15 CheckBox 控件	80	4.4.2 向上下文菜单添加菜单项	108
3.3.16 RadioButton 控件和 GroupBox 控件 ..	81	4.5 向菜单项添加功能	108
3.3.17 TrackBar 控件	82	4.6 实例演练——创建动态上下文菜单	109
3.3.18 Panel 控件	83	小结	111
3.3.19 TabControl 控件	84	综合练习四	111
3.3.20 HScrollBar 控件和 VScrollBar 控件 ..	85	一、选择题	111
3.3.21 Timer 组件	86	二、填空题	111
3.3.22 ProgressBar 控件	87	三、问答题	111
3.3.23 OpenFileDialog 组件	88	第 5 章 对话框	112
3.3.24 SaveFileDialog 组件	89	5.1 消息对话框	112
3.3.25 FolderBrowserDialog 组件	90	5.1.1 对话框显示信息	112
3.3.26 FontDialog 组件	91	5.1.2 对话框接受用户输入的信息	114
3.3.27 ColorDialog 组件	92	5.2 预定义对话框	117
3.4 实例演练——计算器	92	5.3 自定义对话框	118
小结	96	5.3.1 创建自定义对话框	118
综合练习三	96	5.3.2 显示自定义对话框	118
一、选择题	96	5.3.3 自定义对话框中的用户输入	119
二、填空题	97	5.4 实例演练——使用对象整体检索 对话框信息	123
三、问答题	97	小结	125
第 4 章 菜单	98	综合练习五	125
4.1 菜单概述	98	一、选择题	125
4.2 向 Windows 窗体添加下拉式菜单和 子菜单项	98	二、填空题	125
4.2.1 在设计时向 Windows 窗体 添加菜单	99	三、问答题	125
4.2.2 以编程方式向 Windows 窗体 添加菜单	99	第 6 章 面向对象编程	126
4.2.3 在设计时向菜单添加菜单项	99	6.1 对象	126
4.2.4 以编程方式向菜单添加菜单项	100	6.1.1 对象概述	126

6.1.2 对象之间的关系.....	127	7.4.3 OpenGL 工作流程.....	177
6.1.3 创建和使用对象.....	128	7.4.4 创建 OpenGL 应用程序.....	178
6.2 类.....	133	7.4.5 OpenGL 建模.....	181
6.2.1 类的概述.....	134	7.4.6 OpenGL 变换.....	185
6.2.2 共享成员.....	136	7.4.7 OpenGL 光照.....	187
6.2.3 向类添加字段和属性.....	137	7.4.8 OpenGL 纹理.....	189
6.2.4 类方法.....	138	7.5 实例演练——绘制几何图形.....	191
6.2.5 重载属性和方法.....	139	小结.....	193
6.2.6 重写属性和方法.....	141	综合练习七.....	193
6.2.7 多态性.....	142	一、选择题.....	193
6.2.8 继承.....	144	二、填空题.....	194
6.2.9 使用接口.....	146	三、问答题.....	194
6.2.10 类与标准模块.....	146	第 8 章 数据库编程.....	195
6.3 .NET Framework.....	147	8.1 数据库基本概念.....	195
6.3.1 .NET Framework 概述.....	147	8.1.1 数据库概述.....	195
6.3.2 公共语言运行库.....	147	8.1.2 关系数据库设计.....	196
6.3.3 程序集.....	148	8.1.3 SQL 语言.....	200
6.3.4 应用程序域.....	149	8.1.4 存储过程.....	202
6.3.5 运行库宿主.....	149	8.2 可视化数据库工具.....	204
6.3.6 通用类型系统.....	150	8.2.1 使用服务器资源管理器.....	204
6.3.7 元数据和自描述组件.....	154	8.2.2 使用数据库设计器.....	206
6.3.8 语言互用性.....	154	8.2.3 使用查询和视图设计器.....	211
6.3.9 .NET Framework 类库.....	155	8.2.4 使用解决方案资源管理器.....	213
小结.....	157	8.3 ADO.NET.....	214
综合练习六.....	157	8.3.1 ADO.NET 概述.....	214
一、选择题.....	157	8.3.2 Connection 对象.....	216
二、填空题.....	158	8.3.3 Command 对象.....	217
三、问答题.....	158	8.3.4 DataReader 对象.....	218
第 7 章 图形图像.....	159	8.3.5 DataAdapter 对象.....	219
7.1 图形图像概述.....	159	8.3.6 DataSet 对象.....	220
7.1.1 光栅图形与矢量图形.....	159	8.4 数据绑定.....	224
7.1.2 坐标系.....	160	8.4.1 应用数据绑定控件访问数据库.....	224
7.1.3 颜色.....	161	8.4.2 复杂数据绑定控件.....	225
7.1.4 绘图方法的比较.....	163	8.5 数据集更新.....	226
7.2 GDI+图形.....	164	8.5.1 更新本地数据集.....	227
7.3 图像.....	169	8.5.2 添加数据.....	228
7.3.1 用 GDI+呈现图像.....	170	8.5.3 修改数据.....	229
7.3.2 定位和复制图像.....	170	8.5.4 删除数据.....	230
7.3.3 图像编辑器.....	171	8.6 并发处理.....	230
7.4 使用 OpenGL 绘图.....	175	8.6.1 并发控制.....	230
7.4.1 OpenGL 概述.....	176	8.6.2 处理并发错误.....	231
7.4.2 OpenGL 的组成.....	176	8.7 报表设计.....	232

8.7.1 Crystal Report Designer 的组成	233	10.1.2 从 UserControl 类继承	280
8.7.2 Crystal Reports 专家	235	10.1.3 从 Control 类继承	280
8.7.3 报表控件	235	10.2 从现有 Windows 窗体控件继承	281
8.7.4 设计并运行新报表	236	10.3 从 UserControl 类继承	282
8.8 实例演练——销售人员管理系统	238	10.4 自定义控件的属性、方法和事件	282
小结	248	10.4.1 公开构成控件的属性	282
综合练习八	248	10.4.2 自定义控件中的方法实现	283
一、选择题	248	10.5 自定义控件的绘制和呈现	283
二、填空题	249	10.5.1 用户描述的控件	283
三、问答题	249	10.5.2 构成控件	284
第 9 章 Web 开发	250	10.5.3 为控件提供工具箱位图	284
9.1 ASP.NET 概述	250	10.6 实例演练——制作闹钟	285
9.2 Web 窗体	250	小结	293
9.2.1 Web 窗体简介	250	综合练习十	294
9.2.2 创建基本的 Web 窗体	251	一、选择题	294
9.2.3 验证 Web 窗体的数据输入	254	二、填空题	294
9.3 Web 窗体的数据访问	259	三、问答题	294
9.3.1 Web 窗体数据访问简介	259	第 11 章 帮助与部署	295
9.3.2 Web 数据访问策略	260	11.1 工具提示帮助制作	295
9.3.3 在 Web 窗体中通过 DataGrid 控件		11.2 “这是什么”帮助	296
显示数据	261	11.3 HTML 帮助	298
9.3.4 在 Web 窗体中创建只读数据访问	264	11.3.1 HTML 帮助系统概述	298
9.3.5 在 Web 窗体中更新数据	265	11.3.2 创建帮助项目文件	299
9.4 Web 服务	268	11.3.3 创建目录文件	301
9.4.1 Web 服务简介	268	11.3.4 创建索引文件	303
9.4.2 创建 Web 服务	270	11.3.5 编译帮助工程	304
9.4.3 使用 Web 服务	271	11.3.6 将帮助系统与应用程序相连接	305
9.5 实例演练——订单查询系统	273	11.4 部署	305
小结	279	11.4.1 Visual Basic.NET 新的部署功能	306
综合练习九	279	11.4.2 Visual Basic.NET 的安装编辑器	306
一、选择题	279	11.5 实例演练——应用程序的部署	310
二、填空题	279	小结	315
三、问答题	279	综合练习十一	315
第 10 章 创建自定义控件	280	一、选择题	315
10.1 自定义控件的类型	280	二、填空题	316
10.1.1 从 Windows 窗体控件继承	280	三、问答题	316

第 1 章 Visual Basic.NET 概述

教学目标

Visual Basic.NET 是 Microsoft 公司推出的 Visual Basic 的最新版本,在 Visual Basic 6.0 简单易用的基础上做了很大的改进,功能更加强大,获得了过去需要用 C++ 等高级语言才具有的访问系统资源的能力,并且增加了面向对象编程的核心机制。Visual Basic.NET 与 Web 窗体和 Web 服务的结合,使得用户可以开发出基于网络的应用程序。

- (1) Visual Studio.NET 中文版新增功能。
- (2) .NET 框架的组成和内容。
- (3) Visual Basic.NET 中文版的安装和卸载方法。
- (4) 集成开发环境。

1.1 Visual Studio.NET 中文版新增功能

Visual Studio.NET 是一套完整的开发工具,用于生成 ASP Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic.NET、Visual C++.NET 和 Visual C#.NET 全都使用相同的集成开发环境 (IDE),该环境允许它们共享工具并有助于创建混合语言解决方案。另外,这些语言利用了 .NET 框架的功能,它提供对简化 ASP Web 应用程序和 XML Web Services 开发的关键技术的访问。

在 Visual Basic 和 Visual C# 中的新增功能如下:

1. Web 开发中的新增内容

Visual Basic.NET 包括对 Web 窗体和 XML Web Services 的支持。Web 窗体能够方便快捷地为 ASP.NET Web 应用程序创建基于浏览器的界面。XML Web Services 能够将 Visual Basic 或 C# 方法打包,使其可以通过 Web 被访问。

2. 数据中的新增功能

ADO.NET 为所有基于 .NET 框架的 Visual Studio 应用程序提供可伸缩、高性能的数据访问。可以在数据源中直接访问数据,也可以创建内存缓存 (即数据集) 来处理已断开链接的数据,还可以将 XML 作为关系数据直接使用,或者直接创建并编辑 XML 文件。

3. Windows 窗体和控件中的新增功能

可以使用 Windows 窗体和控件来创建分布式应用程序的表示层。Windows 窗体提供一个有条理的、面向对象的、可扩展的类集,能够开发丰富的 Windows 应用程序。

4. 项目中的新增内容

使用项目模板,可以方便地创建各种 Windows 和 ASP.NET Web 应用程序和控件。项目模板将为项目设置必需的引用,并将命名空间和默认项导入到项目。

5. 组件和组件创作的新增功能

可以使用非可视组件和相关功能将资源 (例如消息队列、事件日志和性能计数器) 合并到应用程序中,还可以通过组件设计器类和框架类为创建组件提供 RAD 支持。

6. 新增的调试功能

可以对同一解决方案中不同语言编写的项目进行调试,执行远程调试,以及在全部代码中实现跟踪,检索经过微调的输出。

7. 扩展性和自动化中的新功能

Visual Studio.NET 包含一个可编程的对象模型,利用此模型可访问集成开发环境 (IDE) 的基础组件和事件,能够扩展 IDE 的功能,将重复性任务自动化,以及将 IDE 与其他应用程序集成。

8. 国际应用程序中的新增功能

Visual Basic.NET 中的国际化功能建立在 .NET 框架的全球化和本地化功能基础上，可以很容易地创建面向多个区域设置的应用程序。

1.2 .NET 框架

1.2.1 .NET 的起源

Visual Basic.NET 是基于 Visual Basic 的，而 Visual Basic 是从 BASIC 发展起来的。BASIC 是 Beginner's All-Purpose Symbolic Instruction Code（初学者通用符号指令代码）的缩写，这是一种在计算机发展历史上应用得十分广泛的语言。Visual Basic 在 BASIC 的基础上进行了扩充和发展，提供了方便的图形用户界面设计工具和机制，是开发 Windows 应用程序的简单而功能强大的方法。Visual Basic 也随着用户使用范围的扩大和需求的复杂化而不断地进行着改进和扩充。例如，当对支持数据库的开发工具的需求出现后，Visual Basic 通过将 Microsoft Access 数据库引擎与大量的数据控件相结合解决了这一需求。使用数据访问对象 DAO 实现了在客户/服务器环境中连接数据源的需求。随着 32 位操作系统的应用，Visual Basic 也出现了 32 位版本。在 Visual Basic 的 5.0 和 6.0 版本中，提供了在松散配合的 Internet 环境中编程的机制，能够实现网络应用程序的创建，并创建出高效的以 Internet 为目标的 Microsoft ActiveX 控件。

Visual Basic.NET 将 Visual Basic 的性能和效率又向前推进了一大步。通过真正的面向对象的特性，如继承和封装，使 Visual Basic 的程序设计变得更加优秀和更加易于维护。可以直接利用 Microsoft 丰富的平台来创建基于 Windows 的应用程序、XML 网络服务和移动软件。

1.2.2 .NET Framework 简介

.NET Framework 是一种新的计算平台，它简化了在高度分布式 Internet 环境中的应用程序开发。.NET Framework 旨在实现下列目标：

(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。

(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。

.NET Framework 具有两个主要组件：公共语言运行库和 .NET Framework 类库。公共语言运行库是 .NET Framework 的基础。可以将运行库看作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 Framework 的另一个主要组件是类库，它是一个综合性的面向对象的 reusable 类型集合，可以使用它开发多种应用程序，这些应用程序包括传统的命令行或图形用户界面（GUI）应用程序，也包括基于 ASP.NET 所提供的最新的应用程序（如 Web 窗体和 XML Web Services）。

.NET Framework 可由非托管组件承载，这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行，从而创建一个可以同时利用托管和非托管功能的软件环境。.NET Framework 不但提供若干个运行库宿主，而且还支持第三方运行库宿主的开发。

Internet Explorer 是承载运行库（以 MIME 类型扩展的形式）的非托管应用程序的一个示例。使用 Internet Explorer 承载运行库使能够在 HTML 文档中嵌入托管组件或 Windows 窗体控件。以这种方式承载运行库使得托管移动代码成为可能，但是它具有只有托管代码才能提供的重大改进。

图 1-1 显示了公共语言运行库和类库与应用程序之间以及与整个系统之间的关系, 以及托管代码如何在更大的结构内运行。

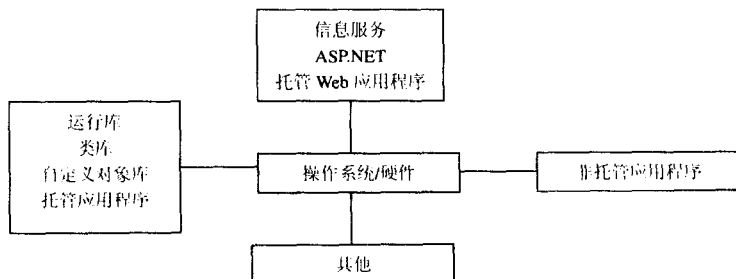


图 1-1 .NET Framework 环境

1. 公共语言运行库的功能

公共语言运行库管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务。这些功能是在公共语言运行库上运行的托管代码所固有的。

运行库强制实施代码访问安全。例如, 用户可以相信嵌入在 Web 页中的可执行文件能够在屏幕上播放动画或唱歌, 但不能访问他们的个人数据、文件系统或网络。这样, 运行库的安全性功能就使通过 Internet 部署的合法软件能够具有特别丰富的功能。

运行库还通过实现称为通用类型系统 (CTS) 的严格类型验证和代码验证基础结构来加强代码的可靠性。CTS 确保所有托管代码都是可以自我描述的。各种 Microsoft 和第三方语言编译器生成符合 CTS 的托管代码, 这意味着托管代码可在严格实施类型验证和类型安全的同时使用其他托管类型和实例。

此外, 运行库的托管环境还消除了许多常见的软件问题。例如, 运行库自动处理对象布局并管理对对象的引用, 在不再使用它们时将它们释放。这种自动内存管理解决了两个最常见的应用程序错误: 内存泄漏和无效内存引用。

运行库还提高了开发人员的工作效率。例如, 程序员可以选择开发语言编写应用程序, 也可以充分利用其他开发人员用其他语言编写的运行库、类库和组件。任何选择以运行库为目标的编译器供应商都可以这样做。以 .NET Framework 为目标的语言编译器使得用该语言编写的现有代码可以使用 .NET Framework 的功能, 这大大减轻了现有应用程序的移植过程的工作负担。

尽管运行库是为未来的软件设计的, 但是它也支持现在和以前的软件。托管和非托管代码之间的互操作性使开发人员能够继续使用所需的 COM 组件和 DLL。

运行库旨在增强性能。尽管公共语言运行库提供许多标准运行库服务, 但是它从不解释托管代码。一种称为实时 (JIT) 编译的功能使所有托管代码能够以它在其上执行的系统的本机语言运行。同时, 内存管理器排除了出现零碎内存的可能性, 并增大了内存引用区域以进一步提高性能。

最后, 运行库可由高性能的服务器端应用程序 (如 Microsoft SQL Server 和 Internet 信息服务 (IIS)) 承载。此基础结构在享受支持运行库宿主的行业最佳企业服务器的优越性能的同时, 能够使用托管代码编写业务逻辑。

2. .NET Framework 类库

.NET Framework 类库是一个与公共语言运行库紧密集成的可重用的类型集合。该类库是面向对象的, 并提供自己的托管代码可从中导出功能的类型。这不但使 .NET Framework 类型易于使用, 而且还减少了学习 .NET Framework 的新功能所需要的时间。此外, 第三方组件可与 .NET Framework 中的类无缝集成。

.NET Framework 类型使能够完成一系列常见编程任务 (包括诸如字符串管理、数据收集、数据库连接以及文件访问等任务)。除这些常见任务之外, 类库还包括支持多种专用开发方案的类型。可使

用 .NET Framework 开发控制台应用程序、Windows GUI 应用程序（Windows 窗体）、ASP.NET 应用程序、XML Web Services、Windows 服务等类型的应用程序和服务。例如，Windows 窗体类是一组综合性的可重用的类型，它们大大简化了 Windows GUI 的开发。如果要编写 ASP.NET Web 窗体应用程序，可使用 Web 窗体类。

3. 客户端应用程序开发

客户端应用程序在基于 Windows 的编程中最接近于传统风格的应用程序。客户端应用程序包括诸如字处理程序和电子表格等应用程序，还包括自定义的业务应用程序（如数据输入工具、报告工具等等）。客户端应用程序通常使用窗口、菜单、按钮和其他 GUI 元素，并且它们可能访问本地资源（如文件系统）和外围设备（如打印机）。

另一种客户端应用程序是作为 Web 页通过 Internet 部署的传统 ActiveX 控件（现在被托管 Windows 窗体控件所替代）。此应用程序非常类似于其他客户端应用程序，它在本机执行，可以访问本地资源，并包含图形元素。

过去，开发人员将 C/C++ 与 Microsoft 基础类（MFC）或应用程序快速开发（RAD）环境（如 Microsoft Visual Basic）一起使用来创建这样的应用程序。.NET Framework 将这些现有产品的特点合并到了单个且一致的开发环境中，该环境大大简化了客户端应用程序的开发。

包含在 .NET Framework 中的 Windows 窗体类旨在用于 GUI 开发。可以轻松创建具有适应多变的商业需求所需的灵活性的命令窗口、按钮、菜单、工具栏和其他屏幕元素。

例如，.NET Framework 提供简单的属性以调整与窗体相关联的可视属性。某些情况下，基础操作系统不支持直接更改这些属性，而在这些情况下，.NET Framework 将自动重新创建窗体。这是 .NET Framework 集成开发人员接口从而使编码更简单、更一致的许多方法之一。

和 ActiveX 控件不同，Windows 窗体控件具有对用户计算机的不完全受信任的访问权限。这意味着二进制代码或在本机执行的代码可访问用户系统上的某些资源，例如 GUI 元素和访问受限制的文件，但这些代码不能访问或危害其他资源。由于具有代码访问安全性，许多曾经需要安装在用户系统上的应用程序现在可以通过 Web 安全地部署，这些应用程序可以在像 Web 页那样部署时实现本地应用程序的功能。

4. 服务器应用程序开发

在托管领域中，服务器端应用程序是通过运行库宿主实现的。非托管应用程序承载公共语言运行库，后者使得自定义托管代码可以控制服务器的行为。此模型在获得主服务器的性能和可伸缩性的同时提供给公共语言运行库和类库的所有功能。

图 1-2 显示了在不同服务器环境中运行托管代码的基本网络架构。在应用程序逻辑通过托管代码执行时，服务器（如 IIS 和 SQL Server）可执行标准操作。

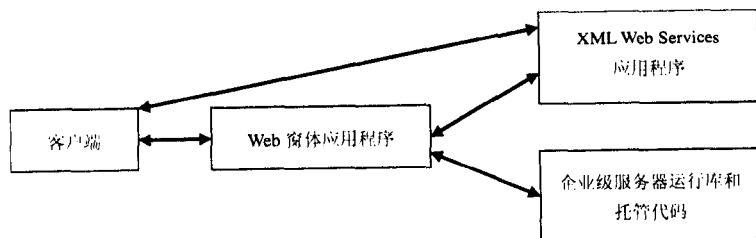


图 1-2 服务器端托管代码

ASP.NET 是使开发人员能够使用 .NET Framework 开发基于 Web 的应用程序的宿主环境。但是，ASP.NET 不仅是一个运行库宿主；它还是使用托管代码开发 Web 站点和通过 Internet 分布的对象的完整结构。Web 窗体 and XML Web Services 都将 IIS 和 ASP.NET 作为应用程序的发布机制，并且两者在 .NET Framework 中都具有支持类集合。

XML Web Services 作为基于 Web 的技术的重要发展,是类似于常见 Web 站点的分布式服务器端的应用程序组件。但是,与基于 Web 的应用程序不同,XML Web Services 组件不具有 UI 并且不以浏览器(如 Internet Explorer 和 Netscape Navigator)为目标。XML Web Services 由旨在供其他应用程序使用的可重用的软件组件组成,所谓的其他应用程序包括:传统的客户端应用程序、基于 Web 的应用程序,甚至是其他 XML Web Services。因此,XML Web Services 技术正迅速地将应用程序开发和部署推向高度分布式 Internet 环境。

.NET Framework 还提供类和工具的集合来帮助开发和使用 XML Web Services 应用程序。XML Web Services 是基于 SOAP(一种远程过程调用协议)、XML(一种可扩展的数据格式)和 WSDL(Web 服务描述语言)这些标准生成的。基于这些标准生成 .NET Framework 的目的是为了提高与非 Microsoft 解决方案的互操作性。

例如,.NET Framework SDK 所包含的 Web 服务描述语言工具可以查询在 Web 上发布的 XML Web Services,分析它的 WSDL 描述,并产生 C#或 Visual Basic 源代码,应用程序可以使用这些代码而成为 XML Web Services 的客户端。这些源代码可以创建从类库中派生的类,这些类使用 SOAP 和 XML 分析处理所有基础通信。虽然可以使用类库来直接使用 XML Web Services,Web 服务描述语言工具和包含在 SDK 中的其他工具可以使其更加方便地用 .NET Framework 进行开发。

如果开发和发布自己的 XML Web Services,.NET Framework 提供了一组符合所有基础通信标准(如 SOAP、WSDL 和 XML)的类。能够将注意力集中在服务的逻辑上,而无需关注分布式软件开发所需要的通信基础结构。

Microsoft Visual Basic、Microsoft C++和 Microsoft JScript 均已更新,以满足用户的开发需要。另外还引入了一种新语言 Microsoft C#。这些语言利用了 .NET 框架的功能,简化了 ASP Web 应用程序和 XML Web Services 的开发。

.NET 框架是用于生成、部署和运行 XML Web Services 和应用程序的多语言环境,为开发人员提供了统一的、面向对象的、分层的和可扩展的类库集(API)。目前,C++开发人员使用 Microsoft 基础类,而 Java 开发人员使用 Windows 基础类。框架统一了这些完全不同的模型并且为 Visual Basic 和 JScript 程序员同样提供了对类库的访问。通过创建跨所有编程语言的公共 API 集,公共语言运行库使得跨语言继承、错误处理和调试成为可能。从 JScript 到 C++的所有编程语言具有对框架的相似访问,开发人员可以自由选择他们要使用的语言。

1.3 Visual Basic.NET 中文版的安装和卸载

1.3.1 Visual Basic.NET 中文版的安装

1. 系统要求

安装 Visual Studio.NET 企业结构设计版的计算机应满足如表 1-1 中所列举的系统要求。

表 1-1 安装 Visual Studio.NET 不同版本的计算机系统要求

硬件	要求
处理器	具有 Pentium II 级处理器的计算机, 450 MHz 建议: Pentium III 级, 600MHz
内存	Windows 2000 Professional——96 MB; Windows 2000 Server——192 MB; Windows NT 4.0 Workstation——64 MB; Windows NT 4.0 Server——160 MB; Windows XP Professional——160 MB 建议: Windows 2000 Professional——128 MB; Windows 2000 Server——256 MB; Windows NT 4.0 Workstation——96 MB; Windows NT 4.0 Server——192 MB; Windows XP Professional——192 MB

续表 1-1

硬件	要求
可用硬盘空间	系统驱动器上有 500 MB，安装驱动器上有 3 GB
操作系统	Windows® 2000、Windows XP、Windows NT 4.03 或更高版本
CD-ROM 驱动器或 DVD-ROM	必选
鼠标	Microsoft 鼠标或兼容的指定设备

Windows NT 4.0 不支持 ASP.NET、COM+或多处理器垃圾回收，这些功能仅在 Windows 2000 和 Windows XP 上提供。

2. 安装过程

在安装 Visual Studio.NET 之前，验证是否满足系统要求，并按照下面的步骤安装 Visual Studio.NET。

1) 卸载 Visual Studio.NET 的预发布版本

如果计算机上曾经安装过 .NET 框架或 Visual Studio.NET 的试用版，可以按照以下步骤来卸载以前安装的版本。

从“开始”菜单中选择“设置”，然后选择“控制面板”。选择“添加/删除程序”，然后选择“更改或删除程序”，删除掉“Visual Studio.NET”和“Microsoft.NET 框架”。

2) 卸载已部署的应用程序

如果已使用 Visual Studio.NET 创建了应用程序，并使用 Visual Studio.NET 部署功能对这些应用程序进行了部署，那么必须在安装 Visual Studio.NET 之前卸载这些应用程序。卸载已部署的应用程序的方法是使用“控制面板”中的“添加/删除程序”功能。如果不卸载已部署的应用程序，Visual Studio.NET 安装将不能继续进行。

3) 安装 Visual Studio.NET

关闭所有打开的应用程序，以防止在安装过程中需要进行系统的重新启动。

插入标为 Visual Studio.NET CD1 或 DVD 光盘。自动运行会启动 Setup.exe。如果自动运行没有启动的话，则从安装 CD 或 DVD 的根目录下直接运行 Setup.exe。安装界面如图 1-3 所示。

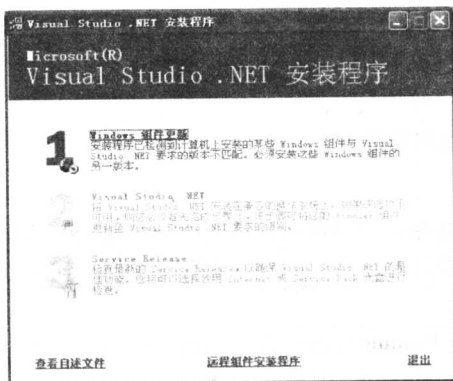


图 1-3 Visual Basic.NET 安装界面

该安装程序对磁盘上已安装的组件进行扫描。如果安装程序确定系统需要进行组件更新，则在“安装”对话框中会出现“步骤 1：安装 Windows 组件更新”，选择步骤 1 来更新系统组件。如果不需要进行组件更新，则该选项会显示为灰色，此时可以直接进入到步骤 2。在安装程序确认系统已经包含有最新的系统组件之后，“安装”对话框将会启用“步骤 2：安装 Visual Studio.NET”。选择步骤 2 来安装 Visual Studio.NET。

3. 为 ASP.NET 项目配置远程 Web 服务器

若要在 Visual Studio.NET 中处理 Web 应用程序，则必须正确配置开发环境和 Web 服务器。

要求 Web 服务器必须运行 Windows 2000、Windows XP 或 Windows 2003 Server，不支持运行 Windows 95、Windows 98、Windows ME 或 Windows NT 4.0 的计算机。

建议将 Web 服务器安装在使用 NTFS 文件系统格式化的计算机上。虽然 Visual Studio.NET 支持 FAT 和 FAT32，但这两种文件系统不如 NTFS 安全。

Web 服务器必须运行 Internet 信息服务 IIS 5.0 或 IIS 6.0 版。有关详细信息，请参阅 Visual Studio.NET CD1 或 DVD 上的 Setup\Web Server.htm。

为 Visual Studio.NET 配置单独的 Web 服务器计算机的方法是：在 Web 服务器计算机上运行 Visual Studio.NET 安装程序并执行步骤 1 以安装“Windows 组件更新”。这将在服务器计算机上安装 .NET 框架。在完成这一步骤后，安装进程将返回到 Visual Studio.NET 安装的步骤 2。开始安装进程的步骤 2，安装窗口的左窗格中有若干安装选项。除“服务器组件”之外，清除对所有选项的选择。展开“服务器组件”节点，并选择“Web 开发”和“远程调试器”，单击“立即安装”，继续安装。最后将要创建 Web 项目的用户添加到“VS Developer”组中。

1.3.2 Visual Basic.NET 中文版的卸载

要卸载 Visual Studio.NET 中文版，进入控制面板，打开“添加或删除程序”对话框，如图 1-4 所示，选择“Microsoft Visual Studio.NET 企业级结构设计版 2003 - 中文版”项，然后单击“更改/删除”按钮。

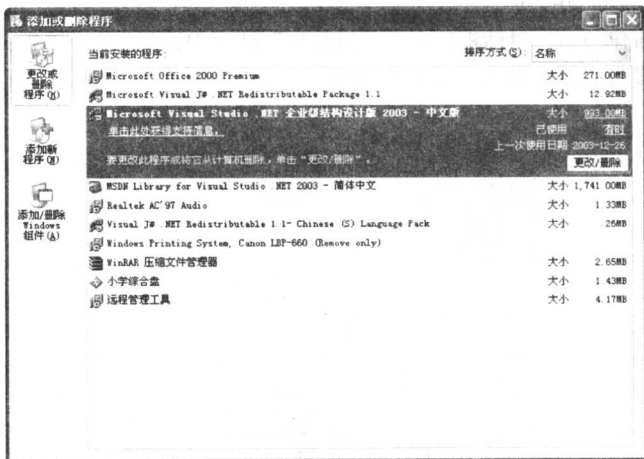


图 1-4 卸载 Visual Basic.NET

卸载 Visual Studio.NET 不会卸载由“Windows 组件更新”安装的组件。对于提供单独卸载的组件，要使用到“添加/删除程序”。

1.4 集成开发环境

Visual Basic、Visual C++、Visual C#和 Microsoft Developer Network (MSDN) 集成在 Visual Studio 集成开发环境 (IDE) 内。

1.4.1 启动和关闭 Visual Basic.NET 中文版

单击“开始”→“程序”→“Microsoft Visual Studio.NET 2003”命令，打开“Microsoft Development Environment [设计] - 起始页”窗口，如图 1-5 所示。这时根据用户的需要来启动 Visual Basic.NET 或

其他.NET 语言。

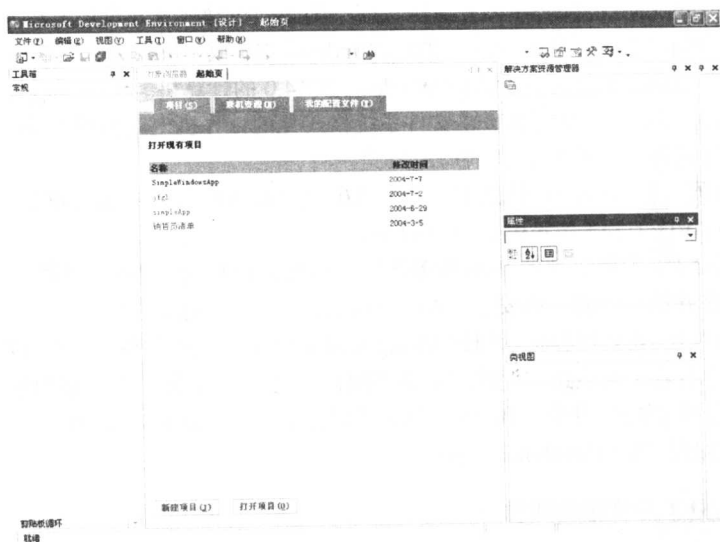


图 1-5 “Microsoft Development[设计]-起始页”窗口

启动 Visual Basic.NET 开发环境可以通过新建一个 Visual Basic.NET 项目或打开一个 Visual Basic.NET 项目来实现。

新建一个 Visual Basic.NET 项目的方法是：单击“文件”→“新建项目”命令或者单击“新建项目”按钮，打开“新建项目”对话框，在“项目类型”列表中选择“Visual Basic 项目”，然后在“模板”中选择一个项目模板，最后单击“确定”按钮就完成了。新建的 Visual Basic.NET 项目如图 1-6 所示。

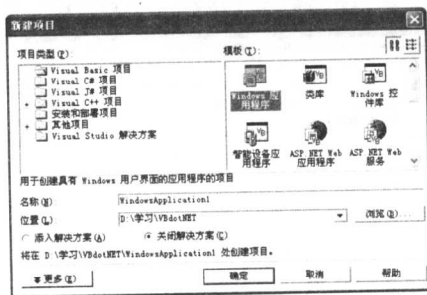


图 1-6 新建的 Visual Basic.NET 项目

要打开一个 Visual Basic.NET 项目，可以单击“文件”→“打开”→“项目”命令或者单击“起始页”中的“打开项目”按钮来实现。在“文件”→“最近的项目”命令项中也可以打开最近打开的若干项目。

关闭 Visual Basic.NET 时，可以同时关闭 Visual Studio.NET，通过单击窗口右上角的“关闭”按钮或单击“文件”→“退出”命令来实现。

如果想在关闭 Visual Basic.NET 之后打开其他语言环境，则可以单击“文件”→“新建”→“项目”命令来新建其他.NET 语言的项目，也可以通过单击“文件”→“打开”命令来打开其他.NET 语言的项目。

1.4.2 项目类型

使用 Visual Basic.NET 可以创建的项目类型与 Visual Basic 6.0 不同。在某些情况下，项目类型基