

新  
版

21世纪

高职高专系列教材

# Visual Basic.NET 程序设计案例教程

◎秦学礼 主编

◎杨儒亮 副主编



提供电子教案增值服务



机械工业出版社  
CHINA MACHINE PRESS



21 世纪高职高专系列教材

# Visual Basic.NET 程序 设计案例教程

主 编 秦学礼

副主编 杨儒亮

参编 丁明浩 金明霞 郇丽华 周晓昕

中国版本图书馆 CIP 数据核字 (2007) 第 117469 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓强

责任编辑: 王 磊

责任印制: 张双喜

北京双管印刷厂印刷

2007 年 8 月第 1 版·第 1 次印刷

184mm×260mm·19.5 印张·473 千字

0001-2000 册

标准书号: ISBN 978-7-111-21604-9

定价: 32.00 元



凡购本书, 如蒙赠页, 函索即寄, 邮费自理。

售后服务部电话: (010) 88379630

发行部电话: (010) 88379630

编辑部电话: (010) 88379630

地址: 北京市百万庄大街 22 号

机械工业出版社

本教材可以使读者掌握 Visual Studio. NET 的集成开发环境,能够利用 Visual Basic. NET 编写 Windows 应用程序、Web 应用程序,掌握 Windows 界面设计的方法和常用控件的使用,特别是加强读者的软件开发工程化的能力,如界面设计、菜单设计与数据库连接和应用技术、程序的打包、安装等技术。

本教材可供高职高专院校作为 Visual Basic. NET 程序设计课程的教材,也可供没有 Visual Basic. NET 程序设计经验的程序设计爱好者自学使用。

# Visual Basic. NET 程序设计案例教程

秦学礼 主编

秦学礼 副主编

图书在版编目(CIP)数据 案例教程 秦学礼 主编

Visual Basic. NET 程序设计案例教程/秦学礼主编. —北京:机械工业出版社,2007.8

(21 世纪高职高专系列教材)

ISBN 978-7-111-21604-9

I. V… II. 秦… III. BASIC 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 117469 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:王 颖

责任印制:洪汉军

北京双青印刷厂印刷

2007 年 8 月第 1 版·第 1 次印刷

184mm×260mm·19.25 印张·473 千字

0001—5000 册

标准书号:ISBN 978-7-111-21604-9

定价:27.00 元

凡购本书,如有缺页,倒页,脱页,由本社发行部调换

销售服务热线电话:(010)68326294

购书热线电话:(010)88379639 88379641 88379643

编辑热线电话:(010)88379739

封面无防伪标均为盗版

## 附录 出

### 21 世纪高职高专计算机专业系列教材

#### 编委会成员名单

主任 周智文 郭立宝

副主任 周岳山 林 东 王协瑞 赵佩华

吕何新 陈付贵 朱连庆 陶书中

委员 (按姓氏笔画排序)

马 伟 马林艺 卫振林 于恩普

王养森 王 泰 王德年 刘瑞新

余先锋 陈丽敏 汪赵强 姜国忠

赵国玲 赵增敏 顾可民 贾永江

顾 伟 陶 洪 龚小勇 睦碧霞

曹 毅 鲁 辉 翟社平

秘书长 胡毓坚

## 出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国近 60 所高等职业院校的骨干教师对在 2001 年出版的“面向 21 世纪高职高专系列教材”进行了全面的修订和增补,并更名为“21 世纪高职高专系列教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师,针对相关专业的课程设置,融合教学中的实践经验,同时吸收高等职业教育改革的成果而编写完成的,具有“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。在几年的教学实践中,本系列教材获得了较高的评价,并有多品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中,除了保持原有特色外,针对课程的不同性质采取了不同的优化措施。其中,核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时,根据实际教学的需要对部分课程进行了整合。

归纳起来,本系列教材具有以下特点:

- (1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- (2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度,强调专业技术应用能力的训练,适当增加实训环节。
- (3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁,多用图表来表达信息;增加相关技术在生产中的应用实例,引导学生主动学习。
- (4) 教材内容紧随技术和经济的发展而更新,及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念,并积极支持新专业的教材建设。
- (5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合,提高教学服务水平,为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快,加之我们的水平和经验有限,因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息,以利于我们今后不断提高教材的出版质量,为广大师生提供更多、更适用的教材。

机械工业出版社

# 前 言

Visual Basic.NET 程序设计语言是由 Microsoft(微软)公司开发的具有面向对象、分布式、可移植等性能的多线程计算机程序设计语言。作为 Visual Studio.NET 主体语言之一的 Visual Basic.NET,不仅可以开发 Windows 应用程序,而且可以开发 Web 应用程序及企业级分布式应用。Visual Basic.NET 程序设计继承了 Basic 语言面向普通使用者和易学易用的优点,同时又引入了可视化图形用户界面的程序设计方法和面向对象的程序设计,使之成为当今使用最为广泛、最有影响的程序设计语言之一。

我们在总结了本门课程教学的经验的基础上,编写了本教材。主要目的是使读者掌握 Visual Studio.NET 的集成开发环境开发面向对象应用程序的知识,能够利用 Visual Basic.NET 编写 Windows 应用程序、Web 应用程序,掌握 Windows 界面设计的方法和常用控件的使用,特别是加强读者的软件开发工程化的能力,如界面设计、菜单设计、与数据库连接和应用技术、程序的部署(打包)、安装等技术。

本教材的特点是:

- (1) 面向初学者,可以作为高职高专学生的程序设计入门课。
- (2) 本书用了两章介绍开发环境,引导读者学会使用 Visual Studio.NET 的集成开发工具设计程序、调试程序,为学习和使用打下良好的基础。
- (3) 每一章的例题有算法分析、源程序和程序分析,并结合每章知识点设计了案例和实训项目。
- (4) 实训项目给出了算法和实现步骤、关键程序,指导读者上机完成实训。
- (5) 全部例题以章为单位建立项目,源程序全部通过调试运行,读者可以修改成自己的程序。
- (6) 本书有一个小型的应用系统案例,包括数据库和全部源程序代码,可以作为教学的综合案例,但因为篇幅的限制,部分的源程序放在机械工业出版社的网站 [www.cmpbook.com](http://www.cmpbook.com) 上,读者可以下载。

参加本书编写的作者都是直接教授这门课程的教师,他们有丰富的教学 and 实践经验。

本书由秦学礼策划、组织和统稿。第 1 章和第 3 章由秦学礼编写,第 2 章由杨儒亮编写,第 4 章和第 5 章由丁明浩编写,第 6 章和第 7 章由金明霞编写,第 8 章和第 11 章由酆丽华编写,第 9 章和第 10 章由周晓昕编写。杨儒亮认真仔细地修改和审核了全书的原稿,编译运行了本书的所有程序。

本书在编写的过程中参考了许多作者的教材和著作,在此表示感谢。

由于作者的水平有限,书中错误在所难免,请读者提出批评和指正。

为了配合本书的教学,机械工业出版社为读者提供了电子教案和习题答案,读者可在网站 [www.cmpbook.com](http://www.cmpbook.com) 上下载。

编 者

# 目 录

|                                   |                                       |    |
|-----------------------------------|---------------------------------------|----|
| 出版说明                              | 2.5.1 案例1 Visual Studio.NET 集成开发环境的配置 | 31 |
| 前言                                | 2.5.2 案例2 建立一个项目                      | 32 |
| 第1章 .NET 与 Visual Basic.NET       | 2.6 实训                                | 33 |
| 1.1 微软的 .NET 技术及 .NET 框架的组成       | 2.6.1 实训1 VS.NET 集成开发环境设置             | 33 |
| 1.1.1 微软的 .NET 技术                 | 2.6.2 实训2 建立控制台应用程序项目                 | 33 |
| 1.1.2 .NET 框架的组成                  | 2.7 习题                                | 34 |
| 1.2 VB.NET 技术特性                   | 第3章 VB.NET 程序设计基础                     | 35 |
| 1.2.1 VB.NET 的主要特点                | 3.1 代码常识与命名约定                         | 35 |
| 1.2.2 VB.NET 的开发环境                | 3.1.1 代码常识                            | 35 |
| 1.3 Visual Studio.NET 集成开发环境(IDE) | 3.1.2 标识符及命名约定                        | 36 |
| 1.3.1 Visual Studio.NET 环境要求      | 3.2 变量、常量和基本类型                        | 36 |
| 1.3.2 Visual Studio.NET 安装和配置     | 3.2.1 数据类型                            | 36 |
| 1.3.3 IIS 安装和配置                   | 3.2.2 常量                              | 38 |
| 1.4 实训 Visual Studio.NET 安装和配置    | 3.2.3 变量                              | 39 |
| 1.5 习题                            | 3.2.4 变量的作用域与生存期                      | 41 |
| 第2章 VS.NET 集成开发环境                 | 3.2.5 类型转换                            | 42 |
| 2.1 认识 VS.NET 集成开发环境              | 3.3 复合数据类型                            | 44 |
| 2.1.1 VS.NET 集成开发环境的启动            | 3.3.1 数组                              | 44 |
| 2.1.2 VS.NET 集成开发环境的配置            | 3.3.2 结构                              | 45 |
| 2.2 VS.NET 的界面                    | 3.3.3 枚举                              | 47 |
| 2.2.1 VS.NET 主界面介绍                | 3.4 运算符与表达式                           | 49 |
| 2.2.2 VS.NET 的菜单栏                 | 3.4.1 算术运算符                           | 49 |
| 2.2.3 VS.NET 的工具栏                 | 3.4.2 比较运算符                           | 49 |
| 2.3 VS.NET 工具类窗口                  | 3.4.3 字符串连接运算符                        | 49 |
| 2.3.1 解决方案资源管理器                   | 3.4.4 逻辑运算符                           | 50 |
| 2.3.2 属性窗口                        | 3.4.5 运算符的优先级                         | 50 |
| 2.3.3 工具箱                         | 3.4.6 VB.NET 新增的运算符                   | 50 |
| 2.3.4 类视图窗口                       | 3.5 String 类                          | 51 |
| 2.3.5 服务器资源管理器                    | 3.5.1 String 类的声明                     | 51 |
| 2.4 VB.NET 的项目组织                  | 3.5.2 String 类的常用方法                   | 52 |
| 2.4.1 项目的建立                       | 3.6 实训                                | 57 |
| 2.4.2 项目的属性                       | 3.6.1 实训1 常量、变量的声明和运算符的综合应用           | 57 |
| 2.4.3 简单的控制台应用                    | 3.6.2 实训2 数组的声明和应用                    | 59 |
| 2.5 案例实践                          |                                       |    |

|                             |    |                          |     |
|-----------------------------|----|--------------------------|-----|
| 3.7 习题                      | 60 | 5.2 Form 类的属性和方法         | 98  |
| 第4章 流程控制结构                  | 62 | 5.2.1 Form 类的属性          | 99  |
| 4.1 选择                      | 62 | 5.2.2 Form 类的方法          | 100 |
| 4.1.1 If 语句                 | 62 | 5.2.3 Form 类的事件          | 103 |
| 4.1.2 选择的嵌套                 | 65 | 5.3 常用控件                 | 103 |
| 4.1.3 Select 语句             | 66 | 5.3.1 Button             | 103 |
| 4.2 循环                      | 68 | 5.3.2 Label              | 105 |
| 4.2.1 For Next 循环           | 69 | 5.3.3 TextBox            | 106 |
| 4.2.2 For Each Next 循环      | 70 | 5.3.4 RadioButton        | 109 |
| 4.2.3 Do Loop 循环            | 72 | 5.3.5 CheckBox           | 112 |
| 4.2.4 While... End While 循环 | 75 | 5.3.6 PictureBox         | 114 |
| 4.3 过程和函数                   | 76 | 5.3.7 MainMenu           | 115 |
| 4.3.1 Sub 过程                | 76 | 5.3.8 ContextMenu        | 116 |
| 4.3.2 Function 函数           | 78 | 5.3.9 Panel              | 118 |
| 4.3.3 事件过程                  | 80 | 5.3.10 GroupBox          | 119 |
| 4.3.4 值类型和引用类型              | 80 | 5.3.11 ToolTip           | 124 |
| 4.3.5 参数传递                  | 82 | 5.3.12 MessageBox        | 124 |
| 4.3.6 过程重载                  | 84 | 5.3.13 窗体的调用             | 127 |
| 4.4 常用函数                    | 84 | 5.4 案例实践                 | 127 |
| 4.4.1 数学函数                  | 85 | 5.4.1 案例1 输出杨辉三角         | 127 |
| 4.4.2 类型转换函数                | 86 | 5.4.2 案例2 设计记事本          | 128 |
| 4.4.3 日期和时间函数               | 86 | 5.5 实训                   | 131 |
| 4.4.4 字符串函数                 | 87 | 5.5.1 实训1 求圆的面积、周长和球的体积  | 131 |
| 4.4.5 格式化输出                 | 88 | 5.5.2 实训2 求1~9999之间的同构数  | 132 |
| 4.4.6 随机函数                  | 88 | 5.6 习题                   | 134 |
| 4.4.7 其他函数                  | 89 | 第6章 界面设计                 | 136 |
| 4.5 案例实践                    | 89 | 6.1 菜单的设计及应用             | 136 |
| 4.5.1 案例1 简单的排序             | 89 | 6.1.1 菜单的分类              | 136 |
| 4.5.2 案例2 找出最大值、最小值和平均值     | 90 | 6.1.2 菜单的组成              | 137 |
| 4.6 实训                      | 93 | 6.1.3 下拉菜单               | 138 |
| 4.6.1 实训1 求最大公约数和最小公倍数      | 93 | 6.1.4 快捷菜单               | 145 |
| 4.6.2 实训2 求能被17与37整除的数      | 94 | 6.2 工具栏                  | 147 |
| 4.7 习题                      | 95 | 6.2.1 工具栏控件              | 147 |
| 第5章 常用控件与应用                 | 97 | 6.2.2 工具栏控件应用实例          | 148 |
| 5.1 控件的属性和方法                | 97 | 6.3 状态栏                  | 149 |
| 5.1.1 控件的属性                 | 97 | 6.3.1 状态栏控件              | 149 |
| 5.1.2 控件的方法                 | 98 | 6.3.2 状态栏控件实例            | 150 |
| 5.1.3 控件的事件                 | 98 | 6.4 键盘事件                 | 151 |
|                             |    | 6.4.1 KeyPress 事件        | 151 |
|                             |    | 6.4.2 KeyDown 和 KeyUp 事件 | 151 |

|                                                   |     |                                         |     |
|---------------------------------------------------|-----|-----------------------------------------|-----|
| 6.5 鼠标事件 .....                                    | 153 | 8.2.4 引发异常 .....                        | 200 |
| 6.5.1 鼠标事件发生顺序 .....                              | 153 | 8.3 非结构化异常处理 .....                      | 203 |
| 6.5.2 MouseDown、MouseUp 和 Mouse-<br>Move 事件 ..... | 154 | 8.3.1 错误捕获 .....                        | 204 |
| 6.6 案例实践 .....                                    | 156 | 8.3.2 错误处理 .....                        | 205 |
| 6.7 实训 .....                                      | 161 | 8.3.3 退出错误处理器 .....                     | 208 |
| 6.7.1 实训 1 下拉菜单的应用 .....                          | 161 | 8.4 案例实践 .....                          | 209 |
| 6.7.2 实训 2 下拉菜单、工具栏的<br>应用 .....                  | 161 | 8.4.1 案例 1 异常处理 .....                   | 209 |
| 6.8 习题 .....                                      | 164 | 8.4.2 案例 2 错误处理 .....                   | 211 |
| 第 7 章 数据库和 ADO.NET 技术 .....                       | 165 | 8.5 实训 .....                            | 213 |
| 7.1 SQL 基础 .....                                  | 165 | 8.5.1 实训 1 算术运算溢出的异常<br>处理 .....        | 213 |
| 7.2 ADO.NET 数据访问技术 .....                          | 166 | 8.5.2 实训 2 四则运算溢出的异常<br>处理 .....        | 214 |
| 7.2.1 .NET 数据提供者 .....                            | 167 | 8.6 习题 .....                            | 216 |
| 7.2.2 DataSet 简述 .....                            | 168 | 第 9 章 文件处理 .....                        | 217 |
| 7.3 ADO.NET 主要组件 .....                            | 168 | 9.1 文件处理基础 .....                        | 217 |
| 7.3.1 Connection 对象 .....                         | 168 | 9.1.1 Directory 与 DirectoryInfo 类 ..... | 217 |
| 7.3.2 Command 对象 .....                            | 170 | 9.1.2 File 与 FileInfo 类 .....           | 218 |
| 7.3.3 DataAdapter 对象 .....                        | 172 | 9.1.3 Path 类 .....                      | 220 |
| 7.3.4 DataSet 对象 .....                            | 173 | 9.1.4 FileStream 类 .....                | 220 |
| 7.3.5 DataSet 对象模型 .....                          | 176 | 9.2 流文件的读取和写入 .....                     | 221 |
| 7.3.6 DataTable 对象 .....                          | 176 | 9.3 磁盘驱动器和文件夹的访问 .....                  | 224 |
| 7.3.7 DataColumn 对象 .....                         | 177 | 9.4 文件系统对象 (FSO) 模型 .....               | 226 |
| 7.3.8 DataRow 对象 .....                            | 177 | 9.5 案例实践 .....                          | 229 |
| 7.4 案例实践 .....                                    | 179 | 9.5.1 案例 1 文件读写实例 .....                 | 229 |
| 7.5 实训 .....                                      | 185 | 9.5.2 案例 2 文件信息 .....                   | 232 |
| 7.5.1 实训 1 ADO.NET 组件的使用<br>之一 .....              | 185 | 9.6 实训 .....                            | 237 |
| 7.5.2 实训 2 ADO.NET 组件的使用<br>之二 .....              | 188 | 9.6.1 实训 1 将文本框内容保存到<br>文件 .....        | 237 |
| 7.6 习题 .....                                      | 190 | 9.6.2 实训 2 文件夹基本操作 .....                | 238 |
| 第 8 章 程序异常和错误处理 .....                             | 192 | 9.7 习题 .....                            | 239 |
| 8.1 错误和异常的基本概念 .....                              | 192 | 第 10 章 多媒体应用与图形处理 .....                 | 240 |
| 8.1.1 语法错误 .....                                  | 192 | 10.1 媒体播放器相关的控件 .....                   | 240 |
| 8.1.2 运行时错误 .....                                 | 193 | 10.1.1 ScrollBar 控件 .....               | 240 |
| 8.1.3 逻辑错误 .....                                  | 193 | 10.1.2 TrackBar 控件 .....                | 241 |
| 8.2 结构化异常处理 .....                                 | 194 | 10.1.3 Timer 控件 .....                   | 242 |
| 8.2.1 捕获特定异常 .....                                | 195 | 10.1.4 MMC 控件 .....                     | 242 |
| 8.2.2 测试各种异常 .....                                | 197 | 10.1.5 MediaPlayer 控件 .....             | 243 |
| 8.2.3 在异常出现之后执行“处理”<br>工作 .....                   | 199 | 10.1.6 示例 简单的多媒体播放器 .....               | 244 |
|                                                   |     | 10.2 VB.NET 中的图形技术 .....                | 249 |
|                                                   |     | 10.2.1 坐标 .....                         | 250 |

|        |                                     |     |        |                          |     |
|--------|-------------------------------------|-----|--------|--------------------------|-----|
| 10.2.2 | 常用结构 .....                          | 252 | 11.1   | 生成安装程序 .....             | 267 |
| 10.2.3 | Graphics 类 .....                    | 252 | 11.2   | 安装文件的制作 .....            | 268 |
| 10.2.4 | 画笔 .....                            | 254 | 11.2.1 | 使用部署向导生成安装项目 ...         | 268 |
| 10.2.5 | 画刷 .....                            | 255 | 11.2.2 | 利用部署工程生成安装文件 ...         | 278 |
| 10.2.6 | 常用 API 函数 .....                     | 255 | 11.3   | 案例实践 .....               | 282 |
| 10.3   | 案例实践 .....                          | 257 | 11.3.1 | 案例 1 打包处理 .....          | 282 |
| 10.3.1 | 案例 1 多功能播放器 .....                   | 257 | 11.3.2 | 案例 2 生成安装文件 .....        | 284 |
| 10.3.2 | 案例 2 椭圆形绘图 .....                    | 260 | 11.4   | 实训 .....                 | 292 |
| 10.4   | 实训 .....                            | 262 | 11.4.1 | 实训 1 项目生成安装程序 ...        | 292 |
| 10.4.1 | 实训 1 MediaPlayer 播放器 ...            | 262 | 11.4.2 | 实训 2 部署项目的常用<br>设置 ..... | 292 |
| 10.4.2 | 实训 2 在 PictureBox 控件上<br>绘制矩形 ..... | 264 | 11.5   | 习题 .....                 | 294 |
| 10.5   | 习题 .....                            | 266 | 参考文献   | .....                    | 295 |
| 第 11 章 | 应用程序的安装与部署 .....                    | 267 |        |                          |     |

# 第1章 .NET 与 Visual Basic.NET

## 本章学习目标

- 了解微软公司的 .NET 框架、VB.NET 的主要特点。
- 熟悉开发环境 VS.NET 的安装和配置。

## 1.1 微软的 .NET 技术及 .NET 框架的组成

### 1.1.1 微软的 .NET 技术

2000年6月,比尔·盖茨向全球宣布了微软公司的下一代软件和服务,即微软.NET平台。这个平台称为“新一代Windows服务,Next Generation Web Services(NGWS)”,微软给这个平台注册了正式的商标——微软.NET。 .NET 家族产品和技术涵盖了构建下一代互联网服务和新一代智能互联网设备的软件。微软.NET从诞生到现在,已经走过了7个年头,伴随微软下一代操作系统Vista的发布,.NET技术又发生了很大的变化。

对于软件开发人员来说,.NET是Windows平台最大的改变之一。.NET的出现,使原来需要通过编程语言来做的一些工作转由.NET来完成。.NET的设计基于以下几个目标。

(1) 互操作性。基于.NET的程序需要和以前的程序进行交互,.NET提供了访问.NET环境外部资源的能力。

(2) 公用运行库引擎。基于.NET的编程语言被编译成公共中间语言,也就是通常说的CIL(Common Intermediate Language)。微软的CIL是微软中间语言(微软Intermediate Language,简称MSIL)。CIL并不是被解释执行的,而是以即时编译(JIT)的方式运行的,这种方式可以将中间代码编译成本地代码执行。这些概念称为公用语言架构(CLI),这个架构微软是以规范的形式发布的。CLI的实现就是公共语言运行库(Common Language Runtime,简称CLR)。

(3) 语言独立。.NET Framework引入了共用类库(Common Type System,简称CTS)。在CTS规范中定义了所有可能的数据类型、CLR支持编程架构以及这些架构之间如何交互。因为.NET提供了这个特性,它支持多种编程语言。

(4) 基础类库。基础类库(BCL),也叫做框架类库(FCL),它是一套支持.NET的语言都可以使用的类库。BCL提供了许多封装了通用函数的类,这些函数包括文件读写、图形图像操作、数据库以及XML操作等。

(5) 简化开发。Windows的注册表设置、文件分发以及DLL一直困扰着应用程序的安装和发布,基于.NET Framework的应用程序较好地解决了这些问题。

(6) 安全。.NET允许代码运行在不同的安全层。

### 1.1.2 .NET 框架的组成

.NET框架有两个组件:公共语言运行库和公共类库。.NET框架体系结构最顶部是VB.

NET、C#、VJ# 和 VC++ .NET 之类的语言,开发者能使用上面的任何语言开发像 Windows 窗体、Web 窗体、Windows 服务和 XML Web 服务,.NET 框架体系结构如图 1-1 所示。

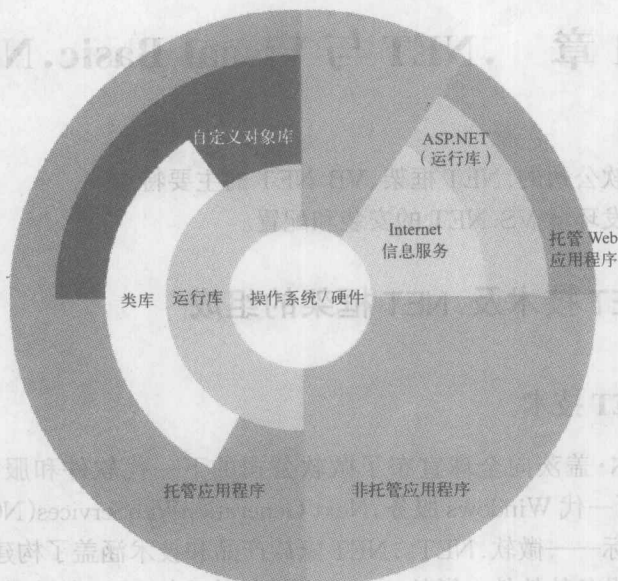


图 1-1 .NET 框架体系结构

### 1. 公共语言运行库(CLR)

公共语言运行库(Common Language Runtime, CLR)是 .NET 框架的运行环境和应用程序的执行引擎。公共语言运行库的设计目标是简化应用程序开发,提供强健和安全的执行环境,支持多种编程语言并简化部署和管理。该运行环境为基于 .NET 平台提供一个统一的、受控的运行环境。CLR 运行环境在 .NET 平台中充当一个类似于代理人的角色,例如内存垃圾回收机制、安全管理代码验证、编译及其他系统服务,CLR 通过中间语言等机制实现基于 .NET 的编程语言的无关性等。公共语言运行库提供许多服务,包括:

- 代码管理(加载和执行)。
- 应用程序内存隔离。
- 类型安全验证。
- IL 到本机代码的转换。
- 强制代码访问安全。
- 异常处理,包括跨语言异常。
- 托管代码、COM 对象和现有 DLL(非托管代码和数据)之间的互操作。

### 2. 公共类库(CTS)

.NET 提供了一个包含许多高度可重用的接口、类型的类库,该类库是一个完全面向对象的类库。公共类库是多信息类型系统,它被内置在公共语言运行库中,支持大多数编程语言中的类型和操作。

.NET 类库的组织是以名称空间(Name Space)为基础的,最顶层的名称空间是 System,查阅 .NET 文档中的 Class Library Reference,可以找到层次分明的各层名称空间下包含的各个

不同功能的类型定义和详细使用说明,这些名称空间是以其功能模块命名的,所以可以很快地找到所需要的类。

.NET 类库包含了许多用以简化编程工作的组件,类库不是 .NET 程序员可以使用的唯一的类库,它们同样可以使用第三方厂商提供的类库,因为类库是以 Name Space 组织的,可以避免命名冲突。

### 3. 数据访问服务

为了提供对数据的访问,服务框架包括 ADO.NET(ActiveX Data Objects+)类库。ADO.NET 的一个主要创新是引入了数据集(Dataset)。一个数据集是内存中提供数据关系图的高速缓冲区。数据集对数据源一无所知,它们可以由程序或通过从数据仓库中调入数据而被生成、填充。不论数据从何处获取,数据集都是通过使用同样的程序模板操作的,并且它使用相同的数据缓冲区。使用 .NET 平台的开发人员能够用数据集代替传统 ADO 中无连接的记录集。

ADO.NET 引入了一些新的特性——基于 XML,并且是松散耦合的(loosely-coupled)。ADO.NET 使用了脱机(disconnected)数据缓存,使用户能快速地创建出高性能、可靠的 XML Web 服务和现在流行的多层应用程序(N-tier applications)。

ADO.NET 为 .NET 框架提供一套统一的数据访问技术,ADO.NET 主要引入了以下几个新特性:

(1) 对 XML 的充分支持。

(2) 新数据对象的引入。

(3) 语言无关的数据访问。

(4) 使用 CLR 一致的类型。

## 1.2 VB.NET 技术特性

### 1.2.1 VB.NET 的主要特点

VB.NET 是一种面向对象的程序设计语言,它主要有以下几个特点:

#### 1. 面向对象编程

VB.NET 利用 .NET 框架提供的功能,引入了面向对象特性,如封装、继承、多态等,实现了面向对象的程序设计,是一种完全面向对象的程序设计语言。

#### 2. 使用 ADO.NET 进行数据访问

VB.NET 使用 ADO.NET 技术访问数据库。在 ADO.NET 中,用 Dataset(数据集)对象代替了 ADO 的 Recordset(记录集)对象,从而大大提高了数据处理的灵活性。ADO.NET 可以使用 XML 在应用程序之间、Web 网页之间进行数据交换。

#### 3. 进行 Web 应用程序的开发

微软公司将 .NET 框架主要定位在开发企业规模的 Web 应用程序及高性能的桌面应用程序上。NET 平台所强调的是网络编程和网络服务的概念,因此,基于 .NET 框架的 VB.NET,在网络应用程序开发方面有了显著的改进。VB.NET 提供了更直观、方便的 Web 应用程序开发环境,它可以用直接编辑 ASP.NET 的方式来开发 Web 应用程序,提供了开发 Web 服务的功能。

#### 4. 结构化错误处理

用 Try、Catch 和 End Try 等关键字,替代 On Error Goto 语句。

### 1.2.2 VB.NET 的开发环境

VB.NET 必须满足以下条件才能实现其强大的功能,它所要求的具体的运行环境如下:

(1) 可用的硬盘空间至少 600 MB。

(2) 操作系统为 Windows 2000、Windows XP 或 Windows 2003 中文版。

(3) 处理器应使用相当于 Pentium III 500Mhz 或以上级别的处理器。

(4) 内存要求 Windows NT4.0 Workstation, 64 MB; Windows NT4.0 Server, 160 MB; Windows 2000 Professional, 96 MB; Windows 2000 Server, 192 MB; Windows XP Professional, 160 MB。

(5) 微软 Internet Explorer 5.0 或 6.0 及以上版本。

只有符合上述条件,VB.NET 的功能才能得以实现。

Visual Basic .NET 的设计和开发不支持 Windows 95、Windows 98 和 Windows 98 Second Edition 等平台。

### 1.3 Visual Studio .NET 集成开发环境(IDE)

Visual Studio .NET 是一套完整的开发工具,用于生成 ASP、ASP.NET 的 Web 应用程序、XML Webservices、桌面应用程序和移动应用程序。VB.NET、Visual C++ .NET、Visual C#.NET 和 Visual J#.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。这些语言利用了 .NET Framework 的功能,提供对简化 ASP、ASP.NET 的 Web 应用程序和 XML Webservices 开发的关键技术的访问。

#### 1.3.1 Visual Studio .NET 环境要求

Visual Studio .NET 2003 这个系统较大,所以对硬件和操作系统都有较高的要求。安装 Visual Studio .NET 的计算机需满足下列系统要求:

(1) 处理器。相当于 Pentium III 级以上,600 MHz。

(2) 内存。Windows 2000 操作系统,要求 96 MB,推荐 128 MB 以上。Windows XP 和 Windows Server 2003 操作系统,要求 192 MB,推荐 256 MB 以上。

(3) 磁盘空间。系统驱动器上可用硬盘空间须大于 900 MB,安装驱动器上可用硬盘空间须大于 4.1 GB,将硬盘分区创建或转变为 NTFS 格式。

(4) 操作系统要求。Windows 2000、Windows XP、Windows Server 2003 或 Windows NT 4.0。

(5) 光驱。CD-ROM 驱动器或 DVD-ROM 驱动器。

(6) 其他软件环境:Visual Studio .NET 中的某些项目类型和功能要求必须在安装某些特定软件组件(可能在安装中作为可选组件列出)之后,才能使用这些功能或项目。其中有些组件必须安装在开发计算机上,有些则可以安装在远程计算机上。例如:开发 ASP、ASP.NET 的 Web 应用程序和 XML Webservices 时必须安装 Internet 信息服务(IIS),并且必须安装和配置 FrontPage 服务器扩展(FPSE)。安装 IIS 后,必须修复 .NET Framework 安装。

### 1.3.2 Visual Studio .NET 安装和配置

Visual Studio. NET 安装步骤。

(1) 将 Visual Studio .NET 2003 安装引导光盘第 1 张放入光驱, 光盘会自动引导, 显示如图 1-2 所示的提示框。

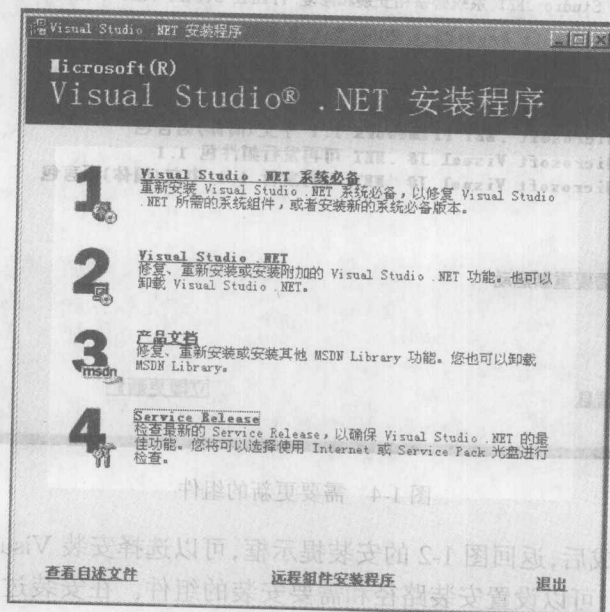


图 1-2 Visual Studio .NET 2003 安装提示框

(2) 安装程序首先要检测本机是否符合安装 Visual Studio .NET 的环境, 选择第 1 个选项“系统必备”。提示插入系统必备光盘, 如图 1-3 所示。

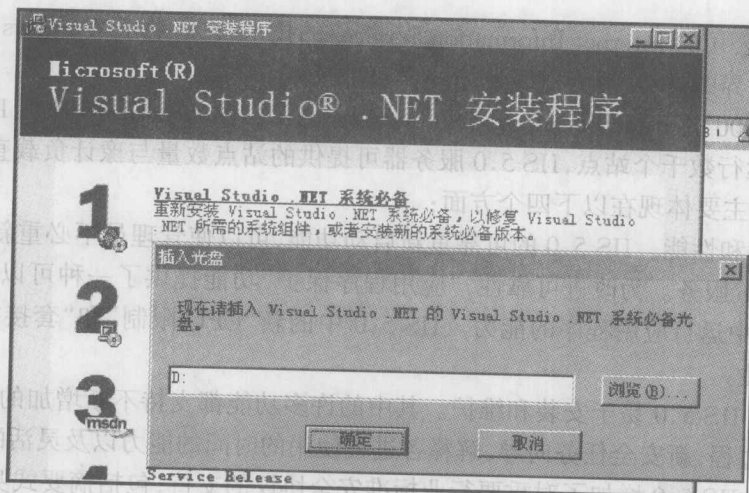


图 1-3 提示插入系统必备光盘

(3) 插入系统必备光盘, 单击“确定”, 将会显示如图 1-4 中列出的所有需要更新的组件。单击“立即更新”, 开始更新组件。

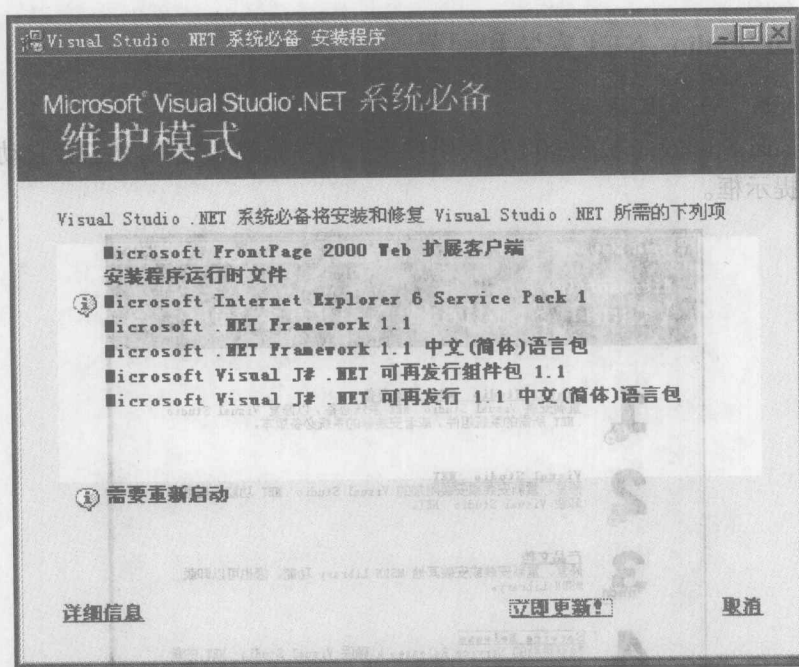


图 1-4 需要更新的组件

(4) 组件更新完成后,返回图 1-2 的安装提示框,可以选择安装 Visual Studio .NET。在安装程序的选项页面中,可以设置安装路径和需要安装的组件。在安装过程中文件的复制需要较长的时间,根据安装的进程,需要更换 3 次光盘。

### 1.3.3 IIS 安装和配置

#### 1. IIS 简介

IIS 是微软公司的 Internet Information Services 的技术,它借助于 Windows NT/2000/XP/2003 系统在 PC 界的优势,成为广泛使用的 Web 服务器软件之一。

Windows 2000 的 Internet Information Services(IIS)5.0(Windows XP 是 IIS 5.1)可以在一台服务器上运行数千个站点,IIS 5.0 服务器可提供的站点数量与预计负载直接成比例,IIS 5.0 的功能改进主要体现在以下四个方面:

(1) 可靠性和性能。IIS 5.0 的可靠重新启动功能,可以使管理员不必重新启动计算机即可重新启动 Web 服务。为改进可靠性,“应用程序保护”功能提供了一种可以在与 Web 服务分开,在缓冲池中运行应用程序的能力。IIS 5.0 中的新“CPU 限制”和“套接字共用”功能也能够改进可靠性。

(2) 管理。IIS 5.0 易于安装和维护。其中的许多功能都支持不断增加的易维护要求,包括简化的安装过程,新安全任务向导,解释各进程使用的的能力以及灵活的远程管理等。

(3) 安全。IIS 5.0 增加了对重要行业标准安全协议的支持,包括摘要式身份验证,Server Gated Cryptography(服务门限密码系统),Kerberos v5 身份验证协议,传输层安全等。另外,三个新任务向导,使得管理员可以更方便地管理站点的安全设置。

(4) 应用程序环境。IIS 5.0 建立在 Windows 2000 Server/XP 的新技术的基础上,从而扩

展了 Web 服务器的应用程序开发环境。这些新技术包括 Active Directory(活动目录)及扩展的组件对象模型(COM+)。还有对 IIS Active Server Pages(ASP)的增强,如无脚本 ASP 处理以及改进的流控制和错误处理让开发人员编写更有效的以 Web 为中心的应用程序。

在 Windows Server 2003 平台上运行的 IIS 6.0 引入了两种用于配置应用程序环境的操作模式,即工作进程隔离模式和 IIS 5.0 隔离模式。在安装 IIS 6.0 时默认的隔离模式取决于执行的是全新安装还是升级。在全新安装 IIS 6.0 之后,IIS 以工作进程隔离模式运行。

IIS 6.0 包含在 Windows Server 2003 服务器的数据中心版、企业版、标准版、Web 版四种版本之中。IIS 6.0 不能在 Windows XP、2000 或 NT 上运行。Windows 2003/IIS 6.0 的一个关键的变化是,除了 Windows 2003 Web 版之外,Windows 2003 的其余版本默认不安装 IIS。

在 Windows 2003 中,安装 IIS 有三种途径:利用“管理您的服务器”向导;利用控制面板“添加或删除程序”的“添加/删除 Windows 组件”功能,或者执行无人值守安装。

在 IIS 6.0 中,所有文档都以帮助文件的形式发布,不再有 IIS Help 虚拟目录。在 IIS 5.0 中,如果从本地访问服务器,默认 Web 网站自动打开 IIS 的文档,但在 IIS 6.0 中,如果打开“http://localhost”,只能看到一个声明网站正在构建之中的页面。

IIS 6.0 体系设计中最重大的变化是加入了 http.sys 驱动程序,http.sys 驱动程序的任务是处理 HTTP 请求,而且它在内核模式下执行操作。不要小看这一改变,将处理 HTTP 请求的任务从 IIS 5.0、IIS 4.0 的用户模式改变到 IIS 6.0 的内核模式标志着新一代 IIS 服务器的诞生。在 Windows 2000 和 NT 4.0 中,IIS 在用户模式下运行。运行在用户模式下的应用程序不直接与硬件通信,它们直接调用的是一些标准过程,这些标准过程或者将数据传入内核模式的组件(例如网卡驱动程序,图形子系统),或者调用内核模式组件的函数,以此完成保存文件、设置 IP 地址、将 HTML 文件发送到网络的任务。

## 2. IIS 管理器的安装和配置

在 Windows 2000 Server/XP 系统添加 IIS 的操作方法如下:

(1) 在“控制面板”中双击“添加/删除程序”对话框。

单击“添加/删除 Windows 组件”,再单击右上角的“组件”,弹出如图 1-5 所示的“Windows 组件向导”窗口。

(2) 在“Windows 组件向导”对话框中,选择“Internet 信息服务(IIS)”组件,单击“详细信息”进行进一步设置,FTP 服务、SMTP 服务、NNTP 服务通常可以选择安装。

(3) 单击“下一步”,从 Windows 2000/XP 安装光盘中复制所需文件。

(4) 重新启动计算机,完成 IIS 安装。

## 3. IIS 管理器的应用

安装完成之后,就可以从开始菜单中启动 IIS 5.0/5.1 的操作界面。

单击“开始”→“程序”→“管理工具”→“Internet 服务管理器”,打开如图 1-6 所示的 IIS 管理界面,称为 Internet 服务管理器,简称为 ISM。这是一个标准的 MMC(微软管理控制台)界面,MMC 是微软专门为各种管理工具开发的一个统一的操作环境。简单地说,任何 MMC 界面都可以分为以下三个部分:

(1) 上部菜单与工具条。

(2) 左侧范围部分,以树型结构表示计算机所包含的站点及其内容。

(3) 右侧内容部分,显示在左侧范围部分选中节点的具体内容。