



程序员指南丛书



Visual C#.NET 编程基础

- Visual Studio 家族的新成员,在编译优化技术方面较其他同类产品具有明显的优势。
- 通过实例全面讲解 Visual C# 的语言知识点和编程技巧,使读者能够迅速掌握这门编程语言。
- 对于初学程序设计的人员,通过本书即可轻松入门;对于已有其他开发工具编程经验的读者,可以进一步提高编程技术。

李满潮 等 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



Visual C#.NET 编程基础

李满潮 等 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Visual C# 是 Visual Studio 家族的新成员,它不仅全面贯彻了面向对象技术,而且在编译优化技术方面较其他同类产品具有明显的优势。本书通过大量实例,深入浅出地介绍了 Visual C# 的使用与编程。全书共 8 章,主要包括:Visual C# 的开发环境的菜单体系、Visual C# 语言、Windows 编程基础、管理和使用 Visual C# 对象、窗体应用程序的构造以及窗体界面设计等。

本书内容全面、深入,适合初中级读者、大中专院校师生、企业技术开发人员学习参考,也适合各类培训班学员学习 Visual C# 程序设计技术用书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Visual C#. NET 编程基础

作 者: 李满潮 等

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 胡先福

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 17.25 字数: 416 千字

版 次: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

书 号: ISBN 7-302-04974-2/TP · 2801

印 数: 0001 ~ 6000

定 价: 25.00 元

前 言

Visual C# 是 Visual Studio. NET 家族的新成员, Visual Studio. NET 家族包括 Visual C#、Visual Basic、Visual C++ 等开发工具。根据习惯, Visual Studio. NET 又被称为 Visual Studio 7.0。

Visual C# 是当今最新、最先进的软件开发工具之一, 它汇集了微软公司的技术精华, 不仅全面贯彻了面向对象技术, 而且在编译优化技术方面较其他同类产品具有明显的优势。它是一个彻底的程序员级的开发环境, “可视化”的设计减少了不少编程的工作量。利用 Visual C# 几乎可以完成任何设计功能, 小至普通的应用系统, 大至复杂的应用开发工具。在 Visual Studio. NET 中, Visual Basic 和 Visual C# 都具备 RAD(快速应用程序开发, rapid-application development)支持, 例如项目模板、设计器和其他开发环境特性。这两种语言都使用 .NET 框架基类。

本书通过大量实例, 深入浅出地介绍了 Visual C# 的使用与编程。全书共 8 章, 主要内容包包括: Visual C# 的开发环境和菜单体系、Visual C# 语言、Windows 编程基础、管理和使用 Visual C# 对象、窗体应用程序的构造以及窗体界面设计等。

本书通俗易懂、易于理解掌握, 对 Visual C# 的几乎所有方面都进行了介绍。对于初学程序设计的人员, 通过本书即可轻松入门; 对于已有其他开发工具编程经验的读者, 通过本书不仅可以快速掌握 Visual C#, 并且可以进一步提高编程技术。

本书并非只是知识点的简单罗列, 而是通过实例向读者全面讲解 Visual C# 的语言知识点和编程技巧, 这使得读者能够掌握并灵活运用这些知识点, 迅速掌握这门编程语言。

除封面署名外, 参与本书编写、校对等工作的还有胡大惟、葛文全、张思明、吴迪、曹立勋、宫小东、吴清萍、安浦、方克志、包利群、于丽闽、李国文、汪浩、陈彬、李高云等, 在此表示感谢。由于水平所限, 书中不足和纰漏之处在所难免, 恳请广大读者批评指正。

作者

2001 年 10 月

目 录

第 1 章 掌握 Visual C# 开发环境	1
1.1 Visual Studio .NET 的安装需求	1
1.1.1 硬件需求	1
1.1.2 软件需求	2
1.2 Visual C# 的新特性	2
1.2.1 Web 开发的新特性	2
1.2.2 数据开发的新特性	3
1.2.3 编程语言的新特性	3
1.2.4 Windows 窗体和控件的新特性	3
1.2.5 项目的新特性	3
1.2.6 调试的新特性	3
1.2.7 组件和组件创建的新特性	4
1.2.8 国际化应用程序的新特性	4
1.3 Visual C# 集成开发环境简介	4
1.3.1 先进的窗口管理技术	5
1.3.2 强大的编辑功能	6
1.4 使用 Visual C# 集成开发环境	6
1.4.1 Visual Studio 起始页	6
1.4.2 方案管理器	11
1.4.3 Class View	13
1.4.4 属性窗口	15
1.4.5 工具箱	16
1.4.6 编辑器	27
1.4.7 服务器管理器	28
1.4.8 对象浏览器	31
1.4.9 任务清单窗口	35
1.4.10 宏管理器	36
1.4.11 命令窗口	39
1.4.12 安装工具	39
本章小结	40

第2章 Visual C# 的菜单体系	41
2.1 File 菜单	41
2.2 Edit 菜单	44
2.3 View 菜单	45
2.4 Project 菜单	46
2.5 Build 菜单	50
2.6 Debug 菜单	51
2.7 Tools 菜单	51
2.8 Window 菜单	56
2.9 Help 菜单	57
本章小结	58
第3章 掌握 Visual C# 语言	59
3.1 C# 与 .NET 框架	59
3.1.1 部件	59
3.1.2 using 指令	60
3.1.3 名称空间	61
3.1.4 全限定名	62
3.1.5 标志	65
3.2 C# 语言的新特性	66
3.2.1 先进的语言特性	66
3.2.2 丰富的类库支持	66
3.2.3 自动内存管理	66
3.2.4 统一类型系统	67
3.2.5 版本兼容	67
3.3 程序结构和编码惯例	67
3.3.1 C# 程序的一般结构	67
3.3.2 “你好,世界!”	68
3.3.3 C# 命名惯例	70
3.4 声明元素	71
3.4.1 声明元素名称	71
3.4.2 声明元素的特征	72
3.4.3 声明空间	73
3.4.4 生命期	74
3.4.5 作用域	76
3.4.6 声明元素的引用	77
3.4.7 名称隐藏	79
3.5 数据类型	80

3.5.1	基本数据类型	80
3.5.2	复合数据类型	86
3.5.3	数据类型实现	86
3.6	类型转换	87
3.6.1	隐式类型转换	88
3.6.2	显式类型转换	90
3.6.3	自定义转换	92
3.6.4	转换中的值变化	93
3.7	变量	93
3.7.1	变量声明	93
3.7.2	默认赋值	95
3.7.3	显式赋值	96
3.8	运算符和表达式	97
3.8.1	运算符的优先级	98
3.8.2	算术表达式	99
3.8.3	逻辑表达式	103
3.8.4	关系运算符	107
3.8.5	前(后)缀表达式	107
3.8.6	移位运算符	107
3.8.7	条件表达式	109
3.8.8	赋值表达式	109
3.8.9	其他表达式	110
3.9	数组	111
3.9.1	数组类型声明	112
3.9.2	数组初始化	113
3.10	结构	115
3.10.1	结构声明	115
3.10.2	结构和类	116
3.11	流程控制	117
3.11.1	选择结构	117
3.11.2	循环结构	120
3.11.3	跳转语句	123
本章小结		124
第4章	Windows 编程基础	125
4.1	Windows 窗体	125
4.1.1	Windows 窗体概述	125
4.1.2	Windows 窗体的新特性	126

4.2	事件驱动	127
4.4	创建第一个 Windows 窗体应用程序	129
4.5	使用向导添加程序元素	132
4.5.1	添加类	132
4.5.2	添加方法成员	135
4.5.3	添加属性成员	137
4.5.4	添加字段成员	139
4.5.5	添加分度器成员	140
4.5.6	添加事件和事件处理函数	141
	本章小结	143
第 5 章	管理和使用 Visual C# 对象	144
5.1	类和对象	144
5.2	定制类	146
5.2.1	类声明	146
5.2.2	构造函数	150
5.2.3	析构函数	152
5.2.4	定制字段和常量	152
5.2.5	定制属性	155
5.2.6	定制方法	162
5.2.7	定制事件	167
5.2.8	定制分度器	170
5.2.9	定制运算符	172
5.3	创建和使用对象	174
5.3.1	设置和获取属性	174
5.3.2	通过方法执行动作	175
5.3.3	执行多个动作	175
5.3.4	操作窗体对象	176
5.3.5	使用 new 关键字	176
5.3.6	资源管理	177
5.3.7	使用对象参数	178
5.3.8	管理对象数组	180
5.4	获取对象类型信息	180
5.4.1	检查对象类型	181
5.4.2	确定类型尺寸	182
5.4.3	获取 System.Type 对象	183
	本章小结	184

第 6 章 构造 Windows 窗体	185
6.1 使用窗体设计器	185
6.1.1 创建窗体	185
6.1.2 设置窗体尺寸	186
6.1.3 设置窗体位置	187
6.1.4 设置窗体边框风格	188
6.1.5 添加和调整控件	189
6.2 窗体中的事件处理	194
6.2.1 事件和 delegate	194
6.2.2 事件处理函数	196
6.2.3 创建事件处理函数	196
6.2.4 创建默认事件处理函数	196
6.2.5 在运行时创建事件处理函数	197
6.2.6 使用单个事件处理函数响应多个事件	197
6.3 使用常用控件	198
6.3.1 标签控件	198
6.3.2 图片框控件	199
6.3.3 按钮控件	200
6.3.4 复选框控件	201
6.3.5 单选按钮控件	203
6.3.6 文本框控件	204
6.3.7 列表框控件	207
6.3.8 组合框控件	209
6.3.9 编组框控件	211
6.3.10 滚动条控件	212
6.3.11 计时器控件	213
6.4 创建不同形式的窗体	213
6.4.1 创建隐藏窗体	213
6.4.2 创建顶层窗体	214
6.4.3 创建透明窗体	214
6.4.4 模态和非模态窗体	215
本章小结	216
 第 7 章 菜单、工具栏和状态栏	 217
7.1 菜单的设计和使用	217
7.1.1 添加常规菜单	217
7.1.2 设计和使用常规菜单	218
7.1.3 使用快捷菜单	223

7.1.4 菜单应用实例	227
7.2 工具栏的设计和使用	230
7.2.1 设计工具栏	230
7.2.2 使用工具栏	231
7.3 状态栏的设计和使用	232
7.3.1 设计状态栏	232
7.3.2 使用状态栏	233
本章小结	234
第8章 设计窗体界面	235
8.1 界面设计基础	235
8.1.1 设计原则	235
8.1.2 Windows 界面规则	237
8.1.3 界面布局原则	237
8.1.4 用户辅助模型	240
8.1.5 选择界面样式	240
8.2 设计对话框窗体	242
8.2.1 创建对话框	242
8.2.2 获取对话框数据	243
8.2.3 关闭对话框	246
8.3 设计资源管理器风格的界面	247
8.4 设计 MDI 风格的界面	249
8.4.1 创建 MDI 父窗体	249
8.4.2 创建 MDI 子窗体	250
8.4.3 确定活动子窗体	251
8.4.4 向活动子窗体中发送数据	252
8.4.5 排列子窗体	253
8.4.6 保存子窗体的信息	253
8.5 窗体程序设计实例	254
8.5.1 创建程序框架	254
8.5.2 编辑子窗体代码	256
8.5.3 编辑主窗体代码	260
本章小结	264

第1章 掌握 Visual C# 开发环境

Visual C#. NET 是 Visual Studio. NET 家族的一个成员, Visual Studio. NET 包括 Visual C#、Visual Basic、Visual C++ 等开发工具。根据习惯, Visual Studio. NET 又被称为 Visual Studio. NET。在 Visual Studio. NET 中, C# 语言得到了全面支持, 这体现在开发环境的工程模板、设计助手、属性页、代码助手、对象模型等功能中。

Visual Studio. NET 主要用于开发企业规模的 Web 应用程序, 以及高性能的桌面应用程序, 它为 .NET 框架(.NET Framework) 提供了全面支持。.NET 框架由通用语言运行库(Common Language Runtime)、统一编程类和 ASP.NET(Active Server Pages. NET) 组成。这些组件主要用于创建 Web 应用程序和 Web 服务。

Visual Studio. NET 为 XML 提供了完全的支持, 例如, 引入了 XML Designer 以简化 XML 编辑和创建等。在 Visual Studio. NET 中, 还能使用 Visual Basic、Visual C# 创建和使用 Web 服务。Visual Studio. NET 通过模板为应用程序提供了结构化控制。企业模板可以降低软件设计开发小组的开发难度和成本。

Visual Studio. NET 为 Visual C#、Visual C++ 和 Visual Basic 等提供统一的集成开发环境 (IDE, Integrated Development Environment), 其中集成了许多可视化辅助工具, 能够大大简化程序的开发和提高编程效率。从界面风格上讲, Visual Studio. NET 属于 XP 系列。而且, 相对于 Office XP, Visual Studio. NET 在开发环境上与先前版本有着更大的不同。因此, 要使用 Visual C# 进行开发, 首先必须掌握其开发环境的使用。本章将向读者介绍 Visual C# 的集成开发环境。

本章要点:

- ❖ Visual Studio. NET 的安装需求
- ❖ Visual C# 的新特性
- ❖ Visual C# 集成开发环境的使用

1.1 Visual Studio. NET 的安装需求

由于 Visual Studio. NET 本身的功能非常强大, 因此对软硬件要求都较高。

1.1.1 硬件需求

Visual Studio. NET 不能安装在映射驱动器上, 而必须在本地驱动器上进行安装。安装的具体硬件需求如下:

- CPU

CPU 至少应为 Pentium II 450 MHz, 推荐使用 Pentium III 600 MHz。

- 内存

如果操作系统为 Windows 2000 Professional, 内存至少应为 96 MB, 推荐使用 128 MB; 如果操作系统为 Windows 2000 Server, 内存至少应为 192 MB, 推荐使用 256 MB。

- 硬盘空间

系统驱动器上至少应有 500 MB 空间, 而安装的目标驱动器上至少应有 2.5 GB 空间。

- CD-ROM 驱动器

除非能直接从网络下载安装, 或硬盘中存在安装内容, 否则必须具有 CD-ROM 驱动器。

- 显示器

显示器至少应为 800 × 600 256 色, 推荐使用 1024 × 768 16 位增强色。

1.1.2 软件需求

安装 Visual Studio. NET 的具体软件需求如下:

- 操作系统

Windows 2000、Windows XP 和 Windows NT 4.0 及以上版本。

- 其他组件

Windows NT 4.0 Service Pack 6a、Windows 2000 Service Pack 2、Microsoft Windows Installer 2.0、Microsoft Windows Management Infrastructure、Microsoft FrontPage 2000 Web Extensions Client、Microsoft FrontPage 2000 Server Extensions Service 1.2、Microsoft Internet Explorer 6.0、Microsoft Data Access Components 2.7、Microsoft Jet 4.0 Service Pack 3 和 Microsoft .NET Framework。

1.2 Visual C# 的新特性

在 Visual Studio. NET 中, Visual Basic 和 Visual C# 都具备 RAD (快速应用程序开发, Rapid-Application Development) 支持, 例如项目模板、设计器和其他开发环境特性。这两种语言都使用 .NET 框架基类。

1.2.1 Web 开发的新特性

Visual Studio. NET 的一个重要新特性就是能创建 Web 应用程序, 也就是说, 运行于 Web 服务器上的 Visual C# 和 Visual Basic 应用程序。

Web 应用程序中最主要的两个开发主题为 Web 窗体和 Web 服务:

- Web 窗体基础允许用户迅速而方便地创建 ASP. NET Web 应用程序的用户界面。Web 窗体页面是对现有 Web 开发工具的变革性进步, 它兼有速度和 RAD 开发环境的强大功能。Web 窗体的输出不依赖于任何浏览器或移动设备, 并且能自动提交正确的、与浏览器 (风格、布局等) 兼容的 HTML。

- Web 服务是能通过 Internet 协议调用其他组件或应用程序的组件。Web 服务允许使用标准协议(例如 HTTP)进行数据交换,而且能通过 XML 消息穿过防火墙移动数据。Web 服务不依赖于某种特定组件技术或对象调用规范。作为结果,无论运行于什么操作系统上,无论使用什么编程语言和组件模型编写的程序,都能访问 Web 服务。

1.2.2 数据开发的新特性

ADO.NET 是微软 ADO(ActiveX 数据对象,ActiveX Data Objects)的革命性进步,它为 Web 应用程序开发提供了标准编程模型。ADO.NET 是根据离线模型设计的,因此能使用与数据源隔离的数据快照。

1.2.3 编程语言的新特性

C# 是 Visual Studio.NET 中引入的全新编程语言。作为 C 和 C++ 的派生语言,C# 具有简单、先进、类型安全和面向对象等特点。C# 就是为了构造能运行于 .NET 平台上的企业级应用程序而设计的。使用 C# 编写的代码将被作为受控代码编译,这意味着它能利用通用语言运行库的服务,例如语言互用、冗码收集、增强的安全性和改进的版本支持等。

Visual Studio.NET 通过项目模板、设计器、属性页、代码辅助、对象模型和其他开发环境特性,为 C# 提供完全支持。C# 的程序库就是 .NET 框架。

1.2.4 Windows 窗体和控件的新特性

Windows 窗体是一种新型的面向对象的框架,它实现了 .NET 框架 Windows 窗体、Windows 窗体控件,并提供了强健的 Windows 应用程序开发结构。

1.2.5 项目的新特性

项目是受控 Visual C# 和 Visual Basic 代码的构造块。已完成的应用程序中可能包含多个项目。以下是在 Visual C# 中引入的新项目模板:ASP.NET Web 服务项目、ASP.NET Web 应用程序项目、Windows 服务项目、控制台应用程序项目、Web 控件库项目等。

1.2.6 调试的新特性

Visual Studio.NET 中的所有语言使用同样的调试器,它同时具有旧版本的 Visual C++ 和 Visual Basic 调试器的功能,并在很多地方进行了改进,这包括:对 Visual Basic、Visual C++、Visual C#、C++ 管理扩展、脚本和 SQL 等语言的交叉调试;Microsoft .NET 通用语言运行库和 Win32 应用程序的调试;对主机或远程机运行程序的附加调试;多个程序的同时调试(例如,单个 Visual Studio 解决方案中运行的多个程序,或附加于已运行的附加程序);

Visual C++ 动态错误检查;缓冲区安全检查等。

1.2.7 组件和组件创建的新特性

使用 Visual C# 和 Visual Basic,能够在代码中方便地创建自己的组件,或配置 .NET 框架类中的复杂组件。使用非可视化组件能够在自己的应用程序中方便地使用消息队列、事件日志、性能计数等资源。

.NET 框架类极大地扩展了 Visual Studio 的组件编程性能。虽然基类有限,但可以使用很大的复杂组件库来配置自己的应用程序,或派生自己的高质量组件。

1.2.8 国际化应用程序的新特性

Visual Studio .NET 为创建国际化应用程序提供了很大支持,这包括两个方面:国际化,设计能用于不同语言的应用程序;本地化,将资源翻译为指定语言。

1.3 Visual C# 集成开发环境简介

Visual Studio .NET 为 Visual C# .NET、Visual C++.NET 和 Visual Basic .NET 等提供统一的集成开发环境,从而为开发者提供了很大方便。Visual Studio 的启动界面如图 1-1 所示。

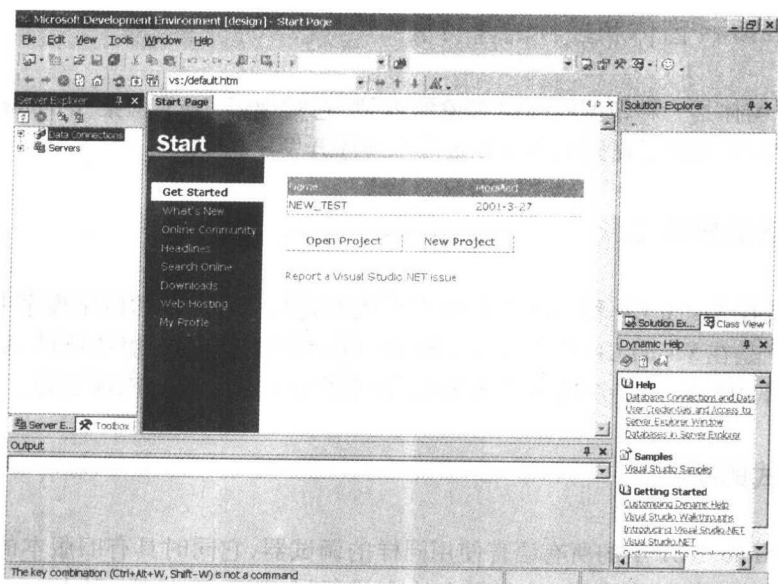


图 1-1 Visual Studio .NET 的启动界面

由图 1-1 可见,Visual Studio .NET 的开发环境有很多新改进,例如在 IDE 中直接显示网

页。当首次打开 Web 浏览器窗口或 Visual Studio 时,其中将显示默认的 Visual Studio 起始页(Start Page)。如果 Web 浏览器被打开,则同时也会出现 Web 工具栏,其功能与 IE 中的相同。

1.3.1 先进的窗口管理技术

Visual Studio. NET 中采用了更有效的窗口管理策略,例如文档标签化、自动隐藏等,从而大大提高了浏览效率,节省了屏幕空间。图 1-2 所示为 Visual Studio. NET 中的窗口布局,其中包括集成开发环境中的所有主要窗口。

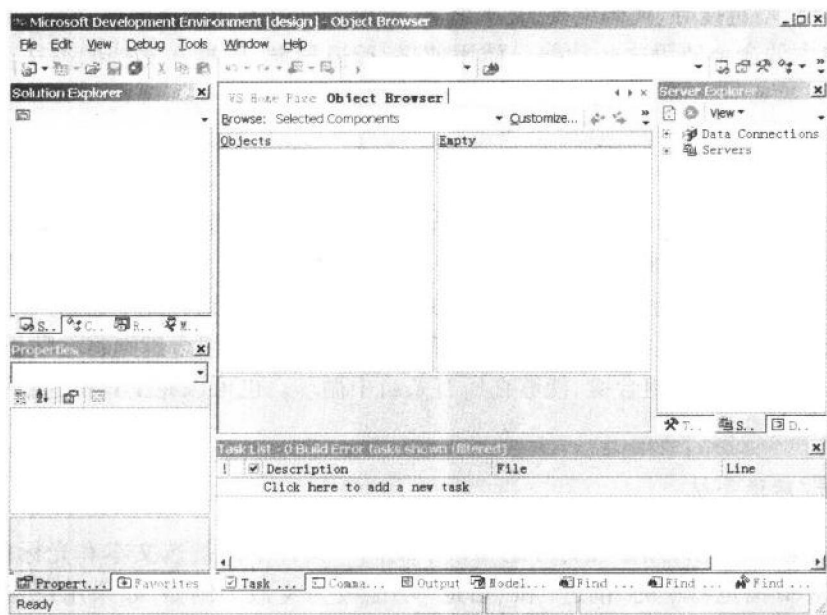


图 1-2 Visual Studio. NET 中的主要窗口

1. 自动隐藏

在窗口标题栏处右击,在弹出的快捷菜单中选择 Auto Hide 命令;或使目标窗口成为活动窗口,并选择 Window | Auto Hide 命令,即可实现自动隐藏。自动隐藏使得窗口在失去焦点时自动最小化,并将其图标排列在 IDE 边缘,从而节省宝贵的屏幕空间,提高编辑器的可视面积。

2. 标签化文档

标签化文档功能会自动为文档窗口设置标签,当窗口失去焦点时其标签仍然会显示于原窗口的边缘。例如,默认情况下 Class View、Resource View、宏管理器和方案管理器使用同一窗口,而通过切换底部标签,即可在同一窗口中查看不同视图中的内容。当用户在编辑器或设计器中编辑多个文档时,虽然一次只能显示一个文档中的内容,但是通过顶部/底部标

签可以方便地浏览其他文档中的内容,参见图 1-2。这较之以前的版本中必须通过 Window 菜单的窗口列表选择,其提供的方便显而易见,尤其是在打开的文档数较多时。

1.3.2 强大的编辑功能

Visual Studio. NET 的编辑工具有了进一步增强。

1. 编辑代码

Visual Studio. NET 的所有产品使用统一的代码编辑器和检索/替换工具。在编辑器中能够实现字回绕、增量搜寻、代码纲要、文本缩进、行编号和书签功能等增强功能。用户能够通过 Edit 菜单中的命令使用这些功能。使用前进/后退导航,能够在文档的被选内容历史中进行切换。

2. 剪贴板环

剪贴板环(Clipboard ring)中能够储存系统剪贴板中被复制或剪切的最后 12 个条目。使用 Ctrl + Shift + V 组合键,或通过工具框中的 Clipboard 标签,用户能够查看剪贴板环中的内容。按 Ctrl + Shift + V 组合键将一个条目从剪贴板环中粘贴到文件中的插入点,并自动将被粘贴的条目设置为被选状态。如果再次按下 Ctrl + Shift + V 组合键则将粘贴下一条目,因此重复按 Ctrl + Shift + V 组合键,能够在所有条目中循环。也可以在工具框的剪贴板标签和文档间通过拖曳交换文本。

3. 前进/后退导航

前进/后退导航允许用户在打开的窗口中或同一文件中的被选文本和光标位置中进行切换,这与 Web 浏览器中的“前进”和“后退”功能极为类似。例如,如果用户在第 12 行中编辑代码后移动到第 102 行,这时就可以通过 Navigate Backward 按钮返回光标上次在 12 行中的位置。Navigate Backward 和 Navigate Forward 按钮都带有下拉列表,其中显示了导航历史。这两个按钮位于标准工具栏中,用户也可以通过按 Ctrl + - 和 Ctrl + Shift + - 组合键来完成导航功能。

1.4 使用 Visual C# 集成开发环境

本节将向读者介绍 Visual Studio. NET 中常用开发辅助工具的使用。

1.4.1 Visual Studio 起始页

默认情况下,每次 Visual Studio 启动时,Visual Studio 起始页都显示在其用户界面的 Web 浏览器窗口中,如图 1-3 所示。它提供了设置诸如 IDE 行为、键盘类型、窗口布局等用

户参数,以及进行打开、新建项目等操作的快速途径。通过起始页还能查看最新的文章、事件和 MSDN 在线的主题。

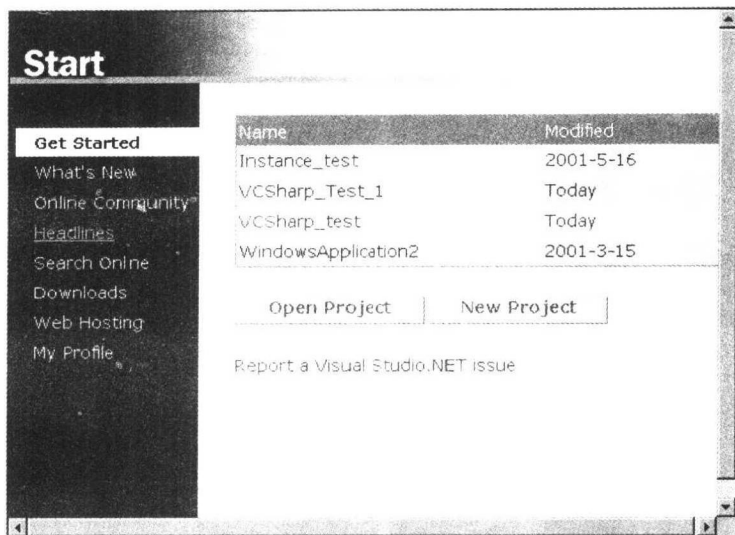


图 1-3 Visual Studio 起始页

Visual Studio 起始页中的各项功能如下:

1. Get Started

单击 Get Started 项,将在起始页中列出最近打开的方案(方案是 Visual Studio.NET 中引入的新概念,将在后文中详加介绍),如图 1-3 所示。图中的 Instance_test、VCSharp_Test_1、VCSharp_test 和 WindowsApplication2 即为最近打开的方案,用户还可以通过不同的对话框创建新项目,或打开已创建的项目。

- Name

列出了先前创建的方案名称。

- Modified

列出了方案被最后修改的日期和事件。

- Open Project

单击该按钮,将显示 Open Project 对话框,以打开现存项目。

- New Project

单击该按钮,将显示 New Project 对话框,以创建新项目。

- Report a Visual Studio.NET issue

单击该超链接,将向微软公司发送使用意见。

2. What's New

点击 What's New 项,将在起始页中显示一些超级链接,如图 1-4 所示。这些超级链接为 Visual Studio.NET 有关的主题提供了 Web 页。