



程序员指南丛书

Visual C++ .NET 编程基础

- 彻底的程序员级的开发环境,可视化的设计减少了编程的工作量,几乎可以完成任何设计功能。
- 通过实例全面讲解 Visual C++.NET 的语言知识点和编程技巧,使读者迅速掌握这门编程语言。
- 通俗易懂、易于理解掌握,对 Visual C++.NET 的几乎所有方面都进行了介绍,适用于初、中级读者。

冯 敏 等 编著



清华大学出版社

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



Visual C++ .NET 编程基础

冯 敏 等 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Visual C++.NET 是微软公司推出的新一代可视化开发工具。本书通过大量实例,深入浅出地介绍了 Visual C++.NET 的使用与编程。全书共 8 章,主要包括:Visual C++.NET 开发环境和菜单体系、C++ 语言、Windows 编程机制、应用程序的构建及分析、程序界面设计、屏幕绘制、对话框及常用控件设计等。

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

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: 冯敏 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: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 20.75 字数: 499 千字

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

书 号: ISBN 7-302-04956-4/TP·2925

印 数: 0001~6000

定 价: 29.80 元

前 言

Visual C++ .NET 是 Visual Studio .NET 家族的一个重要成员, Visual Studio .NET 家族还包括 Visual Basic .NET、Visual C# .NET 等开发工具。根据习惯, Visual Studio .NET 和 Visual C++ .NET 又被称为 Visual Studio 7.0 和 Visual C++ 7.0。

Visual C++ 是当今最流行的软件开发工具之一。Visual C++ .NET 增加或增强了许多功能, 比如: 扩展了 Class View 的功能; 新增了进行相关性检查时的文件排除; 支持标志编程; 程序库进行了大量更新和扩充; 等等。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++ .NET 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。Visual C++ .NET 更加适应当今计算机网络化、运行速度快以及加强数据传输的趋势, 是软件开发的首选工具。

本书通过大量实例, 深入浅出地介绍了 Visual C++ .NET 的使用与编程。全书共 8 章, 主要内容包括: Visual C++ .NET 开发环境和菜单体系、C++ 语言、Windows 编程机制、应用程序的构建及分析、程序界面设计、屏幕绘制、对话框及常用控件设计等。

本书通俗易懂、易于理解掌握, 对 Visual C++ .NET 的几乎所有方面都进行了介绍, 即使是初学者也能够轻松入门; 对于使用过 Visual C++ 以前版本或其他结构化语言的读者来说, 从本书中不仅可以尽快了解到 Visual C++ .NET 的新增功能, 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++ .NET 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、校对等工作的还有张思明、吴迪、曹立勋、宫小东、吴清萍、安浦、方克志、包利群、李国文、汪浩、金立言、陈彬、李高云、张明翔、刘瑞芳、陈建平等人。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++ .NET 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++ .NET 的新特性	2
1.2.1 改进的开发环境	2
1.2.2 增强的语言功能和工具	2
1.2.3 标志编程(Attributed Programming)	3
1.2.4 受控代码和目标 COM +	3
1.2.5 更新的程序库和示例	3
1.3 Visual C++ .NET 集成开发环境简介	3
1.3.1 先进的窗口管理技术	4
1.3.2 强大的编辑功能	5
1.4 使用 Visual C++ .NET 集成开发环境	6
1.4.1 Visual Studio 起始页	6
1.4.2 方案管理器	10
1.4.3 Class View	13
1.4.4 Resource View	15
1.4.5 属性窗口	15
1.4.6 工具箱	17
1.4.7 编辑器	28
1.4.8 服务器管理器	29
1.4.9 对象浏览器	31
1.4.10 任务清单窗口	35
1.4.11 宏管理器	37
1.4.12 命令窗口	39
1.4.13 安装工具	40
本章小结	40
第 2 章 Visual C++ .NET 的菜单体系	41
2.1 File 菜单	41

2.2	Edit 菜单	44
2.3	View 菜单	46
2.4	Project 菜单	47
2.5	Build 菜单	50
2.6	Debug 菜单	51
2.7	Tools 菜单	52
2.8	Window 菜单	56
2.9	Help 菜单	57
	本章小结	58
 第3章 掌握C++ 语言		59
3.1	C++ 语言概述	59
3.1.1	C++ 语言的起源	59
3.1.2	C++ 语言的优越性	60
3.1.3	C++ 程序的结构	61
3.2	C++ 符号	62
3.2.1	C++ 注释	62
3.2.2	C++ 标识符	63
3.2.3	C++ 关键字	64
3.2.4	C++ 标点	65
3.2.5	C++ 运算符	65
3.3	数据类型	68
3.3.1	基础数据类型	69
3.3.2	基础扩展类型	73
3.3.3	精确值	74
3.3.4	数字界限	79
3.3.5	数组	81
3.3.6	指针	83
3.3.7	对象引用	85
3.3.8	结构	86
3.3.9	枚举	87
3.3.10	联合	88
3.3.11	类型定义	89
3.4	表达式	90
3.4.1	基础表达式	90
3.4.2	算术表达式	91
3.4.3	赋值表达式	91
3.4.4	前(后)缀表达式	92

3.4.5	关系和逻辑表达式	92
3.4.6	位处理表达式	93
3.5	语句	93
3.5.1	表达式语句和块语句	94
3.5.2	选择语句	94
3.5.3	分支语句	95
3.5.4	循环语句	96
3.5.5	转移语句	97
3.6	函数	99
3.6.1	函数定义	99
3.6.2	内联函数	100
3.6.3	函数重载	100
3.6.4	虚函数	101
3.7	程序启动和终止	102
3.7.1	启动程序	102
3.7.2	使用 wmain	103
3.7.3	命令行参数	103
3.7.4	通配符扩展	104
3.7.5	终止程序	104
3.8	类和对象	106
3.8.1	类的定义	106
3.8.2	对象与类	108
3.8.3	继承和派生	108
3.8.4	运算符重载	110
3.9	C++ 存储类型	111
3.9.1	自动型	111
3.9.2	静态型	111
3.9.3	寄存器型	113
3.9.4	外部型	113
3.9.5	对象初始化	114
本章小结		115
第 4 章	Windows 编程机制	116
4.1	窗口	116
4.2	句柄	117
4.3	Windows 消息	118
4.4	消息驱动	120
4.5	MFC 类库	122

4.5.1	MFC 类库的优越性	122
4.5.2	MFC 类库的基本类结构	123
4.6	MFC 应用程序框架及消息处理机制	129
	本章小结	131
第 5 章	应用程序构建基础	132
5.1	文档与视图	132
5.1.1	文档	132
5.1.2	视图	134
5.1.3	文档/视图结构	136
5.2	框架窗口	137
5.3	创建应用程序框架	137
5.3.1	创建框架的基本步骤	137
5.3.2	结构代码剖析	146
5.4	向应用程序添加程序元素	165
5.4.1	添加类	165
5.4.2	添加成员函数	165
5.4.3	添加成员变量	168
5.4.4	添加事件和事件处理函数	170
	本章小结	171
第 6 章	设计程序界面	172
6.1	界面设计基础	172
6.1.1	设计原则	173
6.1.2	Windows 界面规则	174
6.1.3	界面布局原则	174
6.1.4	用户辅助模型	177
6.2	菜单的设计与使用	177
6.2.1	CMenu 类	178
6.2.2	编辑菜单资源	192
6.2.3	实现菜单命令	195
6.2.4	改变菜单项状态	197
6.2.5	快捷菜单	199
6.3	工具栏的设计与使用	201
6.3.1	CToolBar 类	201
6.3.2	编辑工具栏资源	206
6.3.3	实现工具栏按钮命令	208
6.3.4	创建 IE 风格工具栏	210

6.3.5 调整工具栏的排列方式	212
6.4 状态栏的制作与使用	213
6.4.1 CStatusBar 类	213
6.4.2 使用状态栏	217
6.4.3 在状态栏中输出时间	219
本章小结	221
第7章 绘制图形	222
7.1 图形设备接口(GDI)	222
7.2 设备环境	223
7.2.1 设备环境概述	223
7.2.2 设备环境的获取与释放	224
7.2.3 映射模式	227
7.2.4 设备环境管理类	229
7.2.5 GDI 对象	237
7.3 图形重绘	260
7.3.1 重绘机制	260
7.3.2 OnDraw 函数与重绘	261
7.4 响应用户输入	262
7.4.1 鼠标输入	262
7.4.2 键盘输入	266
7.5 所见即所得的绘制	269
7.5.1 添加鼠标消息处理函数	269
7.5.2 实现绘制	270
7.5.3 维护视图中的图形	272
本章小结	272
第8章 设计对话框接口	273
8.1 对话框基础	273
8.1.1 概述	273
8.1.2 对话框数据交换机制	274
8.1.3 CDialog 类	275
8.1.4 编辑对话框资源	278
8.2 对话框常用控件	280
8.2.1 静态控件	281
8.2.2 按钮	283
8.2.3 文本框	287
8.2.4 组合框	295

8.2.5 列表视图	302
8.3 数据接口	309
8.4 实时调色板	312
8.4.1 对话框条	312
8.4.2 创建对话框条	313
8.4.3 设计对话框条管理类	314
8.4.4 将对话框条载入应用程序	318
本章小结	319

第 1 章 掌握 Visual C++ .NET 开发环境

Visual C++ .NET 是 Visual Studio .NET 家族的一个成员,其他成员包括 Visual C#、Visual Basic 等开发工具。

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的开发难度和开出成本。

Visual Studio .NET 为 Visual Basic .NET、Visual C++ .NET 和 Visual C# 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,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++ .NET 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++ .NET 的集成开发环境。

本章要点:

- ❖ Visual Studio .NET 的安装需求
- ❖ Visual C++ .NET 的新特性
- ❖ Visual C++ .NET 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

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, 或硬盘中存在安装源文件, 否则必需具有 CD-ROM 驱动器。

- 显示器

显示器至少应支持 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 Beta 2 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++ .NET 的新特性

Visual C++ 是当今最流行的 Windows 软件开发工具之一, 而 Visual C++ .NET 是其最新版本, 其中引入了很多新特性。

1.2.1 改进的开发环境

在 Visual C++ .NET 中, 开发环境扩展了 Class View 的功能; 新增了进行相关性检查时的文件排除; 属性窗口提供了更优化的表格, 以查看和修改对象的各种属性; 用户可以在工程属性页中制定工程构建设置; 使用 ClassBar 和属性窗口取代了 WizardBar; 去除了 Database Wizard 等。

1.2.2 增强的语言功能和工具

Visual C++ .NET 的编译器增加了 /CLR、/Fx、/GH 等编译选项; 链接器增加了 /ALLOW-BIND、/IDLOUT、/IGNOREIDL 等链接选项; Dumpbin 增加了 /RAWDATA、/UNWINDINFO、

/IMPORTS、/COMHEADER 等选项; C++ 语言增加了 `_alignof`、`_interface` 等关键字, 以及 `_m64`、`_m128`、`_m128i`、`_m128d` 等数据类型。而且预处理程序也作了相应的更新。

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++、C#、C++ 管理扩展、脚本和 SQL 等语言的交叉调试; 对 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, 或附加于已运行程序的附加程序); Visual C++ 动态错误检查; 缓冲区安全检查等。

1.2.3 标志编程 (Attributed Programming)

标志提供了一种简化 Visual C++ COM 编程的手段, 它在源文件中使用而由编译器解释, 因而类似于 C++ 的关键字。标志不但能够修改代码的行为, 而且还能插入附加的框架代码以实现基本功能, 例如 ActiveX 控件的执行、类创建和数据库命令的格式化等。几乎所有的 C++ 对象都能够使用标志, 例如类、数据成员和成员函数。此外, 标志还能作为独立语句插入到源代码中。

1.2.4 受控代码和目标 COM +

C++ 受控扩展使得由 C++ 编写的代码也依然能使用 .NET 框架服务。Visual C++ .NET 中有两种编写受控代码的方法: C++ 受控扩展 (Managed Extensions) 和 C# 语言。这两种方法都能简化 COM 编程, 并可利用 .NET 通用语言运行库提供的服务。

1.2.5 更新的程序库和示例

与以前版本相比, Visual C++ .NET 的程序库进行了大量更新和扩充, 这包括 ATL、ATL 服务器、MFC、C 的动态库、OLE DB 模板、共享类、标准 C++ 库和原先的 `iostream` 库, ATL 中增加了 `CAdapt` 类、新的控件和宏等。此外, Visual C++ .NET 中还增加了 ATL 服务器、属性、受控 C++ 扩展和 C# 等的丰富实例。

1.3 Visual C++ .NET 集成开发环境简介

Visual Studio .NET 为 Visual Basic .NET、Visual C++ .NET 和 Visual C# 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。

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 .NET 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

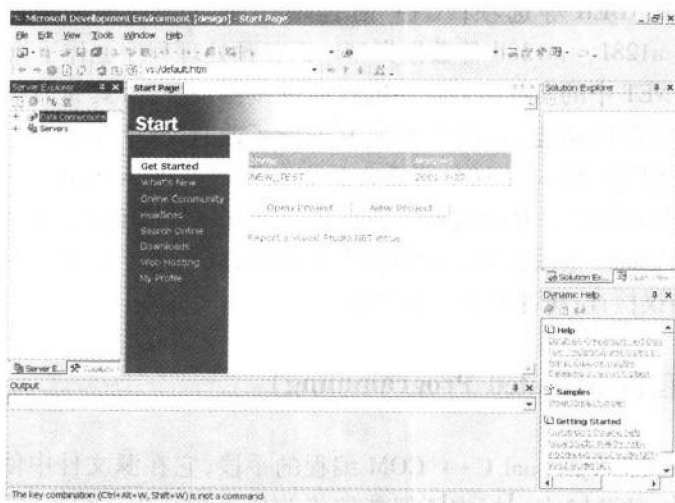


图 1-1 Visual Studio .NET 的启动界面

中的相同。

1.3.1 先进的窗口管理技术

Visual Studio .NET 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 中的典型窗口布局,其中包括集成开发环境中的所有主要窗口。

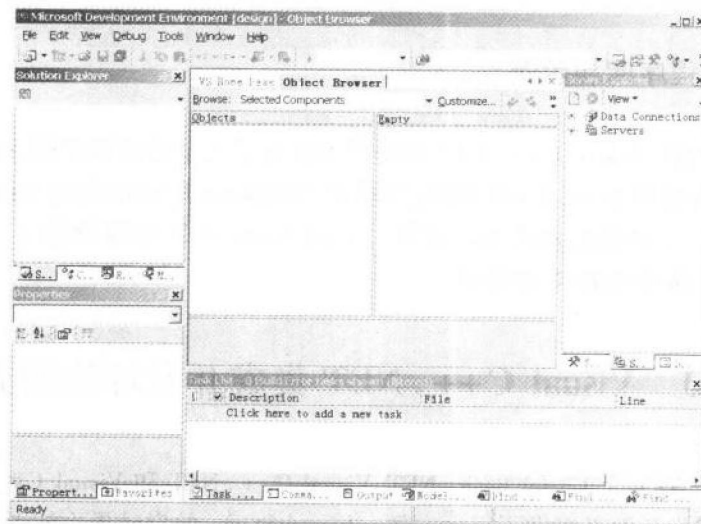


图 1-2 Visual Studio .NET 集成开发环境中的主要窗口

1. 自动隐藏

在窗口标题栏处单击鼠标右键,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,并将其图标排列在集成开发环境边缘,从而节省宝贵的屏幕空间,增加编辑器的可视面积。

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。较之以前版本中必须通过 Window 菜单中的列表选择窗口,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查找/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(Clipboard Ring)

剪贴板环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 热键,或通过工具箱中的 Clipboard 标签,用户能够查看剪贴板环中的内容。Ctrl + Shift + V 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 则将粘贴下一条目,因此重复按下 Ctrl + Shift + V 能够在所有条目中循环。也可以在工具箱中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 Ctrl + - 和 Ctrl + Shift + - 来完成导航。

功能。

1.4 使用 Visual C++ .NET 集成开发环境

本节将向读者介绍 Visual Studio.NET 中常用辅助开发工具的使用。

1.4.1 Visual Studio 起始页

在默认情况下,每次启动 Visual Studio 时,Visual Studio 起始页都显示在用户界面的 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。

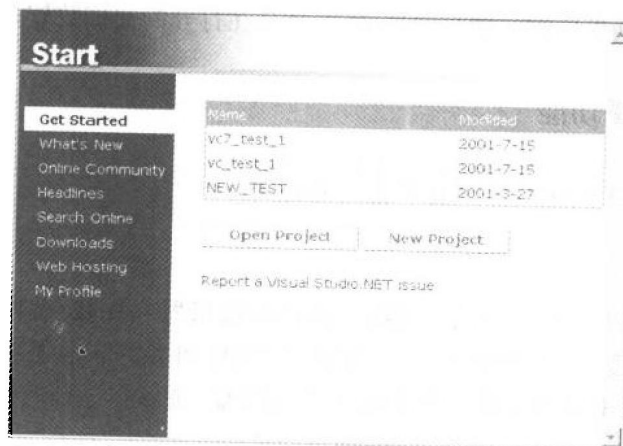


图 1-3 Visual Studio 起始页

Visual Studio 起始页中各项的功能如下:

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,将在后文中详细介绍),如图 1-3 所示。图中的 vc7_test_1、vc_test_1 和 NEW_TEST 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 Microsoft 发送使用意见。

2. What's New

单击 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。

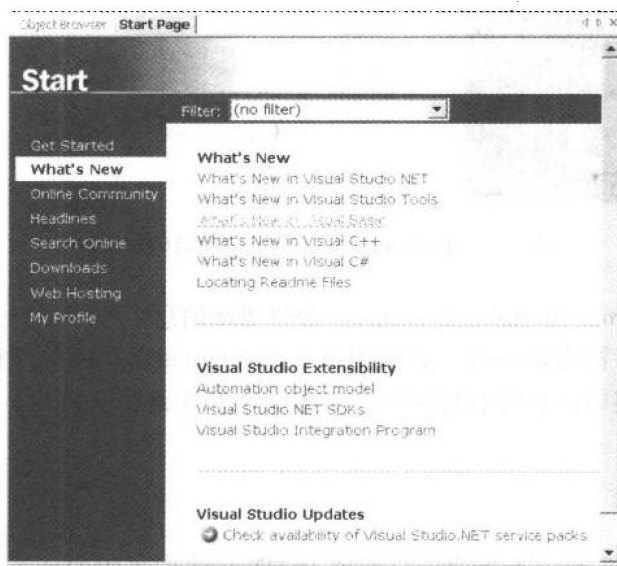


图 1-4 单击 What's New 项后的起始页

- Filter

此下拉列表框中显示了可用的预定义 Help 筛选器。这些筛选器将限制起始页中出现的帮助链接。

- What's New

列出了 Visual Studio .NET 中可用的 What's New 主题。

- Visual Studio Extensibility

列出了自动化模型、SDK 和 Visual Studio 整合程序信息。

- Visual Studio Updates

允许用户链接到 Internet,并显示相关的 Web 页,以下载 Visual Studio 的最新版本。

3. Online Community

单击 Online Community 项,将在起始页中显示到 Web 站点、新闻组和其他类型信息的链接,如图 1-5 所示。这些链接能够帮助用户与其他使用 Visual Studio .NET 的开发者进行