

Framework 程序设计

孙永强 陆 波 杨丽昆 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

随着互联网的蓬勃发展, Microsoft 公司对自身作了全面的调整, 推出了全新的 .NET 策略。Microsoft.NET 策略是将互联网本身作为构建新一代操作系统的基础, 对互联网和操作系统的设计思想进行合理延伸。为了开发 .NET 应用程序, Microsoft 公司推出了 Visual Studio.NET 7.0 集成开发环境。在该版本中, Visual C++.NET 7.0 作为重要的组成部分, 拥有许多重大的更新。本书详细介绍了如何使用 Visual C++.NET 7.0 以及 .NET Framework 类库来开发新型的 Windows 应用程序, 内容由浅入深, 结构清晰, 论证充分。此外, 本书还提供了大量的例程, 使读者能够切实掌握 Visual C++.NET 7.0 各个方面的概念。

本书要求读者具有 C++ 语言的基础知识, 以理解书中的示例代码。通过本书的学习, 读者可以了解 .NET Framework 这种全新的开发平台以及 Visual C++ 针对 .NET 所作的更新。

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: Framework 程序设计

作 者: 孙永强 陆 波 杨丽昆 编著

责任编辑: 许瑛琪

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>

印 刷 者: 北京广益印刷有限公司

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 26.75 字数: 554 千字

版 次: 2001 年 10 月第 1 版 2001 年 10 月第 1 次印刷

书 号: ISBN 7-302-04845-2/TP·2872

印 数: 0001~5000

定 价: 39.00 元

前言

.NET 是 Microsoft 公司今后战略的重点，也是下一代的网络应用基础。Microsoft 公司的所有产品，包括操作系统、办公软件和开发工具等都将转移到 .NET 平台上来。 .NET 所包含的技术全面而先进，必将对今后的软件开发和应用形式产生巨大的影响，因此，及时地了解与掌握它对于每一个开发人员来说都是必须的。

为了促进 .NET 应用程序的开发，Microsoft 公司推出了 Visual Studio.NET 7.0(以下简称为 Visual Studio.NET)集成开发环境。这个集成环境中包含了许多强大的工具，并且支持多种编程语言，如 Visual C++(以下简称为 C++)、C#、Visual Basic、JavaScript、DHTML 等。相对于 Visual Studio 6.0 来说，Visual Studio.NET 的改进是非常彻底的，不仅使用方式完全不同，而且编程概念和提供的类库也是全新的。

虽然 Visual Studio.NET 中包含了多种编程语言，并且所有语言开发出的 .NET 应用程序的效果也是一样的；但是，C++ 本身作为一种功能强大的编程语言，有着顽强的生命力，在 .NET 中起着不可或缺的作用(为了使用 C++ 开发 .NET 应用程序，Microsoft 公司专门对 C++ 语言进行了一些扩展)，而且 .NET 的主打语言 C# 也是从 C++ 演化而来的，它的使用方式和语法与 C++ 基本相同，可以说学会了 C++ 也就掌握了 C#，所以本书以 C++ 语言为基础来介绍如何开发 .NET Framework 应用程序。

本书主要集中于介绍如何使用 .NET Framework 开发 Windows 桌面应用程序，在详细介绍基本概念的同时，提供了大量的针对性很强的示例代码。本书的作者具有多年的编程经验，因此在对概念的理解上和示例代码编写上都充分体现了编程人员的观点，条理非常清晰，重点非常明确。

本书共包括 15 章，内容由浅入深，基本上涵盖了 .NET Framework 中 Windows 应用程序开发方面的知识。

第 1 章介绍如何安装 Visual Studio.NET 7.0 及其新特性。

第 2 章详细地介绍了集成开发环境的各种工具。

第 3 章带领用户逐步地建立一个简单地应用程序。

第 4 章从整体上介绍了 .NET Framework 的基本概念。

第 5 章介绍了 C++ 语言的托管扩展。

第 6 章介绍了 Windows 窗体和对话框。

第 7 章介绍了菜单和状态栏。

第 8 章介绍了如何使用 GDI+ 进行绘图操作。

第 9 章和第 10 章详细而全面地介绍了常用的窗体控件。

第 11 章介绍了几种常见的通用对话框。

第 12 章介绍了基本的文件和目录操作。

第 13 章介绍了打印和打印预览。

第 14 章介绍了如何创建 HTML 帮助并在应用程序中使用它们。

第 15 章介绍了多文档界面的编程并提供了一个完整的体育彩票分析程序。

第 16 章介绍了线程的基本概念。

本书要求读者具有 C++ 语言的基本知识。本书提供了部分示例代码，帮助用户加深对概念和代码的理解。另外，如果用户使用过 MFC，通过本书的学习，可以发现 .NET Framework 所提供的类库比 MFC 更加直观与有效。

目 录

第 1 章 Visual Studio.NET 的新增特性1	2.4.3 查找符号.....37
1.1 Visual Studio.NET 的新特性.....1	2.5 【属性】窗口.....37
1.1.1 .NET Framework.....2	2.6 命令窗口.....39
1.1.2 集成开发环境的新特性.....2	2.6.1 命令模式.....39
1.1.3 Visual Studio.NET 的其他新特性.....6	2.6.2 立即模式.....40
1.1.4 Visual C++.NET 的新增特性.....6	2.6.3 命令介绍.....40
1.2 安装 Visual Studio.NET.....8	2.7 任务列表.....43
1.2.1 系统需求.....9	2.7.1 使用不同的视图显示任务列表.....44
1.2.2 在 Windows 2000 Server 中安装 Visual Studio.NET.....9	2.7.2 结合使用注释和任务列表.....45
第 2 章 Visual Studio.NET 开发环境17	2.8 调试器.....47
2.1 开始页面.....17	2.8.1 控制应用程序的执行过程.....47
2.2 【解决方案资源管理器】.....20	2.8.2 附加到进程.....48
2.2.1 解决方案和项目.....20	2.8.3 断点.....50
2.2.2 使用【解决方案资源管理器】.....21	2.8.4 查看程序的状态.....52
2.3 代码编辑器.....27	第 3 章 创建第 1 个程序55
2.3.1 设置代码的显示格式.....28	3.1 创建程序.....55
2.3.2 迅速定位代码.....29	3.2 检查新建的项目.....56
2.3.3 查找与替换.....30	3.3 生成和运行程序.....59
2.3.4 智能感知.....33	3.4 调试程序.....60
2.4 对象浏览器.....34	3.5 生成发行版应用程序.....63
2.4.1 查看代码.....35	第 4 章 .NET Framework 基本概念65
2.4.2 指定浏览范围.....36	4.1 公共语言运行时环境.....65
	4.1.1 托管执行.....66
	4.1.2 元数据.....67
	4.1.3 应用程序域.....68
	4.2 程序集.....69

4.2.1	程序集清单.....	71	6.3	对话框.....	139
4.2.2	程序集的自定义属性.....	73	6.3.1	创建对话框.....	140
4.2.3	共享名.....	75	6.3.2	获取对话框的返回结果.....	141
4.2.4	版本管理.....	76	第 7 章	菜单和状态栏	149
4.2.5	安全性.....	79	7.1	菜单.....	149
4.2.6	全局程序集缓冲区.....	79	7.1.1	向窗体添加菜单.....	150
4.2.7	创建和使用程序集.....	80	7.1.2	处理命令.....	153
4.3	公共类型系统.....	86	7.1.3	示例程序.....	156
4.4	.NET Framework 类库.....	89	7.2	快捷菜单.....	163
4.5	公共语言规范.....	91	7.2.1	创建快捷菜单.....	163
第 5 章	C++托管扩展	95	7.2.2	示例程序.....	164
5.1	创建示例程序.....	95	7.3	状态栏.....	168
5.2	托管类型.....	98	7.3.1	创建状态栏.....	168
5.2.1	无用单元收集类.....	99	7.3.2	为状态栏添加状态格.....	169
5.2.2	值类型.....	101	7.3.3	示例程序.....	170
5.2.3	托管接口.....	104	第 8 章	GDI+编程	173
5.3	托管数组.....	108	8.1	基本概念.....	173
5.4	字符串操作.....	112	8.1.1	Graphics 类.....	173
5.5	代理.....	114	8.1.2	Paint 事件.....	174
5.6	托管对象的属性.....	117	8.1.3	颜色.....	174
5.7	异常处理.....	119	8.1.4	常用的数据结构.....	177
5.7.1	抛出异常.....	120	8.2	显示文字.....	177
5.7.2	try/catch 代码块.....	121	8.2.1	字体.....	178
5.7.3	__finally 关键字.....	122	8.2.2	画刷.....	178
第 6 章	Windows 窗体	125	8.2.3	示例程序.....	179
6.1	Form 类.....	126	8.3	绘制图形.....	181
6.2	创建窗体.....	127	8.3.1	画笔.....	182
6.2.1	Application 类.....	128	8.3.2	示例程序.....	183
6.2.2	改变窗体的属性.....	129	8.4	显示图像.....	185
6.2.3	改变窗体的大小.....	131	8.4.1	处理位图和图标.....	185
6.2.4	设置窗体的位置.....	132	8.4.2	示例程序.....	187
6.2.5	处理事件.....	133	第 9 章	控件编程(一)	191
6.2.6	向窗体中添加控件.....	135	9.1	概述.....	191
6.2.7	动态调整控件.....	137	9.1.1	控件类的继承层次结构.....	192

9.1.2 控件中的事件处理	193	10.3.3 滑动杆控件	263
9.1.3 Tab 顺序	194	10.3.4 微调控件	265
9.2 按钮	195	10.3.5 示例程序	266
9.3 单选按钮	200	第 11 章 通用对话框	275
9.4 复选框	202	11.1 消息框	275
9.5 标签	204	11.2 文件对话框	276
9.6 编辑框	209	11.3 【字体】对话框	280
9.7 滚动条控件	211	11.4 【颜色】对话框	282
9.8 列表框	213	11.5 示例代码	284
9.9 组合框	215	第 12 章 基本文件操作	307
9.10 图像列表	217	12.1 概述	307
9.11 列表视图	219	12.1.1 Stream 类	308
9.11.1 创建列表视图	219	12.1.2 FileStream 类	309
9.11.2 向列表视图中添加选项	222	12.1.3 读写操作类	310
9.11.3 处理选项的选择	223	12.2 文件和目录类	313
9.11.4 示例程序	223	12.2.1 文件类	314
9.12 树视图	229	12.2.2 目录类	316
9.12.1 创建树视图	229	12.3 示例程序	318
9.12.2 添加节点	230	第 13 章 打印和打印预览	339
9.12.3 操作树视图	231	13.1 PrintDocument 类	339
9.12.4 示例程序	232	13.1.1 设置类的属性	339
第 10 章 控件编程 (二)	239	13.1.2 处理打印事件	342
10.1 逻辑分组控件	239	13.2 【打印】对话框	344
10.1.1 分组框控件	239	13.3 打印预览	346
10.1.2 面板控件	240	13.4 示例程序	347
10.1.3 选项卡控件	241	第 14 章 帮助系统	355
10.1.4 示例程序	244	14.1 使用 HTML Help Workshop 工具	
10.2 工具栏控件	253	创建 HTML 帮助	355
10.2.1 创建工具栏	254	14.1.1 创建帮助项目文件	356
10.2.2 向工具栏添加按钮	255	14.1.2 加入组成帮助系统的文件 ..	358
10.2.3 处理工具栏控件的事件	256	14.1.3 创建帮助目录	359
10.2.4 示例程序	256	14.1.4 创建帮助索引	361
10.3 其他控件	262	14.2 在应用程序中显示 HTML 帮助	364
10.3.1 时钟控件	262		
10.3.2 进度条控件	263		

14.3	HelpProvider 类.....	365	15.4	示例程序——彩票分析程序.....	378
14.4	示例程序.....	366	15.4.1	制作示例程序的帮助文件.....	378
第 15 章	多文档界面.....	373	15.4.2	示例程序的源代码.....	380
15.1	创建多文档界面的主窗体.....	373	第 16 章	多线程编程.....	407
15.2	创建多文档界面子窗体.....	374	16.1	创建并启动线程.....	407
15.3	操作子窗体.....	375	16.2	控制线程的执行.....	410
15.3.1	管理子窗体的布局显示.....	376	16.3	线程之间的同步.....	412
15.3.2	在菜单中显示子窗口列表..	377			
15.3.3	获取当前活动子窗口.....	377			

第 1 章 Visual Studio.NET 的新增特性

随着 Internet 的发展, Microsoft 公司推出了 .NET 战略。Microsoft .NET 策略是将互联网本身作为构建新一代操作系统的基础, 对互联网和操作系统的设计思想进行合理延伸。这样, 开发人员必将创建出摆脱硬件设备束缚的应用程序, 以便轻松地实现互联网连接。Microsoft .NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑。

.NET 对最终用户来说非常重要, 因为计算机的功能将会得到大幅度提升, 同时计算机操作也会变得非常简单; .NET 对开发人员来说也十分重要, 因为它不但会改变开发人员开发应用程序的方式, 而且使开发人员能创建出各种全新的应用程序。

为了配合 .NET 应用程序的开发, 促进 .NET 战略的发展, Microsoft 公司推出了新一代的开发工具: Visual Studio.NET。Visual Studio.NET 是快速建立企业级 Web 应用程序和高性能桌面应用程序的工具。Visual Studio.NET 中包含基于组件的开发工具, 像 C#、Visual Basic、Visual C++, 同时还包含一些简化团队设计、开发和配置解决方案的技术。Visual Studio.NET 完全支持 .NET Framework。

以下各节将介绍 Visual Studio.NET 的新增特性、Visual C++.NET 的新增特性以及 Visual Studio.NET 的安装过程。

1.1 Visual Studio.NET 的新特性

相对于以前的版本, 这一版本 Visual Studio 的变化可以说是天翻地覆。首先, Visual Studio.NET 支持新的应用程序开发平台, 即 Microsoft .NET Framework, 并针对它做了彻底的更新; 其次, Visual Studio.NET 把 Microsoft 的所有语言的开发环境统一起来, 所有语言使用同一套工具, 在同一个集成开发环境中进行开发。

1.1.1 .NET Framework

.NET Framework 是一个建立、配置和运行 Web 服务及应用程序的多语言环境，它是 Microsoft 公司的下一代 Web 应用程序开发平台。它包括以下 3 个主要部分：

- 公共语言运行时(Common Language Runtime, CLR)：这个运行时环境在组件的运行期间和开发期间都起着重要的作用。在组件的运行期间，这个运行时环境管理内存分配、启动和结束线程和进程并且执行安全策略，同时还满足该组件对其他组件的所有依赖需求。在开发期间，这个运行时环境主要用来简化开发工作。
- 统一编程类：.NET Framework 提供一个统一的、面向对象的、层次化的和可扩展的类库集。现今，C++开发人员使用的是 Microsoft 基础类库(MFC)，Java 开发人员使用的是 Windows 基础类库(WFC)，而 Visual Basic 用户使用的则是 Visual Basic API 集。.NET Framework 统一了这些不同的类库，所有的编程语言都可以使用这个类库。通过创建一个公共的跨编程语言的 API 集，.NET Framework 可实现跨语言的继承、错误处理和调试。开发者可以根据需要，自由选择从 JavaScript 到 C++的任何语言进行开发。
- ASP.NET(即 ASP+)：ASP.NET 建立在.NET Framework 编程类库之上，提供了一个 Web 应用程序模型。该模型由一个控件集和一个基本结构组成。其中，ASP.NET 控件封装了通常的 HTML 用户界面元素，如文本框和下拉列表等。这些控件实际运行在 Web 服务器上，但却把它们封装的用户界面元素以 HTML 的形式发送到浏览器上。在服务器上，这些控件展示了一个面向对象的编程模型，向开发人员提供面向对象编程技术拥有的丰富功能。ASP.NET 还提供一些基本服务，比如会话状态管理和进程循环，这些服务进一步减少了开发人员要编写的代码量，并使应用程序的可靠性得到了大幅度提高。

实际上，在.NET 框架中，所有的编程语言，从相对简单的 JavaScript 到复杂的 C++语言，都是等同的，有了.NET 框架后，开发人员便可对选用的任何编程语言使用统一的命令集。语言互用性允许编程人员使用不同编程语言编写各个组件，再将这些组件连接成一个大型的应用程序，还可充分利用互联网和企业内部网的优势，访问整个网络上的 Web 服务。

开发人员可利用.NET 框架自动执行许多任务，逐步地从繁重的重复性编程中解脱出来，以便能在更短的时间内创造出更多更有价值的东西。

1.1.2 集成开发环境的新特性

这一版本的集成开发环境(IDE)作了很大的更新，首先是统一了所有语言的开发环境，

其次是新增了一些工具窗口，并使用全新的窗口管理方式。在这个单一的开发环境中，可以进行 Visual Basic、Visual C++、C#、HTML、XML 等语言的开发。

1. IDE的组成

IDE(Integrated Development Environment, 集成开发环境)的窗口组成和使用方式发生了很大的变化。图 1.1 是 Visual Studio.NET 集成开发环境的启动界面。

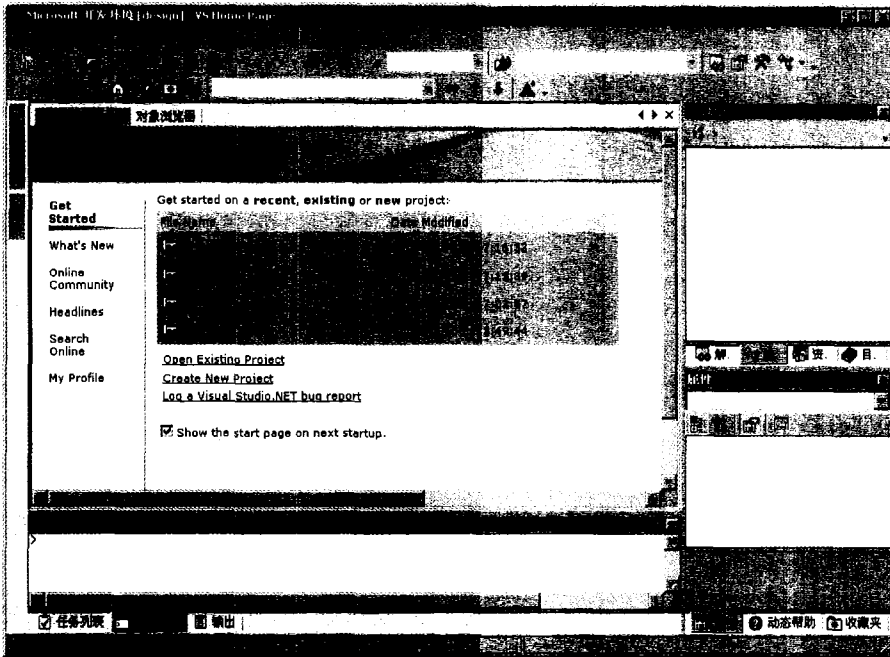


图1.1 Visual Studio.NET集成开发环境的启动界面

启动界面的组成和 Visual Basic 6.0 在某些方面有些类似，主要由以下几个部分组成：

- Visual Studio 起始页面

起始页面是在 Visual Studio 中进行开发工作的起点。用户可以在起始页面中设置集成开发环境的一些参数，包括用户配置文件、活动键盘映射方案和帮助过滤器等。还可以在起始页面中访问最近的、已有的和新的工程项目文件。同时，还可以查看 MSDN 中的最新文章、事件和主题的链接。默认情况下，每次启动 Visual Studio 时，都会启动起始页面。

- 内置的 Web 浏览器

用户可以在集成开发环境中通过使用内置的浏览器直接显示 Web 页面。默认情况下，第 1 次打开 Web 浏览器窗口时会显示起始页面。

- 【解决方案资源管理器】

用户可以在这个管理器中对解决方案进行各种操作，例如设置解决方案的属性，

添加、浏览项目并设置项目的各种属性等。

- **【属性】窗口**

在**【属性】**窗口中可以设置各种开发元素的属性，就像在 Visual Basic 6.0 中一样，而且这种特性可以用于所有集成开发环境支持的编程语言中。

- **【动态帮助】窗口**

【动态帮助】窗口可以根据用户当前的操作和状态动态地显示相应的帮助链接。单击动态窗口中的超级链接，就可以在 IDE 内直接查看帮助信息。

- **【工具箱】窗口**

【工具箱】窗口是一个动态显示的窗口。当鼠标指针停留在工具箱标签上时，**【工具箱】**窗口就会从主窗口的边界上自动滑入。在**【工具箱】**窗口中，根据编辑窗口的状态提供了许多常用的工具。

- **【服务器浏览器】**

该窗口的工作方式和工具箱窗口一样。用户可以在该窗口中浏览和操作所连接的服务器上的各种资源，并且可以浏览和操作数据库连接。

- **【命令】窗口**

【命令】窗口是命令行和 Visual Basic 立即窗口的有机结合。它具有两种模式：命令模式和立即模式。在命令模式下，可以在尖括号(>)的后面输入 IDE 命令来执行相应的 IDE 功能；在立即模式下，可以执行代码语句、设置变量或对变量赋值和对表达式求值等。

- **【输出】窗口**

【输出】窗口主要显示用户在 IDE 中执行各种操作的输出结果，如编译/链接结果和调试结果等。

2. 窗口管理方式

在 Visual Studio.NET 中，窗口的显示和管理方式也有极大的增强，增加了许多新特性来更加有效地组织各种窗口。

- **标签化窗口**

在使用集成开发环境时，将会打开大量的工具窗口和文档窗口，为了有效地组织和显示这些窗口，Visual Studio.NET 采用了标签化的显示方式。当窗口被其他窗口覆盖时，将只显示该窗口的标签。用户可以单击窗口标签方便地在各个窗口之间切换。

- **自动隐藏**

这个特性允许沿着 IDE 的边框最小化工具窗口，比如解决方案浏览器和工具箱窗口，使这些窗口不会占据有效的屏幕空间。通过这种方式可以增加编辑器的可视空间。

- **前后导航**

前后导航使得用户可以像在 Web 浏览器中浏览网页那样，通过前进和后退按钮查

看 IDE 中打开的各个文档窗口。

3. 编辑工具的新特性

Visual Studio.NET 对编辑工具也作了一些增强,此外,Visual Studio 还包含一个集成的对象浏览器,并且支持代码注释。下面列出了一些编辑器的新特性:

- 代码编辑器
 - ◆ Visual Studio.NET 使用统一的代码编辑器编写所有语言的代码。这个编辑器具有几个增强特,比如单词回绕、递增查询、代码轮廓、行编号、折叠文本和书签等。此外,还可以在文档中和文档之间前后导航。
 - ◆ 另一个有用的特性是剪贴板循环队列,这个队列至少可以保存 20 个剪贴板项目。用户可以查看这个队列并从中粘贴。
- HTML 编辑器

HTML 编辑器有两个视图模式:设计视图和 HTML 视图。在设计视图中可以使用所见即所得的方式设计 HTML 页面,即通过拖放控件完成 HTML 页面的设计;在 HTML 视图中,可以直接编写 HTML 代码,充分利用 HTML 语言的强大功能。
- 编辑层叠式样式表

新版本的 Visual Studio 允许直接在 IDE 中编辑层叠式样式表。用户可以使用样式表生成器(Style Builder)编辑样式表,也可以手工输入 CSS 样式规则来进行编辑。
- XML 编辑器

XML 编辑器允许在 IDE 中创建 XML 模式、数据集和文档。可以在编辑 HTML 和 XML 文档时指定 XML 模式定义(XDS, XML Schema Definition)。
- 对象浏览器

对象浏览器现在可以在 Visual Studio 的所有语言上使用。它允许浏览、查询项目和外部库中的对象以及它们的引用情况。

4. 调试工具的新特性

Visual Studio.NET 在所有语言上使用单一的集成化调试器。这个调试器可以进行跨语言调试,既可以调试为公共语言运行时环境编写的应用程序,也可以调试 Win32 本地应用程序,调试方式可以是本地的也可以是远程的。

5. 配置工具的新特性

Visual Studio.NET 中的配置工具提供许多支持 Microsoft Window Installer 技术和简化分布式配置的特性。可以使用这个配置工具在设计期间配置部分应用程序。其中,合并模块功能允许用户创建共享组件的标准化安装。Visual Studio.NET 的图形化配置编辑器使安装程序的创建更加容易。

6. 宏和自动化对象模型工具的新特性

用户可以在 IDE 中记录并回放宏。可以通过宏浏览器、命令窗口或者按键组合来运行宏。自动化对象模型也进行了明显的扩充和更新。用户可以通过编写程序控制 IDE 的每一个元素。

1.1.3 Visual Studio.NET 的其他新特性

除了对集成开发环境所做的更新外, Visual Studio.NET 还新增了一些功能来支持新的应用程序开发方式。

1. Web服务

Web 服务是基于网络的分布式应用程序的基本构造模块, 这些分布式程序独立于平台, 通常使用对象模板和多种语言创建。Web 服务建立在 HTTP 和 XML 等开放的 Internet 标准之上, 并且形成了 Microsoft 公司可编程网络理念的基础。Web 服务可以通过 HTTP 接收请求和 XML 数据。Web 服务不特定于某一种组件技术和对象调用约定, 所以它可以被任何语言、组件模型或操作系统访问。Web 服务是一种应用程序, 它可以使用标准的互联网协议, 将功能纲领性地体现在互联网和企业内部网上。可将 Web 服务视作 Web 上的组件编程。在 Visual Studio.NET 中, 可以使用 Visual Basic、C#或 ATL 服务器创建和引用 Web 服务。

2. XML支持

可扩展标记语言(XML, eXtensible Markup Language)提供了一个描述结构化数据的框架。它是一个国际标准, 是 SGML(Standard Generalized Markup Language 标准通用标记语言)的一个子集, 并且专门针对 Web 进行了优化。XML 是 .NET 战略的核心技术, 它创建出可供任何人从任何地方访问和使用的功能强大的应用程序。它极大地扩展了应用程序的功能, 并实现了软件的不间断传输。Visual Studio.NET 完全支持 XML, 并且提供了简化 XML 和 XML 方案创建工作的 XML Designer。

除了以上所介绍的新增功能外, Visual Studio.NET 中的示例程序和文档也有了很大的变化。新版本提供了两个全新的企业级例程: DuwamishOnline 和 Fitch and Mather 7.0, 这两个例程演示了如何使用 Visual Studio.NET 的各种技术和工具来开发企业级应用程序。此外, 帮助文档的内容和使用方式都针对 .NET 的出现而进行了明显的调整。

1.1.4 Visual C++.NET 的新增特性

Visual C++.NET 主要有两个方面的新增特性: 一个是针对 .NET Framework 所作的增强

和更新；另一个是针对传统 C++ 开发所作的增强和更新。这些更新体现在编译器、链接器和语言本身上，就是增加了一些编译和链接选项，并在 C++ 语言中增加了一些关键字以及对 C++ 所使用的各种代码库/类库做了一些更新。在 Visual Studio.NET 中，Visual C++ 是唯一能够同时编写受管代码和非受管代码的语言。实际上，可以在同一个源文件中混合受管和非受管代码。

另外，为了适应新的开发方式，Visual C++ 的开发环境也发生了一些变化，主要体现在集成度更高和使用更加方便有效方面。针对以上的更新，Visual C++ 所附带的例子程序也做了相应的调整，对已有的例子做了一些修改，例示说明改变的特性，并增加了一些新的例子来说明新增的特性。

1. 受管代码和面向 .NET Framework

Visual C++.NET 的一个主要增强就是提供了一个强大并且简单的方法，来设计面向 .NET Framework 公共语言运行时环境 (CLR) 的应用程序。CLR 提供了一个管理代码执行的运行时环境。所谓受管代码，就是在公共语言运行时环境中运行并使用其提供的服务代码。受管代码是自描述的，具有很多优点，比如跨语言交互、自动内存管理、跨语言异常处理、增强的安全性以及简化的组件交互模型。

.NET Framework 是下一代应用程序的基础。Visual C++.NET 作为 Visual Studio.NET 家族中不可缺少的一员，必然针对 .NET Framework 作了一些更新和增强。使用 Visual C++.NET 可以开发功能强大的 .NET 应用程序。

2. 受管扩展

从 Visual C++.NET 开始，对标准的 C++ 语言进行了扩展以支持受管编程。C++ 的受管扩展主要由一组关键字和属性组成，它使用垃圾收集堆并且面向 .NET Framework 平台。受管代码的一些代码内容和编程过程对于 C++ 开发人员来说是全新的。这些全新内容主要有：受管类型、受管数组、受管扩展中的字符串、受管扩展中的代理和受管对象的属性等。

3. 属性化编程

设计属性的目的是提供一个快速有效的方法来简化 Visual C++ 中的 COM 编程。类似于 C++ 的关键字，属性在源文件中使用并由编译器进行解释。属性可以改变已有代码的行为，甚至可以插入额外的框架代码完成基本任务，如实现 ActiveX 控件、创建类工厂或格式化数据库命令。属性几乎可以应用于任何一个 C++ 对象上，例如类、数据成员和成员函数，也可以把属性作为独立语句插入到源代码中。

4. 事件处理

过去，在 C++ 中，没有处理事件的有效方法。通常使用的方法有窗口消息、事件接口和回调函数等。这些方法比较复杂，并且不同的编程环境使用不同的事件处理方法。现在，Visual C++ 中的事件处理使用统一事件模型 (Unified Event Model)，它允许用户在所有类的事

件处理中使用相同的编程模型。这些类包括：本地 C++类(没有实现 COM 对象的 C++类)、COM 类(实现了 COM 对象的 C++类)和受管的类(使用 `__gc` 关键字声明的 C++类或在受管上下文中声明的 C++类)。

使用统一事件模型可以简化事件的处理，并可以在本地代码和 .NET 代码中使用一致的方式处理事件。

5. 库的更新

Visual C++使用的一些代码库/类库包括 C 运行时库、标准 C++库、MFC 和 ATL 等。在新版本中，对这些库都进行了一些更新和增强。其中 C 运行时库和标准 C++库新增了一些功能和函数。在 MFC 中，对静态转换和消息映射做了一些增强；新增了一些函数以使用超过 2038 年的时间；增加了一些 DHTML 编辑组件和 DHTML 对话框类，比如 `CHtmlEditCtrl`、`CHtmlEditView`、`CDhtmlDialog` 等；ISAPI 部分还增加了支持解析参数列表的类 `CHttpArg` 和 `CHttpArgList`；其他的 MFC 更新包括对 HTML 帮助的更好支持和取消对 DAO 的支持等等。对 ATL 所做的更新与 MFC 类似，主要是增加了一些新类。ATL 的一个全新特性就是 ATL 服务器。ATL 服务器是一组本地 C++类，它允许开发人员创建 Web 应用程序、Web 服务和其他服务器应用程序。此外，OLE DB 模板库也新增了一些类和宏。

从 Visual C++.NET 7.0 开始，修改了一些类，以使它们不再依赖于特定的类库。这些类有 `CPoint`、`CRect`、`CSize`、`CString`、`CStringT`、`CImage`，可以在 MFC 或 ATL 项目中以同样的方式使用这些类。

另外，在 Visual C++中将逐步取消对 `iostream` 库的支持。在使用 `iostream` 函数的时候将会得到警告信息。

以上列出了 Visual C++.NET 中的一些新特性，当然还有许多其他的特性没有列出。从这些新特性中可以看出，在 Visual Studio.NET 中，Visual C++依然是不可缺少的一员，而且功能更加强大，在将来的应用程序开发中仍将起着重要的作用。

1.2 安装 Visual Studio.NET

Visual Studio.NET 本身的安装过程比较简单，只需进行几步简单的操作，安装向导就可以自动完成全部产品的安装。Visual Studio.NET 可以安装在 Windows 98/2001 Me 和 Windows NT 4.0 操作系统上。但是，在不同的操作系统中，所能开发的应用程序的种类和所需进行的设置都有所不同。例如，在 Windows 98/Me 操作系统中，就不能进行 Web 应用程序的开发。如果想要在 Windows 98/Me 环境中开发 Web 应用程序，则必须把开发机器连接到运行 Internet 信息服务器(Internet Information Server, IIS 4.0 或更新版本)的服务器上，并且需要对 IIS 服务器进行正确的配置。而在 Windows NT/2000 环境中则没有这些限制，因为 Visual Studio.NET 支持 Windows NT/2000 中的 PWS(Per Web Server 个人 Web 服务器)和

IIS, 用户可以在同一台机器中开发 Web 应用程序。另外, 在 Windows 98/Me 中也不能进行 Windows 服务应用程序的开发。

本节主要介绍在 Windows 2000 Server 下安装 Visual Studio.NET 的步骤。在其他操作系统中安装 Visual Studio.NET 的过程与此大致相同。

1.2.1 系统需求

Visual Studio.NET 是一个比较庞大的系统软件, 它对机器的要求较高。下面给出了安装和运行 Visual Studio.NET 的系统需求:

- 处理器: 奔腾 II 以上级别的处理器, 速度最小为 450MHz, 建议使用奔腾 III 733MHz 或速度更快的处理器。
- 内存: 最小内存为 96MB, 建议使用 128MB 或更多的内存。
- 硬盘空间: 完全安装 Visual Studio.NET 大致需要 2G 的硬盘空间, 再加上用户开发的项目文件占用的空间, 最少需要 3G 以上的硬盘空间。
- 显示设备: 最小屏幕分辨率为 800×600 像素, 并且使用 16 位颜色。但是, Visual Studio.NET 集成开发环境包含大量的窗口, 在 800×600 像素的分辨率下将会显得非常拥挤。建议使用 1024×768 像素以上的分辨率。
- 光驱: 在安装 Visual Studio.NET 时, 光驱是必须的。如果所需的组件都已安装在硬盘上, 则在开发过程中可以不再使用光驱。
- 操作系统: Windows 98/Me、Windows NT 4.0 和 Windows 2000。建议使用 Windows 2000 或更新版本的操作系统。使用更新和更稳定的操作系统有助于应用程序的开发。

上面的最小需求仅能使 Visual Studio.NET 得以运转。Visual Studio.NET 在运行过程中要使用大量的内存和处理器资源, 而且其众多的工具窗口在方便用户使用的同时也占据了宝贵的屏幕空间。所以要想流畅地开发应用程序, 最好还是满足建议需求。

1.2.2 在 Windows 2000 Server 中安装 Visual Studio.NET

在 Windows 2000 Server 中, 使用 Visual Studio.NET 有许多优点。首先, 可以在同一台机器中建立起和现实使用环境基本相同的开发和测试环境。这样, 就可以在一台机器中模拟客户/服务器环境或浏览器/服务器环境对开发的组件进行测试; 其次, 许多软件只有在服务器环境中才能具备全部功能, 如数据库软件。下面详细介绍在 Windows 2000 Server 中安装 Visual Studio.NET 的步骤:

- (1) 检查是否正确安装了 IIS 4.0 或更新版本。IIS 是开发 Web 应用程序所必需的, 只