

精通GDI+编程



周鸣扬 赵景亮 编著

清华大学出版社

精通 GDI+编程

周鸣扬 赵景亮 编著

清华大学出版社

北 京

内 容 简 介

作为新一代操作系统的图形处理内核, GDI+在 Windows XP 和 Windows Server 2003 操作系统中扮演着极其重要的角色。

GDI+的出现, 是对传统程序员的一种解脱。本书是目前国内少有的全面介绍 GDI+编程的参考书。本书从画笔与画刷、文本与字体、区域与路径等基础知识谈起, 将 GDI+的技术细节一一展开。此外, GDI+的矩阵运算、图像的编码与解码、图像色彩信息校正等深层次知识, 也都能够在本书中找到详细的说明。为了让读者更快地掌握 GDI+编程, 在每一章的内容中, 都配有详尽的程序源代码, 以强化具体的理论阐述。本书是作者长期从事 GDI+编程的经验总结, 所提供的源代码具有一定的代表性。

本书适合于能够熟练使用 C++语言进行程序开发的中、高级程序设计人员阅读使用。另外, 不论读者对 GDI 或 GDI+编程熟悉与否, 都可通过阅读本书全面掌握 GDI+编程的每一个技术细节。同时, 对于使用其他语言如 C#、VB、Delphi 等进行程序开发的读者, 也可以通过本书对 GDI+的基本原理及高级应用有一个全面的认识。

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。

图书在版编目(CIP)数据

精通 GDI+编程/周鸣扬, 赵景亮编著. —北京: 清华大学出版社, 2003.11

ISBN 7-302-07552-2

I. 精… II. ①周… ②赵… III. 窗口软件, Windows—程序设计 IV. TP316.7

中国版本图书馆 CIP 数据核字(2003)第 099455 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 胡伟卷

文稿编辑: 刘金喜

封面设计: 王 永

版式设计: 康 博

印 刷 者: 北京牛山世兴印刷厂

装 订 者: 三河市新茂装订有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 29.75 字数: 705 千字

版 次: 2004 年 2 月第 1 版 2004 年 2 月第 1 次印刷

书 号: ISBN 7-302-07552-2/TP · 5557

印 数: 1 ~ 4000

定 价: 42.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或 (010)62795704

前 言

Windows XP 操作系统的推出,为广大计算机用户带来了全新的用户界面,这其中最主要的原因是 Windows XP 采用了全新的图形界面引擎 GDI+。使用 GDI+图形系统后,由计算机显示的图形和艺术字将具有更平滑的轮廓;用户可以任意使用具有透明度级别的色彩;在调整图片大小时,图片的显示将更加清晰。正如其名称所暗示的那样,GDI+是早期 Windows 版本中包括的图形设备接口 GDI 的继任者。

GDI+是一种新型的图形设备接口,其主要特点在于能够创建全新的用户桌面体系、能够容易完成二维或三维的图形处理,为桌面带来一种新型的数字化图片。Windows XP 或 Windows Server 2003 中的 GDI 是为了与现有的应用程序兼容,但是开发新应用程序的程序员应当使用 GDI+来满足已有的图形需求,因为 GDI+将 GDI 的很多功能进行了优化,而且还提供了新的功能。实际上,GDI+的主要特色就在于通过硬件加速来达到良好的视觉感受。

基于 GDI+技术的方便、快捷和高效率,国内已有不少程序开发人员在程序设计中开始使用 GDI+技术来取代传统的 GDI 编程。本书详细、全面地介绍了 Windows GDI+提供的各种服务,包括位图、画笔、画刷、颜色、坐标空间和坐标变换、填充形状、线条和曲线、文本与字体、图元文件、绘图与绘画、路径、矩形、区域、图像的编码与解码及图像特技处理等方面的知识。本书的内容涵盖了 GDI+大多数的技术细节,可使读者在较短的时间内快速熟练地掌握 GDI+编程技术。

本书共分为 12 章,第 1~2 章介绍了 GDI+程序的开发平台及 GDI+程序的快速入门。第 3~5 章介绍了 GDI+中的画笔、画刷、文本、字体、路径及区域的基本使用。第 6~7 章介绍了 GDI+的坐标变换与色彩变换,以及矩阵的基本编程使用。第 8~9 章介绍了 GDI+中对图像文件的基本使用及对图像色彩的调整处理。第 10 章介绍了在 GDI+中处理图像文件的编码与解码、图像文件格式的转换以及对图形文件属性的修改处理。第 11 章介绍了 GDI+在图像特技处理中的高级编程应用。第 12 章介绍了 GDI+编程的局限与解决方法。

本书中的所有例程在 Windows XP Professional、Visual Studio.NET 及 Office XP 环境下全部编译通过,使用的程序开发语言是 Visual C++.NET。对于本书中的例程,其程序代码同样能在 Visual Studio 6.0 下编译,只是在编译时需要程序的框架及 SDK 的内容做一定的修改,具体过程及方法请参见第 2 章。书中的例程,如无特别说明,均可在单文档(SDI)程序项目的视图类函数 OnDraw 中直接使用。需要本书例程源代码的读者,请与 healthboy@163.net 联系。

本书由周鸣扬、赵景亮执笔,刘昕、周旋、宁可、王青、富晓静、刘阶萍、周健、于伟、龚瑞男、倪晚成、李清、王大智、徐冰、吴淑宁、过承、曾广银、贾超锋、李斌、程陶亚、李建强、赵戡、赵勇、王井东等参与了本书的编写工作。作者 E-mail: liangzair@263.net。

由于创作时间紧迫,加上作者水平有限,书中难免有疏漏之处,欢迎广大读者批评指正。

作 者

目 录

第 1 章 Visual C++ .NET 简介.....	1
1.1 Visual C++ .NET 的新发展.....	1
1.1.1 Visual C++ .NET 的新特性.....	1
1.1.2 Visual C++ .NET 集成开发环境的改进.....	2
1.2 Visual Studio .NET 集成环境的窗口对象.....	5
1.2.1 Visual Studio 起始页.....	5
1.2.2 “解决方案资源管理器”窗口.....	6
1.2.3 “类视图”窗口.....	7
1.2.4 “资源视图”窗口和“属性”窗口.....	8
1.2.5 “工具箱”窗口.....	8
1.2.6 编辑器窗口.....	9
1.2.7 “对象浏览器”窗口.....	10
1.3 Visual C++.NET 的菜单.....	10
1.3.1 “文件”菜单.....	10
1.3.2 “项目”菜单.....	12
1.3.3 “生成”菜单.....	12
1.3.4 “调试”菜单.....	12
1.3.5 “工具”菜单.....	13
1.4 本章小结.....	13
第 2 章 GDI+编程基础.....	14
2.1 GDI+体系.....	14
2.2 GDI+的新特色.....	15
2.3 从 GDI 编程到 GDI+编程.....	17
2.3.1 有关设备环境句柄的概念.....	17
2.3.2 将画刷、路径、图像、字体当做参数.....	19
2.3.3 函数重载.....	19
2.3.4 当前位置.....	20
2.3.5 绘制与填充.....	20
2.3.6 区域的操作.....	20
2.4 GDI+程序的开发与项目分发.....	21
2.4.1 一个简单的 GDI+程序.....	21
2.4.2 有关 UNICODE 编程.....	26
2.5 GDI+编程基本操作.....	28
2.5.1 构造 Graphics 对象.....	28

2.5.2	绘制直线、矩形、曲线和多边形	31
2.5.3	填充区域	38
2.5.4	使用色彩	39
2.5.5	输出文本	43
2.6	本章小结	45
第 3 章	画笔和画刷	46
3.1	在 GDI+中使用画笔	46
3.1.1	画笔的线型	47
3.1.2	画笔的对齐方式	49
3.1.3	画笔的缩放及旋转	50
3.1.4	画笔的线帽属性	52
3.1.5	直线的连接点属性	54
3.1.6	画笔的透明度	57
3.2	在 GDI+中使用画刷	59
3.2.1	单色画刷的使用	59
3.2.2	影线画刷的使用	62
3.2.3	纹理画刷的使用	67
3.2.4	线性渐变画刷	76
3.2.5	路径渐变画刷	93
3.3	本章小结	111
第 4 章	文本和字体	112
4.1	在 GDI+中使用字体	112
4.1.1	理解字体系列	114
4.1.2	使用 GDI+字体	115
4.1.3	列举出系统目前安装的字体信息	118
4.1.4	定制增强型字体选择对话框	121
4.1.5	字体轮廓的平滑处理	134
4.1.6	创建私有字体集合	139
4.1.7	获取字体(系列)尺寸	144
4.1.8	定制文本输出基线	149
4.2	在 GDI+中输出文本	150
4.2.1	测量字符串	153
4.2.2	分栏显示文本	155
4.2.3	字符串的去尾	158
4.2.4	文本的剪裁输出	160
4.2.5	测量文本的局部输出区域	162

4.2.6	格式化文本输出	166
4.2.7	控制文本输出方向	167
4.2.8	设置文本对齐方式	168
4.2.9	使用制表位	171
4.2.10	显示快捷键前导字符	176
4.2.11	使用单色画刷绘制文本	178
4.2.12	使用影线画刷绘制文本	179
4.2.13	使用纹理画刷绘制文本	180
4.2.14	使用渐变画刷绘制文本	181
4.3	本章小结	183
第 5 章	路径和区域	184
5.1	在 GDI+ 中使用路径	185
5.1.1	在 GDI 中使用路径	185
5.1.2	在 GDI+ 中定义路径	186
5.1.3	向路径中添加几何图形	188
5.1.4	开放图形与封闭图形	190
5.1.5	填充路径	190
5.1.6	添加子路径	192
5.1.7	提取子路径的信息	193
5.1.8	访问路径的点信息	196
5.1.9	访问路径的点类型信息	198
5.1.10	标记路径区间	203
5.1.11	修改路径的外观	207
5.1.12	扭曲路径	209
5.1.13	拓宽路径	212
5.1.14	深入理解路径变换的原理	215
5.2	在 GDI+ 中使用区域	218
5.2.1	构造区域	218
5.2.2	计算区域	223
5.2.3	用矩形表示区域	227
5.2.4	击中测试区域	228
5.3	本章小结	230
第 6 章	在 GDI+ 中使用变换	231
6.1	变换的基础	232
6.2	简单的矩阵变换	233
6.3	GDI+ 中的坐标系统	234

6.4	绘图平面的简单矩阵变换	238
6.4.1	绘图平面的平移变换	238
6.4.2	绘图平面的旋转变换	239
6.4.3	平移变换与旋转变换的具体运用	241
6.4.4	绘图平面的缩放变换	243
6.5	变换在文字特效处理中的运用	246
6.5.1	旋转输出文本	246
6.5.2	镜像输出文本	249
6.6	对绘图平面实施复杂的坐标变换	251
6.6.1	使用 Matrix 类表示矩阵变换	251
6.6.2	矩阵的前置与后缀	254
6.6.3	逆矩阵在变换中的运用	257
6.6.4	矩阵的复合变换	258
6.6.5	使用矩阵批量修改点信息	261
6.6.6	二阶矩阵运算	264
6.6.7	矩阵的旋转	266
6.6.8	矩阵的投射变换	268
6.6.9	使用矩阵变换实现文本的异形输出	269
6.7	本章小结	273
第 7 章	GDI+ 的色彩变换	275
7.1	色彩变换的基础	275
7.2	色彩的几种运算方式	279
7.2.1	平移运算	280
7.2.2	缩放运算	281
7.2.3	旋转运算	284
7.2.4	投射运算	289
7.3	色彩的映射	291
7.4	使用色彩变换矩阵实现 RGB 输出通道	292
7.5	本章小结	295
第 8 章	图像的基本处理	297
8.1	图像、位图和图元文件	297
8.2	图像的基本操作	299
8.2.1	图像的打开与显示	299
8.2.2	GDI+ 对图元文件的支持	301
8.2.3	图像的剪裁与缩放	305
8.2.4	使用插补模式控制图形的缩放质量	306

8.2.5 图片的简单旋转	309
8.2.6 图片的反射和倾斜	310
8.2.7 在 GDI+中使用缩略图	313
8.2.8 在 GDI+中使用图片克隆功能	315
8.2.9 在 GDI+中局部缩放图片	316
8.3 本章小结	317
第 9 章 调整图像的色彩信息	318
9.1 色彩校正的基础	318
9.2 启用与禁用色彩校正	320
9.3 设置不同的色彩调整对象	322
9.4 使用色彩配置文件调整色彩信息	328
9.5 图像的 Gamma 曲线校正	330
9.6 设置图片色彩输出通道	332
9.7 使用图片的关键色显示图片	334
9.8 GDI+对阈值的支持	337
9.9 调整图像调色板信息	338
9.10 设置色彩校正的环绕模式和颜色	340
9.11 本章小结	342
第 10 章 图形的编码与解码	344
10.1 图形格式的基础	344
10.2 认识编码与解码	345
10.2.1 PNG 文件的特点	345
10.2.2 PNG 文件的组成	346
10.2.3 PNG 文件的数据块结构	347
10.3 获取图形文件的编码器信息	349
10.4 获取图形文件的解码器及编码参数信息	359
10.4.1 列出系统可用的图片解码器信息	359
10.4.2 处理图形文件的编码参数	360
10.4.3 获取指定图像格式的编码类标识函数	363
10.4.4 还原有关的编码参数	367
10.4.5 将 BMP 文件保存为 PNG 文件	371
10.4.6 将 BMP 文件保存为 JPEG 文件	373
10.4.7 GDI+对 JPEG 文件的保护	375
10.4.8 保存多帧图片	379
10.4.9 从多帧图片文件中读取子图片	382
10.5 获取图像的属性信息	384

10.6	使用图像属性和解码器显示 GIF 文件	388
10.7	GDI+在多格式图像转换程序中的运用	393
10.8	本章小结	397
第 11 章	GDI+图形特技处理编程	398
11.1	使用 GDI+实现图形的淡入淡出效果	398
11.2	GDI+在图像灰度化及伪彩色处理方面的应用	400
11.3	GDI+在图像滤镜制作方面的运用	405
11.3.1	底片(负片)滤镜的制作	405
11.3.2	制作浮雕及雕刻滤镜	407
11.3.3	制作油画滤镜	409
11.3.4	制作木刻滤镜	411
11.3.5	制作强光照射滤镜	412
11.3.6	制作柔化与锐化滤镜	414
11.4	GDI+在图形合成中的运用	417
11.5	本章小结	419
第 12 章	GDI+的使用局限与解决方法	420
12.1	GDI+在游戏程序设计中的运用	420
12.1.1	游戏程序的编程准备	421
12.1.2	定义子弹类	422
12.1.3	定义行星类	425
12.1.4	定义飞机类	431
12.1.5	定义游戏变量及进行事件响应申明	435
12.1.6	初始化全部的游戏元素	437
12.1.7	绘制游戏运行界面	439
12.1.8	GDI+对双缓存技术的支持	442
12.2	GDI+在屏幕抓图程序中的运用	443
12.3	本章小结	447
附录 A	绘图平面类函数列表	448
附录 B	GDI+画笔、画刷类函数列表	452
附录 C	GDI+文本及字体类函数列表	456
附录 D	GDI+图形路径类函数列表	458
附录 E	GDI+图像类函数列表	460
附录 F	GDI+中所有的枚举列表	462

第1章 Visual C++ .NET简介

Visual C++ .NET 是微软公司可视化开发工具系列 Visual Studio .NET 中的产品，是创建 Windows 应用程序最简便、最快捷的开发工具之一。本书中关于 GDI+编程的介绍，使用的编程语言是 Visual C++ .NET 语言，所以，在对 GDI+的编程应用展开讲述之前，本章先对编程语言进行简单介绍。如果读者已经很熟悉 Visual C++ .NET，可以直接跳过此章。

本章介绍 Visual C++ .NET 的基础知识和基本操作。相对于 Visual C++ 6.0，Visual C++ .NET 在编程环境、使用方法等方面都有了不小的变动。本书编程语言使用的是 Visual C++ .NET，所以在开始本书的讲述之前，有必要对 Visual C++ .NET 的一些基本特点及使用方法进行简单介绍，因为“工欲善其事，必先利其器”。

本章的主要内容：

- Visual C++ .NET 的新发展
- Visual Studio .NET 集成环境的窗口对象
- Visual C++ .NET 的菜单

1.1 Visual C++ .NET 的新发展

Visual C++是当今最流行的软件开发工具之一。它强大的功能大大加速了程序员的程序开发速度，提高了程序代码的开发效率。与 Visual C++ 6.0 相比，Visual C++ .NET 增加了一些新的特性。

1.1.1 Visual C++ .NET 的新特性

与 Visual C++ 6.0 相比，Visual C++ .NET 具有以下新特性。

- 引入了两种编写受控代码的方法：Visual C++ .NET 引入了 C++受控扩展和 C#语言两种编写受控代码的方法。这两种方法都使得 COM 编程更加容易，同时也使程序能够使用由 .NET 通用语言运行库提供的支持。由于 C++受控扩展的引入，使得由 C++编写的代码也能够使用 .NET 框架服务。
- 提供了属性编程手段：属性提供了一种简化 Visual C++ .NET COM 编程的手段，它类似于 C++关键字(在源文件中使用，由编译器解释)。属性不但能够修改代码的行为，甚至还能够插入附加的框架代码以实现基本功能，例如，ActiveX 控件的执行、类的创建和数据库命令的格式化等。几乎所有的 C++对象都能够使用属性，例如，类、数据成员和成员函数，而且属性还能作为独立语句插入到源代码中。

- 增强了编译器、连接器等构建工具：编译器增加了/CLR、/Fx、/CH 等编译选项；连接器增加了/ALLOWBIND、/IDLOUT、/IGNOREIDL 等连接选项；Dumpbin 增加了/RAWDATA、/UNWINDINFO、/IMPORYS、/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。此外，预处理程序也做了相应的更新。
- 改进、增强了调试器功能：Visual Studio .NET 为它包括的所有开发语言提供了共同的集成调试器。新的调试器整合了原先 Visual C++ 6.0 和 Visual Basic 6.0 的调试器功能，并添加了许多新特性。新的调试器可以进行 Visual Basic、Visual C++、C#、C++管理扩展、脚本和 SQL 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(可以在单个 Visual Studio 解决方案中运行的多个程序，也可以附加于已运行的程序)，Visual C++动态错误检查，缓冲区安全检查等。
- 提供了更实用、更方便的开发环境：Visual C++ .NET 扩展了类视图的功能，如新增加了进行相关性检查时的文件排除，“属性”窗口提供了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设置工程，使用 ClassBar 和“属性”窗口取代了“类向导”，去除了“Database 向导”等。
- 更新扩充了程序库：Visual C++ .NET 对程序库进行了更新和扩充。例如，ATL(ActiveX Template Library)、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(适配器类)类、新的控件和宏等。
- 提供了更多的 Visual C++实例程序：在 Visual C++ .NET 的帮助中增加了 ATL 服务器、属性、C++受控扩展和 C#丰富的实例。

上述这些新特性使 Visual C++.NET 更加适应计算机网络化和数据传输海量化的趋势。

1.1.2 Visual C++ .NET 集成开发环境的改进

升级后的 Visual Basic .NET 和 Visual C++ .NET 使用同样的用户界面，即 Visual Studio .NET 的集成开发环境(IDE, Integrated Development Environment)。在开发过程中使用同一个 IDE 为开发者提供了很大的方便。Visual Studio .NET 具有源代码创建、资源编辑、编译、连接和调试等功能，其初始启动界面如图 1-1 所示。



图 1-1 Visual Studio .NET 的启动界面

Visual C++ .NET 集成开发环境的改进主要表现在以下几个方面。

1. 采用通用的集成开发环境工具

Visual Studio .NET 中新增的工具为开发者提供了很大的方便。新增的开发工具主要有下面几个。

- Web 浏览器: Visual Studio .NET 的 IDE 能够直接显示网页。要使 Web 浏览器在 IDE 中出现, 只要选择“视图”|“Web 浏览器”命令即可。首次打开 Web 浏览器窗口时, 将显示默认的 Visual Studio .NET 起始页。如果 Web 浏览器被打开, 同时也会出现 Web 工具栏, 其功能与 Internet Explorer 中的 Web 工具栏相同。
- Visual Studio 起始页: 默认情况下, 每次启动 Visual Studio .NET, Visual Studio .NET 起始页都将显示在 Web 浏览器窗口中, 如图 1-1 所示。在 Visual Studio .NET 的起始页中, 用户可以设置 IDE 外观、键盘类型、窗口布局等个性化参数, 还可以设置打开、新建项目等操作的快捷途径。通过起始页, 还能查看到“新增功能”、“标题新闻”、“下载”和“联机搜索”等相关内容。
- 命令窗口: 命令窗口(如图 1-2 所示)具有两个模式——命令模式和即时模式。在命令模式下, 可以在大于号(>)后输入 IDE 的命令名。



图 1-2 IDE 的命令窗口和代码窗口

2. 改进了对窗口的管理

Visual Studio .NET 对窗口管理进行了改进, 使得屏幕一次能够显示大量的代码。图 1-2 右图所示为 Visual Studio .NET 中的窗口布局, 其中包括了集成开发环境的所有主要窗口。

- 标签化文档: 该功能在 IDE 中自动为文档窗口设置标签。例如, 默认情况下“类视图”、“资源视图”和“解决方案资源管理器”使用同一个窗口, 通过切换底部标签可以在此窗口中查看不同视图的内容。在编辑器或设计器中编辑多个文档时, 它们将全部显示于多文档界面(MDI)区域中。虽然一次只能显示一个文档的内容, 但是通过位于顶部(底部)的标签可以方便地浏览其他文档的内容, 如图 1-3 所示。



图 1-3 IDE 中的标签化文档管理

- 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“自动隐藏”命令, 可以实现窗口的自动隐藏。自动隐藏可最小化工具窗口(例如, “解决方案资源管理器”和“工具箱”)并将其排列在 IDE 的边缘, 从而节省屏幕空间, 提高

编辑器的可视范围。

- 前进/后退导航：“前进/后退导航”包含两个方面的内容，一方面是 IDE 允许用户在多个打开的窗口之间进行切换，这个功能可以通过位于代码窗口右上角的左右滚动按钮来实现。另一方面，在同一个被打开的文件中，如果有多处文本被选中(标记)，可以使用工具条中的“向前(后)定位”按钮实现在多个被选中的文本之间进行切换。“前进/后退导航”与 Web 浏览器中的“前进”和“后退”按钮功能类似。

3. 采用应用程序模板

Visual Studio .NET 通过模板为应用程序提供了结构化控制。通过使用企业模板，可降低开发的难度和成本。

Visual Studio .NET 对编辑工具做了进一步的增强，整合了对象浏览器和代码注释。其主要改进如下。

- 代码编辑环境的改进：Visual Studio .NET 中所有的语言都使用统一的代码编辑器。编辑器具有一些新增功能，例如，字回绕、增量搜寻、代码纲要、文本缩进、行编号和书签功能等，用户可以通过“编辑”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之间进行切换。编辑器的另一个强大的功能就是拥有“剪贴板循环(Clipboard Ring)”。“剪贴板循环”能够存储系统剪贴板中被复制或剪切最后 20 个条目。使用 Ctrl+Shift+V 组合键或通过工具箱中的 Clipboard 项目，能够查看“剪贴板循环”中的内容。
- 编辑 HTML：HTML 设计器中有“设计”和 HTML 两个视图，从而保证了网页设计时的灵活性。设计视图为一个 WYSIWYG(所见即所得)设计界面，支持对 HTML 元素进行双击和拖动的操作。在设计视图进行编辑时，可以从工具框的 HTML 项目中将新元素拖到正在设计的网页中，同样也可以通过单击并拖动来改变表单的尺寸，还可以使用快捷菜单为表单添加新行或新列。
- 编辑层叠样式表：在创建样式表时，可以使用“样式”菜单来创建样式。“样式创建器”将给出为不同的 HTML 元素构造不同风格准则的方法。在 CSS 编辑器的主窗口“编辑”表单中，可以使用样式创建器或手动输入 CSS 样式准则。
- 编辑 XML：XML 编辑器允许用户在 IDE 中创建 XML 纲要、数据集和文档。在编辑 HTML 和 XML 文档时，可以制订将使用的 XML 纲要定义。
- 对象浏览器：在 Visual Studio .NET 中，所有的语言使用同样的对象浏览器。这样，就可以方便地查看或检索项目、引用外部库中的对象或成员。

1.2 Visual Studio .NET 集成环境的窗口对象

1.1 节简单介绍了 Visual Studio .NET 集成环境相对于 Visual Studio 6.0 的变化, 本节将详细介绍 Visual Studio .NET 窗口对象。

1.2.1 Visual Studio 起始页

如图 1-4 所示为 Visual Studio .NET 的起始页。该窗口页面的功能如下。

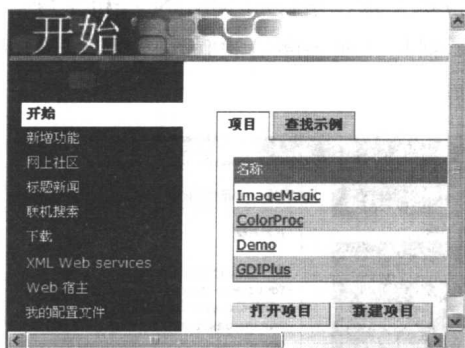


图 1-4 “起始页”窗口

- 最近项目名: 如果已经使用 Visual Studio .NET 创建或编辑过项目, 则“起始页”窗口会显示最近打开过的项目名(如图 1-4 中的 ImageMagic 和 ColorProc 等), 并列列出最近的修改日期。
- 打开项目: 单击“打开项目”按钮, 将显示“打开项目”对话框。
- 新建项目: 单击“新建项目”按钮, 将显示“新建项目”对话框, 该对话框列出了当前 Visual Studio .NET 中已经安装的项目类型, 如图 1-5 所示。

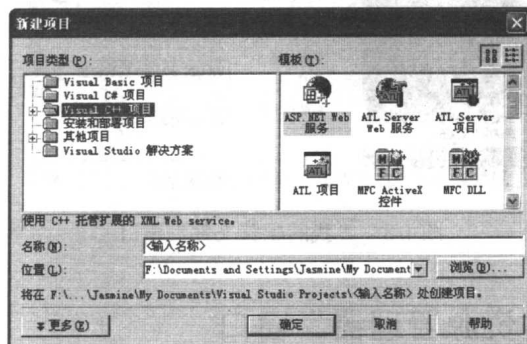


图 1-5 “新建项目”对话框

- 新增功能: 单击图 1-4 所示窗口中的“新增功能”项后, 屏幕将显示图 1-6 所示的“新增功能”窗口, 通过选择选项可查看相应的新增功能, 如图 1-6 所示。



图 1-6 “新增功能”窗口

- 网上社区、标题新闻、下载、Web 宿主和联机搜索：通过选择这些选项可以查看和检索最新的编程技术文章、事件和 MSDN 联机主题。这几项功能为程序开发人员获取网上资源、加强与其他程序员的联系提供了有力的支持。图 1-7 所示为使用“网上社区”时的界面。

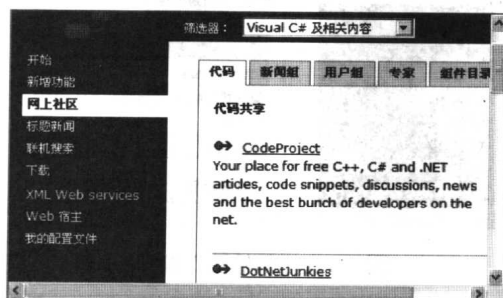


图 1-7 使用“网上社区”

- 我的配置文件：选择该选项，将弹出图 1-8 所示的窗口，在此窗口中可以修改 Visual Studio .NET 的界面风格以及 IDE 行为、“键盘方案”、“窗口布局”等用户参数。

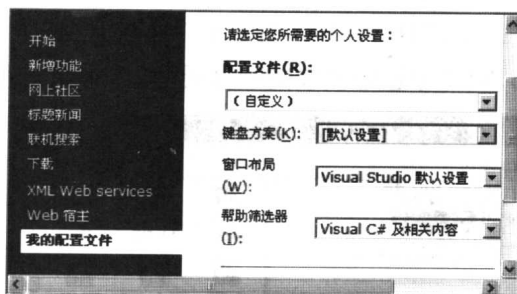


图 1-8 “我的配置文件”窗口

1.2.2 “解决方案资源管理器”窗口

Visual Studio .NET 引入了“解决方案资源管理器”这一新工具，用来管理和监控方案中的项目。选择“视图”|“解决方案管理器”命令，可以打开“解决方案资源管理器”窗口，如图 1-9 所示。

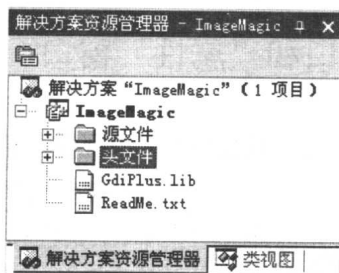


图 1-9 “解决方案资源管理器”窗口

如果说项目是 Visual Studio .NET 应用程序的构造块，那么方案就是应用程序的基础。使用方案和方案管理器能为用户带来很多方便。

一个解决方案中可以包含多个项目，使用“解决方案资源管理器”窗口能够方便地组织需要开发和设计的项目及文件，或者是应用程序或组件。在“解决方案资源管理器”窗口中，还能够显示解决方案的层次结构。

解决方案既包含项目及项目中的条目，还包含“共享方案条目”和“杂项”两种可选文件。在“解决方案资源管理器”窗口中创建或复制某个条目的引用非常简单，只要将该条目拖到另一个位置，同时按下相应的组合键即可。双击“解决方案资源管理器”窗口内的某个文件，则可以打开该文件。

1.2.3 “类视图”窗口

选择“视图”|“类视图”命令，将弹出“类视图”窗口，如图 1-10 所示。“类视图”窗口以树形结构显示当前工程中的所有类，并在每个类中列出成员变量和成员函数。每一个类首先列出带有紫色图标成员函数，然后是带有蓝色图标的成员变量。每个成员图标的左边都有一个标志，以表示成员类型和存取类别的信息。“保护成员”图标旁边的标志为一把钥匙；“私有成员”的标志是一把锁；而“公有成员”图标旁边没有标志。

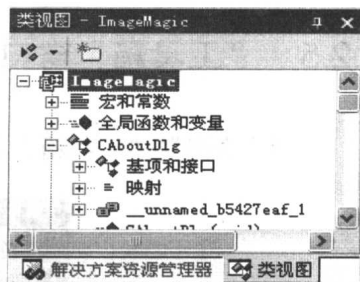


图 1-10 “类视图”窗口

在“类视图”窗口中双击类名，可在主工作区中打开这个类的头文件，显示类的声明；双击某个类的成员，可在代码窗口中显示该成员的定义代码。

在“类视图”窗口中能够很容易地为类添加成员变量和函数，方法很简单：选中某个类，单击鼠标右键，在弹出的快捷菜单中选择“添加函数”或“添加变量”命令，在弹出的“添加成员函数”对话框或“添加变量成员向导”对话框中根据提示添加函数或成员变量即可。