



GDI+ Programming:
Creating Custom Controls Using C#

GDI+ 程序设计

Eric White

杨浩 张哲峰

著
译



清华大学出版社
<http://www.tup.com.cn>



GDI+程序设计

Eric White 著

杨浩 张哲峰 译

清华大学出版社

(京) 新登字 158 号

北京市版权局著作权合同登记号: 01-2002-3175

内 容 简 介

.NET Framework 的出现给编程带来了极大的改变, 图形编程也同样受到了巨大的影响。微软在 .NET 中把 GDI 升级为 GDI+, 它包含了许多命名空间和类, 专门用于 Web 和 Windows 控件中的自定义绘图操作。

本书专门讲述了 GDI+ 编程中与开发自定义控件相关的功能, 重点放在 Windows Forms 自定义控件的创建上, 同时也讨论了 ASP.NET 服务器控件的创建问题。全书共分为 4 个部分: 第 1 部分是对 GDI+ 的用途和结构的概述, 第 2 部分结合代码实例讲述了 GDI+ 编程的具体细节, 如绘图表面、坐标系、钢笔和笔刷、文本和字体、图像和图像操作、路径和区域、变换的处理和打印等。其中涉及了与 GDI+ 在自定义控件中的应用最相关的所有的类、方法、属性和事件。第 3 部分集中讲解了自定义控件的设计和开发, 包括自定义控件的定义、特征, 控件与 VS.NET 的集成, 在控件中如何实现滚动等。第 4 部分讲述了 GDI+ 功能在 Web 应用程序中的使用。如在 Internet 上提供 GDI+ 图像、创建 ASP.NET 服务器控件和使用 GDI+ 创建 Web 服务等知识。

本书适合希望了解如何使用 GDI+ 编写 Windows Forms 和 Web Forms 自定义控件的中、高级 C# 程序员和其他熟悉面向对象语言的开发人员阅读。

Eric White: GDI+ Programming—Creating Custom Controls Using C#

EISBN: 1-861006-31-4

Copyright©2002 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由英国乐思出版公司授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

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

GDI+程序设计/(美)怀特著; 杨浩, 张哲峰 译.—北京: 清华大学出版社, 2002

书名原文: GDI+Programming—Creating Custom Controls Using C#

ISBN 7-302-06091-6

I. G... II. ①怀...②杨...③张... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 090238 号

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.com.cn>

责任编辑: 于平

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 29.25 字数: 748 千字

版 次: 2002 年 12 月第 1 版 2002 年 12 月第 1 次印刷

书 号: ISBN 7-302-06091-6/TP·3636

印 数: 0001~4000

定 价: 58.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前 言

.NET Framework 的发布改变了那些基于 Microsoft 开发应用程序的人员的编程方式。像 C#和 Visual Basic .NET 这样的新型面向对象的语言就是专门为这个新平台创建的语言,所以,我们要更新许多从以前环境中获得的知识,从而把握新的机会。

图形编程领域和其他领域一样也受到了影响。Windows Graphical Device Interface (GDI)非常便于在设备上定制绘图,但.NET Framework 现在提供了 GDI+。GDI+包含许多命名空间和类,可以用作.NET Framework 的一部分,专门用于在 Web 和 Windows 控件中进行定制绘图。GDI+可以有效地用作包装旧 GDI 的一个包装器,提供了易于理解、基于继承的对象模型,供.NET 程序员使用。

GDI+允许程序员进行更多的定制绘图。有 C++/MFC 背景的程序员应很熟悉 GDI+中的许多方式,而且,使用过 Visual Basic 的人会发现, GDI+是开发显示控件的一种可行方式,能扩展.NET Framework 附带的控件库。

.NET Framework 基类库中属于 GDI+的那一部分是非常丰富的,本书不可能解释每一个类。因此,本书采用的方法是集中介绍 GDI+中与程序员使用这些类最相关的那些领域和特征,即开发自定义控件。本书将深入讨论 Windows Forms 自定义控件的创建,在最后的 3 章中则阐述定制 ASP.NET 服务器控件的知识。

0.1 本书读者对象

本书主要针对那些希望使用.NET Framework 提供的 GDI+基类功能的开发人员而编写的。特别是,本书非常适合于那些工作在商业应用环境中的人员,适合于那些提供技术和信息,以完成 Windows Forms 和 Web Forms 编写 GDI+代码的任务,而创建自定义控件的人员。

本书中的例子(包括本书前半部分的 GDI+演示例子,和后半部分自定义控件中的 GDI+应用程序)都是用 C#编写的。中级 C#程序员和有相当经验的、能辨识出面向对象语言的标准结构和特征的开发人员都可以读懂它们。

但是,与讲述 GDI+开发和.NET 中自定义控件的图书相比,本书中的例子选择 C#语言并不是要关注的重要方面。实际上,这里论述的 GDI+技术都是与语言不相关的,使用其他.NET Framework 语言,例如 Visual Basic .NET 和 Visual C++ .NET,都可以应用 GDI+技术和相应的语言知识。



0.2 本书内容

在结构上看,本书分为4部分。第1部分只包含一章,即第1章,它简要介绍了GDI+的功能和结构,并给出了在.NET应用程序中开发和使用自定义控件的一些语境。在本章的最后,用一个小例子说明Visual Studio .NET环境的主要特征,本书将一直使用这些特征。

本书的第2部分包含第2~10章,主要论述了GDI+编程的核心内容。我们将讨论GDI+应用程序在自定义控件方面最重要的类、方法、属性和事件,并进行了许多描述和说明:

- 第2章介绍了GDI+的基本知识,例如绘图表面和坐标系。其中包括对Graphics对象的介绍,该对象封装了绘图表面。我们还要阐述绘图表面的3种类型(屏幕、位图和打印机)。Graphics对象提供了在绘图表面进行绘图和描绘的方法,以及管理许多其他特性的方法,例如,测量单位、转换、剪切和失效等。因此,Graphics类是学习本书中许多内容的基础。
- 第3章介绍钢笔和笔刷。GDI+在绘图和填充上是有区别的,这种区别被封装到Pen和Brush类中。这两个类都是基本图形工具,本章将对它们的多种功能进行论述。
- 第4章则把注意力转移到文本和字体上面——介绍GDI+如何以编程的方式控制图形中文本的内容和样式。我们将详细讨论Font和FontFamily类,理解它们如何与图形的其余部分相关联。
- 第5章讨论图像和图像的操作,以及如何创建和保存图像。
- 第6和第7章则论述路径和区域这两个重要论题,它们封装在GDI+的GraphicsPath和Region类中。首先讨论的是这两个类如何进一步突出绘制轮廓和填充区域之间的区别,与以前相比,它们还可以处理更不规则的图形,能够把一组图形组合在一起,进行更高效的处理。有了这个基础后,将解释这些对象如何用于剪切和失效等操作,创建有效而强大的效果。
- 第8章说明如何在GDI+中进行转换。转换、旋转和变形在自定义控件的开发中并不常用,但本章将主要介绍一些有趣的技术。
- 在前面的几章中讨论完基于屏幕的显示后,第9章就讨论打印的主题。我们将论述如何控制屏幕上显示的内容和打印机输出到页面上的内容之间的差别。
- 第10章描述了如何建立其他坐标系,这对自定义控件的开发非常有帮助。这涉及到Graphics类语法中的微妙变化。我们不能继承Graphics类本身(因为它定义为sealed),但可以使用设计模式(称为装饰图案)“修改”其行为,其中,可以定义自己的类,该类的方法签名与Graphics类中的方法相同。

后面的章节开始专门讨论自定义控件的结构体系和开发。本书的第三部分介绍自定义控件的概念,讨论决定把什么特性内置于应用程序中时需要考虑什么问题,以及用户界面的要求:

- 首先,第11章复习自定义控件,包括它的定义、特性以及与组件的比较。我们将从一定高度介绍自定义控件理论,之后演示几个自定义控件的例子,说明它们的行为,例如属性、焦点、事件、从自定义控件中派生和复合自定义控件。
- 第12章解释的是为自定义控件建立功能的一些方式,这样,使其他开发人员更容易在

他们的代码中使用这些自定义控件。我们还将专门论述如何更好地把自定义控件与 Visual Studio .NET 集成起来。例如创建模式框和下拉框来编辑属性。所有这些特性都会使开发人员在设计期间更容易使用自定义控件。

- 在建立自定义控件时，常常需要一个比实际空间更大的虚拟空间。第 13 章将解决这个问题，介绍在自定义控件中实现滚动时可以使用选项。我们将阐述在 Windows Forms 类中对滚动的内置支持，说明为什么通过自定义控件来实现滚动比较合适。我们还会解释如何实现平滑的滚动，使读者对应用程序有更好的理解。
- 第 14 章将描述另一个用户交互特性：鼠标。主要介绍的是如何开发控件，使之在以不同的方式使用鼠标时执行不同的功能，包括单击测试、鼠标事件例程、拖放滚动。在本质上看，图形用户界面是可见的，所以我们还要介绍应用程序中的光标以及如何处理它们。

在本书的最后一部分，主要介绍如何在 Windows Forms 和基于 Web 的环境下使用 GDI+ 功能。本书的最后 3 章将讨论下述情形：开发人员需要使用 ASP.NET 的功能来处理 GDI+ 图像、在 Web 上创建自定义控件，利用 Web 服务技术。我们不深入介绍这些内容，但提供一个在这些不同的情形下使用 GDI+ 的架构。

- 第 15 章讨论的是通过 Internet 处理 GDI+ 图像。首先，从 Web 服务器的文件系统中抓取一个图像，使用 GDI+ 代码给客户应用程序视图(即一个 ASP.NET 页面)提供该图像。我们还会介绍其他 Web 图像处理技术，例如混合 HTML 和动态图像，用 GDI+ 修改图像，最后介绍了创建图表的内容。
- 第 16 章论述如何创建一些定制的 ASP.NET 服务器控件，加深对自定义控件的理解。这些控件在本质上类似于 Windows Forms 自定义控件，但执行方式完全不同。本章会演示一些 ASP.NET 用户控件的例子，包括一个按钮用户控件，和一个创建和处理小图标的例子，我们还将讨论对设计时特征的支持。
- 在本书的最后一章即第 17 章中，将讲述如何在 Web 服务中使用 GDI+，如何返回图像和图像数据。首先简要论述 Web 服务背后的理论，接着介绍 3 个逐渐复杂的、使用 GDI+ 的 Web 服务的例子。先介绍一个简单的 Hello World 服务，接着是一个从 URL 中返回图像的 Web 服务，最后是一个从图像中返回数据，绘制一个饼图的 Web 服务。

0.3 使用本书需要的条件

要运行本书中的例子，需要：

- 一个合适的操作系统。在编写本书时，Microsoft 推荐使用安装了 Service Pack 2 的 Windows 2000 (Professional, Server 或 Advanced Server 版本)或 Windows XP Professional 版本。
- .NET Framework SDK。

本书中的例子是用 Visual Studio .NET IDE 开发的。没有 VS.NET IDE 也能开发这些例子，但本书没有介绍这方面的知识。

读者还可以从 <http://www.wrox.com> 上下载本书所有例子的完整源代码(参见下面的“客户



支持和反馈”一节)。无论是从这些源文件中剪切并粘贴代码，还是自己键入代码，源代码都提供了一种检查自己的代码中错误的好方式。

0.4 客户支持和反馈

我们总是希望倾听读者的意见，我们想知道您对本书的看法，读者喜欢哪些，不喜欢哪些部分，这些信息将有助于我们下一次做得更好。如果您有什么意见和建议，请向 feedback@wrox.com 发邮件。但是，请一定要在您的信中注明本书的 ISBN 和书名。

0.4.1 源代码与更新内容

当学完本书中介绍的各示例时，读者也许决定自己手工输入全部的示例代码或使用本书附带的源代码。许多读者愿意采用前者，因为这样做是熟悉例子中使用的编码技巧的好方法。

无论读者是否想输入代码，手边有一份源代码的拷贝都是很有用的。如果读者愿意输入代码，可以通过作者所提供的对应文件来验证所得到的结果——如果认为可能输入了一个错误，可以把这些文件作为首选的参考。如果读者不喜欢自己输入代码，则可以从本书的 Web 站点下载源代码。无论使用任何一种方式，均有助于升级与调试。

本书使用的所有源代码都可通过 <http://www.wrox.com/> 下载。在登录到该站点后，只需使用 Search 工具或使用书名列表就可以找到本书。接着单击本书信息页面上的 Download Code 链接，就可以获得所有的源代码。

可以从该站点上下载的文件使用 WinZip 进行压缩。在把附件保存到硬盘的一个文件夹中时，需要使用解压缩文件如 WinZip 或 PKUnzip 对该文件解压缩。在解压缩时，代码常常放在各自的章节文件夹中。在开始解压缩过程时，确保解压缩软件设置为使用文件夹名。

0.4.2 勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。请给 support@wrox.com 发电子邮件，我们会检查您的信息，如果是正确的，就把它传送到该书的勘误表页面上，或在本书的后续版本中采用。

要在网站上找到勘误表，可以登录 <http://www.wrox.com>，通过 Advanced Search 工具或书名列表查找本书，然后在本书的信息页面上，单击 Book Errata 链接。在这个页面上，可以查看已经提交并通过编辑检查的所有勘误，还可以单击 submit errata 链接，告知我们您找到的错误。

0.4.3 技术支持

如果您想直接向了解该书内容的专家询问问题，可以向 support@wrox.com 发送电子邮件，通常的电子邮件包括下面的内容：

- 在“主题”栏填写问题的名称、最后 4 位 ISBN 号码(本书是 6314)以及页码。

- 在邮件正文中包含您的姓名、联系信息和问题。

我们不会向您发送垃圾邮件。但我们需要您提供细节以节省您的时间和我们的时间。当您发送电子邮件时，它将经过下面的支持链。

客户支持——您的消息会发送到一位客户支持成员处，他是读取消息的第一人。他们具有关于最常见的问题的答案，并会立即回答常见的问题。他们会回答关于本书和网站的常见问题。

编辑支持——更深一层的问题将发送到负责该书的技术编辑处。他们有关于编程语言以及使用特定产品的经验，可以回答详细的技术问题。解决了问题之后，编辑会将勘误表发送到网站上。

作者支持——最后，如果编辑无法回答问题(这种情况非常罕见)，他就会请求作者帮助。我们希望使作者专心写作，但是我们也非常高兴给他们发送一些问题。所有 Wrox 作者可以提供他们的图书的技术帮助。他们会向客户和编辑发送电子邮件，回答问题，这样所有的读者都可以从中受益。

注意：

Wrox 支持过程只能提供直接与已出版的图书相关的问题。对于超出此范围的问题可以通过 <http://p2p.wrox.com/> 论坛的团体列表来提供支持。

0.4.4 p2p.wrox.com

P2P 邮件列表是为作者和程序员之间的讨论而设立的。我们在邮件列表、论坛和新闻组中提供“程序员到程序员的支持”(programmer to programmer support)，还包括一对一的电子邮件支持系统。如果把问题发送给 P2P，就可以相信，您的问题不仅仅是由支持专家解答，而且还要提供给我们邮件列表中的许多 Wrox 作者和其他业界专家。在 p2p.wrox.com 上，可以从许多不同的列表中获得帮助，不仅在阅读本书时获得帮助，还可以在开发应用程序时获得帮助。

要订阅一个邮件列表，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，从左侧的菜单栏中选择合适的列表。
- (2) 单击想加入的邮件列表。
- (3) 按照指示订阅和填写电子邮件地址和密码。
- (4) 回复接收到的确认电子邮件。
- (5) 使用订阅管理器加入更多的列表，设置自己的邮件选项。

目 录

第 1 章 GDI+和自定义控件概述	1
1.1 GDI+和.NET	1
1.1.1 GDI+概述	2
1.1.2 GDI+命名空间	5
1.2 自定义控件	6
1.2.1 建立 Windows Forms 自定义控件	6
1.2.2 Web Forms 自定义控件	7
1.2.3 为 Windows Forms 和 Web Forms 创建共用代码	8
1.3 第一个例子	9
1.4 小结	21
第 2 章 绘图表面	22
2.1 绘图表面概述	22
2.1.1 基于光栅的绘图表面和基于矢量的绘图表面	22
2.1.2 绘图表面的特性	24
2.1.3 颜色的结构	25
2.1.4 在不同环境下的绘图表面	28
2.1.5 绘图表面的小结	30
2.1.6 Graphics 类	31
2.2 GDI+坐标系	34
2.3 小结	44
第 3 章 钢笔和笔刷	45
3.1 用 Pen 对象绘图	45
3.1.1 控制钢笔的属性	47
3.1.2 Pens 类	54
3.2 用 Brush 对象填充	54
3.2.1 使用 Brush 类	55
3.2.2 Brushes 类	68
3.2.3 从笔刷中创建钢笔	68
3.2.4 性能的注意事项	69
3.3 小结	69



第4章 文本和字体	71
4.1 文本和字体概述	71
4.1.1 在 GDI+中使用字体	73
4.1.2 绘制基本文本	76
4.2 格式化文本	78
4.2.1 字体样式	79
4.2.2 行、对齐和方向	80
4.3 处理字体	89
4.3.1 查找可用的字体	89
4.3.2 字体的规格和精确的文本位置	90
4.3.3 文本的质量	96
4.4 小结	102
第5章 图像	104
5.1 图像处理概述	105
5.1.1 图像、位图和元文件类	105
5.1.2 不同类型的位图	107
5.1.3 编码器和解码器	109
5.1.4 像素格式	111
5.2 显示图像	113
5.2.1 查看大小和分辨率	114
5.2.2 改变位图的分辨率	116
5.2.3 图像调整大小和插补	117
5.3 处理图像	119
5.3.1 修剪图像	119
5.3.2 变形、翻转和旋转图像	120
5.3.3 复制图像	122
5.3.4 获取图像的略图	123
5.4 创建和绘制图像	124
5.4.1 创建新位图	125
5.4.2 双倍缓存	127
5.4.3 处理图像中的 Alpha 成分	128
5.4.4 获取和设置像素	131
5.4.5 为整个图像设置 Alpha	133
5.5 播放动画	136
5.6 把 BMP 转换为 JPEG	137
5.7 小结	139

第 6 章 GraphicsPaths 和 Regions	141
6.1 什么是 GraphicsPath 和 Region	141
6.1.1 GraphicsPath 类	142
6.1.2 Region 类	150
6.2 小结	157
第 7 章 裁剪和失效	159
7.1 裁剪区域	160
7.1.1 剪切裁剪区域	160
7.1.2 通过裁剪区域选择性地重新绘制	166
7.1.3 像素级别的裁剪	175
7.2 失效	176
7.2.1 在控件中使用状态	177
7.2.2 失效的工作方式	178
7.3 裁剪的技巧	181
7.4 小结	182
第 8 章 变换	183
8.1 GDI+中的变换	183
8.1.1 坐标系和变换	184
8.1.2 通用变换	185
8.2 失效和裁剪	197
8.3 变换和自定义控件	198
8.4 小结	198
第 9 章 打印	199
9.1 打印进程	200
9.1.1 GDI+中与打印相关的类	200
9.1.2 如何打印一个文档或进行打印预览	201
9.1.3 使用打印类	202
9.1.4 一个简单的打印示例	203
9.2 选择度量单位	213
9.2.1 默认的图形单位	215
9.2.2 打印示例 Ruler	215
9.3 打印度量	217
9.3.1 输出打印度量数据	218
9.3.2 控制打印机绘图表面	220
9.3.3 用户修改设置的位置	225
9.4 小结	226



第 10 章	一种可选择的坐标系	227
10.1	定义问题	228
10.2	轮廓模型坐标系	230
10.3	创建 3D 效果	236
10.4	小结	240
第 11 章	Windows Forms 自定义控件的结构和设计	241
11.1	基本原理	241
11.1.1	什么是组件	242
11.1.2	什么是自定义控件	244
11.2	创建一个 Windows Forms 自定义控件	245
11.2.1	简单的自定义控件示例	246
11.2.2	使用一个自定义控件	248
11.2.3	创建一个测试应用程序	250
11.3	向自定义控件中添加属性	251
11.3.1	使用属性的自定义控件示例	251
11.3.2	创建一个测试应用程序	255
11.4	属性的默认值	256
11.5	焦点	260
11.5.1	焦点提示	261
11.5.2	实现一个带焦点控件的步骤	261
11.5.3	带焦点的控件示例	262
11.5.4	创建一个测试应用程序	270
11.6	生成事件	272
11.6.1	委托和事件概述	272
11.6.2	事件示例	273
11.6.3	创建一个测试应用程序	276
11.7	派生一个现有控件	277
11.7.1	NumberTextBox 示例	277
11.7.2	创建控件	278
11.7.3	创建一个测试应用程序	282
11.8	复合自定义控件	283
11.8.1	DataButtons 示例	284
11.8.2	创建一个测试应用程序	287
11.9	设计组件和自定义控件	287
11.9.1	非过程结构的优势	288
11.9.2	通过此方法设计组件	289
11.9.3	设计事件	289

11.10 小结	290
第 12 章 设计时支持	291
12.1 基本概念	291
12.1.1 在 Properties 窗口中编辑属性	291
12.1.2 在设计视图中编辑控件	293
12.2 为属性和事件分类	294
12.3 在 Properties 窗口中改进编辑操作	295
12.3.1 NumberRange 结构	297
12.3.2 TypeConverter 派生的类	299
12.3.3 控件 NumberTextBoxA	303
12.3.4 创建一个测试应用程序	305
12.4 调试设计时代码	305
12.5 创建一个模态对话框属性编辑器	306
12.5.1 控件 NumberTextBoxB	307
12.5.2 NumberRangeDialog	307
12.5.3 NumberRangeEditor	310
12.5.4 创建一个测试应用程序	313
12.6 创建一个下拉属性编辑器	313
12.6.1 NumberRangeDropDown 控件	314
12.6.2 NumberRangeEditor	317
12.6.3 创建一个测试应用程序	319
12.7 实现一个自定义设计器	320
12.7.1 SimpleBlankControl	321
12.7.2 SimpleLineControl	323
12.8 小结	329
第 13 章 滚动	330
13.1 构建一个滚动示例	330
13.1.1 设置滚动条位置	332
13.1.2 在视区绘图	333
13.2 滚动方法	334
13.3 把一个非滚动控件放置在 Panel	336
13.4 平滑滚动	337
13.4.1 SmoothScrollableControl 示例	337
13.4.2 测试此控件	344
13.5 小结	345



第 14 章 鼠标事件和指针	346
14.1 鼠标事件	347
14.2 鼠标指针	347
14.2.1 标准的鼠标指针	348
14.2.2 CursorsExample 示例	348
14.2.3 自定义鼠标指针	350
14.3 单击测试以及在鼠标事件中绘图	351
14.4 鼠标事件的路由选择	356
14.4.1 RoutingMouseEvents 示例	357
14.4.2 处理 CaptureChanged 事件	361
14.5 GDI 和 BitBlt	363
14.5.1 为什么使用 GDI	364
14.5.2 使用 BitBlt()	365
14.5.3 简单的 BitBltExample 示例	366
14.6 在拖动时绘图	367
14.7 拖动-滚动	375
14.7.1 演示“摆动鼠标程序错误”	375
14.7.2 排除摆动程序错误	384
14.8 小结	386
第 15 章 ASP.NET 中的 GDI+图像	387
15.1 Internet 信息服务(IIS)	388
15.1.1 文档	389
15.1.2 安全和虚拟目录	390
15.2 读取并传递图像	391
15.2.1 流	393
15.2.2 从远程 Web 服务器读取图像	393
15.3 混合 HTML 图像和动态图像	395
15.4 修改图像	398
15.5 创建一个新的图像	403
15.6 小结	410
第 16 章 创建自定义 ASP.NET 服务器控件	411
16.1 自定义控件	411
16.2 创建一个 ASP.NET 用户控件	412
16.2.1 用户控件属性	414
16.2.2 自定义按钮用户控件	415
16.3 创建一个缩略图控件	421
16.3.1 制作一个缩略图	421

16.3.2 创建一个自定义控件	423
16.4 创建一个缩略图的图片库	426
16.5 设计时支持	428
16.5.1 创建一个 Toolbox 图标	429
16.5.2 其他的设计时特性	430
16.6 小结	431
第 17 章 Web 服务和 GDI+	432
17.1 什么是 Web 服务	432
17.2 如何在 Web 服务中使用 GDI+	433
17.3 一个简单的 Web 服务	433
17.4 返回一个图像 URL	436
17.5 返回图像数据	440
17.6 小结	444

第1章 GDI+ 和自定义控件概述

编写图形代码是计算机编程领域中最令人愉快的工作之一。我们可以建立一个定制的图形窗口，以可见、更易于获取的方式显示数据。可以交互式地创建图形，作为网站的一部分显示给用户。无论在这个区域中做什么，编写代码以创建色彩斑斓的高效图形都是一件非常赏心悦目、能给人以满足感的任务。

在其他地方，在自定义控件中，有一种开发应用程序的最强大的方式。设计良好的自定义控件可以在工具箱中添加能多次使用的工具。有许多能够这样为应用程序提供大部分功能的控件，建立应用程序可以简化把每一种控件添加到应用程序中，并设置其属性的操作。

因此，把这两个应用领域合在一起，就有了相当强大的工具。在自定义控件中使用图形代码时，有一种非常好的方式可以极大地提高程序员的效率，同时增加应用程序的可用性和可视性。建立许多自定义控件，可以影响应用程序的整个外观和使用性。我们可以添加自定义控件，灵活地适应标准的 Windows 控件，或者用自己控件的外观和操作方式来替代所有 Windows 控件的外观和操作方式，创建出类似于游戏的环境，取悦于完全不同的用户群。

本书将讨论这种组合，特别是讨论在 .NET Framework 中使用 C# 语言和图形创建自定义控件。实际上，我们将深入研究 GDI+，即 .NET Framework 基类库中用于图形编程的那一部分。首先介绍 GDI+ 的基础知识，接着讲述它们在创建有用控件中的应用。

第1章介绍基础知识，为以后的讨论打下基础，主要内容如下：

- GDI+ 是什么——它与 .NET Framework 的关系，它的作用，它如何与以前的图形库相关联，并简要介绍它包含的命名空间和类。
- 自定义控件是什么——简要论述 Windows Forms 和 Web Forms 自定义控件，以及 .NET 如何创建它们。
- 第一个使用 GDI+ 编程的例子(使用 Visual Studio .NET IDE 开发)

下面首先介绍 GDI+ 和 .NET Framework。

1.1 GDI+ 和 .NET

本书的大多数读者都很熟悉采用 .NET 的各种原因，也很熟悉 C#。但首先我们仅重点介绍 .NET 平台与本书内容有密切关系的两个方面：

- 通过 Windows Forms 进行功能丰富的客户应用程序的开发：Windows Forms 是建立功能丰富的客户应用程序的一种设计优秀的类层次结构。高度交互式的环境、属性和事件，以及真正面向对象的方式，提供了异常强大的工具集。它们与 .NET Framework 中的分布式计算功能一起使用，就可以很容易地建立起 n 层应用程序。建立自定义控件的便利性也是作者见过的最好的一种。