

商业开发 代码库系列

Visual Basic.NET

案例开发集锦

康祥顺 蔡 宏 张小丽 编著
杨继锋 审校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

商业开发代码库系列

Visual Basic.NET案例开发集锦

康祥顺 蔡 宏 张小丽 编著

杨继锋 审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要通过具体的实例介绍如何运用Visual Basic.NET开发工具开发实际的应用程序,内容从基本应用到高级处理都有所涉及,包括界面设计、图像处理、多媒体应用、系统文件处理、硬件控制、基本数据处理、系统控制、网络应用、数据库应用编程、综合案例等方面。每个案例的编排都是严格按照读者的阅读习惯进行组织的,由具有丰富经验的项目开发人员亲自编写,大部分案例都已经在项目开发中经过了实践的检验,因此,对读者提高自己的应用程序开发能力有极大的帮助。

本书适用于大中专院校学生、程序设计人员和Visual Studio爱好者。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual Basic.NET案例开发集锦/康祥顺等编著. —北京:电子工业出版社, 2006.1

(商业开发代码库系列)

ISBN 7-121-02064-5

I. V… II. 康… III. BASIC语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2005)第143044号

责任编辑:徐云鹏

特约编辑:卢国俊

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:33.625 字数:860千字

印 次:2006年1月第1次印刷

定 价:50.00元(含光碟1张)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换,若书店售缺,请与本社发行部联系。联系电话:010-68279077。质量投诉请发邮件至zlts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

前 言

你需要阅读本书吗

微软推出的第三代网络平台.NET, 激发了新的程序开发方式。在新的架构中, .NET强调语言归语言, 系统归系统, 在平台上推出新的.NET Framework, 通过一致的Common Language Runtime执行应用程序。

现今对于.NET平台的看法各有不同。有些人说.NET是微软的下一代Visual Studio开发环境; 有些人说它是一个新的程序语言(C#); 还有些人说它是以XML和SOAP为基础的资料交换与传递信息的机制。其实, 上述三者都是.NET所扮演的角色, 而且还不止于此。

Visual Basic曾经是微软公司推出的优秀的前端开发工具, 而且现在也有很多人利用Visual Basic来开发项目。自Visual Basic问世以来, 其友好的集成开发界面、可视化的双向开发模式、良好的数据库应用支持以及高效的程序开发就备受广大程序设计师的好评。随着计算技术的发展, 对于系统的跨平台集成和相互通信的需求越来越大, 好在.NET能够解决这一问题, 对于支持.NET开发的语言之一Visual Basic.NET, 它能够使当今拥有庞大用户群的Visual Basic开发人员成功地跨入.NET的开发领域。随着.NET技术的进步和发展, 它必将成为今后最佳的编程语言之一。

学习一门编程语言, 最为关键的不是去死记函数或者方法, 而是掌握如何应用这些函数、方法去解决实际的问题。在学习的过程中, 我们很可能都有过这样的经历: 对于这门编程语言的函数、方法也知道不少, 利用函数、方法也能够编写几个简单的小程序来, 但是, 当再翻看别人的书或者使用别人的小软件时, 才明白原来这个函数还能这样用。这就是你自己钻研所不能取得的成果。本书汇集了Visual Basic.NET编程语言所能解决的问题的许多经典实例, 无论是界面设计、还是网络处理, 无论是简单的函数应用, 还是高级的底层API函数应用都包括在本书之中。阅读本书后, 你能够在最短的时间内迅速提升自己的编程水平, 避免闭门造车所造成的低水平徘徊。因此, 如果你已经学习了Visual Basic, 有了一定的Visual Basic.NET语法基础, 那么本书将会使你的编程水平发生质的飞跃!

本书涵盖的内容以及编排格式

本书共10章, 每一章都是由既相互独立, 又相互间有一定逻辑关系的实例组成。每一个实例都分为5个部分编写, 包括:

→ 案例运行效果与操作: 让读者对本案例有一个直观的印象, 以便于读者进一步了解





如何解决这个问题;

→ 制作要点: 让读者明白, 要解决这个问题, 应该采取什么方法, 使用哪些函数、组件和技巧;

→ 步骤详解: 告诉读者如何一步一步地实现这个案例, 以便于读者自己动手实现案例效果, 达到加深理解、锻炼自我的目的;

→ 代码添加与解释: 告诉读者如何实现此案例的功能, 让读者理解代码的运用以及解决问题的思路; 同时在添加的代码中, 还会告诉读者代码的功能和解决的问题;

→ 程序源代码与解释: 附带了所有的该案例的源代码, 读者只要将这些源代码复制到自己的程序中就能够实现程序的功能, 同时对于程序中应用到的难以理解的函数或者对象都添加了详细的解释, 以便于读者弄清楚问题的根源。

这些案例的源代码, 既可以用来试验某个编程的概念, 也可以剪贴到自己的程序中, 实现某种特定的功能, 达到迅速解决问题的效果。通过阅读本书, 你将学习到以下几个方面的知识:

→ 学会如何设计专业级的用户界面;

→ 学会如何在程序中处理多媒体编程的问题;

→ 学会如何在程序中处理系统文件;

→ 学会如何在程序中控制系统硬件和外围设备;

→ 学会如何在程序中调用Windows的底层API函数;

→ 学会如何在系统中处理数据库;

→ 学会如何应用网络组件、静态类处理网络操作;

→ 学会如何在程序中处理复杂的跨平台的应用问题;

→ 学会如何应用提供的组件、函数、对象开发功能复杂的应用系统, 包括B/S结构的应用系统、C/S结构的应用系统, 让读者了解开发一个应用系统的过程。

本书配套光碟的使用

为了方便读者阅读、测试和使用本书提供的案例, 本书附带了每个案例的源代码。源代码严格按照章节编排, 例如第1章第一个案例的源代码存放位置为: 源代码\1\案例1, 每个案例目录下附带了所有用到的资源文件, 如图片、数据表等。每个案例都是作者在Windows XP+Visual Studio.NET 2003下进行严格测试后再组织起来的, 尽管放心使用。但是可能存在这样的问题: 案例工程文件打开后无法找到引用的文件、资源文件, 这主要有两个方面的原因, 一个是作者保存案例文件的位置与你的不同, 另一个就是作者的机器名与你的设置不一致。如果出现这两个方面的问题, 你可以遵照实现步骤自己制作案例程序, 然后把光碟中的源代码复制到相应的程序部分, 或者打开程序中的对象检查器, 重新根据自己的机器设置



相关属性。

但是，鱼和熊掌不能兼得。虽然方便了读者，但也可能使读者对代码的熟悉程度、技术要点和编程思路等问题的领会程度降低。因此，强烈建议读者先看书案例的运行效果，然后再看源代码，如果心中已经有数了，可以自己动手举一反三，设计类似的程序来检验自己的想法，或者对程序进行修改和改进。

本书每一章内容都是由具有丰富经验的从事实际项目开发的人员编写的，大部分的案例都已经在项目开发中得以运用，其操作性、实践性和指导性都很强，因此这是一本难得的专业级实例图书。感谢与我们共同完成此书的合作者，他们是：张庆、崔竞、张强、董军、欧立奇、保春艳、李欣、梅光耀、黄显堂、康祥琴、赵辉。这些同志虽然从事着繁重的项目开发工作，但是仍然在工作之余，兢兢业业、严谨认真地编写了这本图书，保证了本书的质量和进度。对电子工业出版社和北京美迪亚电子信息有限公司的所有人员表示感谢，感谢他们为本书的顺利出版所付出的辛勤劳动。

由于水平有限，尽管对实例做了严格的审核和测试，但书中仍难免有一些错误，敬请广大读者不吝赐教，编者在此表示感谢！



目 录

第1章 界面设计	1
案例1 程序执行进度指示器	1
案例2 动态菜单	7
案例3 向ListBox列表项添加自定义图片	11
案例4 在程序状态栏做图	17
案例5 向系统状态栏添加程序启动图标	21
案例6 创建系统工具栏	25
案例7 淡入淡出的窗体	30
案例8 具有渐变色彩背景的窗体	32
案例9 任意形状的窗体	40
案例10 渐明渐暗的窗体	46
案例11 可分割的窗体	48
案例12 多文档MDI界面设计	51
第2章 图形图像处理	56
案例1 风格各异的画笔	56
案例2 绘制贝塞尔 (Bezier) 曲线	66
案例3 具有背景花纹的各式形状	69
案例4 Windows画板	75
案例5 抓图软件的实现	77
案例6 模拟Photoshop的滤镜效果	82
案例7 通过剪贴板Clipboard操作图像	89
案例8 渐变色图形	92
案例9 在窗体中绘制图形文字	100
案例10 图像的百叶窗效果	104
案例11 图像的淡入淡出效果	108
案例12 图像漫游技术	114
案例13 动画处理——Java Applet小向导	117
案例14 动画处理——跳跃的小球	120
案例15 电子石英钟	124
第3章 多媒体应用	129
案例1 屏幕保护程序	129
案例2 可爱的小人动画	136





案例3	MP3媒体播放器	142
案例4	在窗体中播放Flash动画	156
案例5	高清晰多媒体播放器	160
第4章	文件处理	167
案例1	FSO文件模型的应用	167
案例2	记事本	176
案例3	图形文件的复制——大型二进制文件的读写	187
案例4	访问注册表	193
案例5	文件的加密与解密	198
案例6	创建自己的文件类型	203
案例7	文件系统监控器	207
案例8	读写事件日志	213
第5章	基本数据处理	218
案例1	福利彩票机选模拟器	218
案例2	冒泡排序法	221
案例3	线程同步——存款机	226
案例4	二维数组排序法	229
案例5	汉诺塔数学智力游戏	233
案例6	字符串查找、替换	236
案例7	中英文字数统计	241
案例8	多线程应用举例——收发邮件	247
第6章	硬件控制	252
案例1	电话拨号程序	252
案例2	控制设备驱动程序	255
案例3	检测光驱中是否已准备好光碟	263
案例4	获取系统打印机列表	267
案例5	获取CPU信息	270
案例6	预览打印文件	272
案例7	让鼠标失灵	284
第7章	系统控制	289
案例1	获取系统信息	289
案例2	拦截和筛选消息	293
案例3	获取系统环境变量信息	296
案例4	将程序添加到系统弹出菜单	299



案例5 调用外部应用程序	303
案例6 禁止同一个程序启动多个实例	305
案例7 设置输入法	307
第8章 网络应用	311
案例1 E-mail邮件发送和接收	311
案例2 TCP点对点 (P2P) 联机程序——窗口模式	325
案例3 TCP点对点 (P2P) 联机程序——控制台模式	343
案例4 UDP点对点 (P2P) 联机程序	350
案例5 域名与IP地址的相互转换	358
案例6 在不同的ASP.NET Web页面间传递数据	361
案例7 验证Web窗体的用户输入	370
案例8 在Web页面中上传文件	377
案例9 Web Service服务应用	385
第9章 数据库应用编程	395
案例1 连接数据库	395
案例2 单记录浏览数据	404
案例3 参数化查询数据表	418
案例4 编程实现对数据表的编辑	424
案例5 直接更新数据源	440
案例6 更新一对多关系表	453
案例7 动态创建临时数据表	467
案例8 三层数据库应用程序	475
第10章 综合案例	495
案例1 和平电影院影票预售系统 (两层架构C/S)	495
案例2 Internet图书馆信息管理系统 (三层架构B/S)	503





第1章

界面设计

本章内容

- ✓ 程序执行进度指示器
- ✓ 动态菜单
- ✓ 向ListBox列表框添加图片
- ✓ 在程序状态栏作图
- ✓ 向系统状态栏添加程序启动图标
- ✓ 创建系统工具栏
- ✓ 淡入淡出的动态窗体
- ✓ 具有渐变色彩背景的窗体
- ✓ 任意形状的窗体
- ✓ 渐明渐暗的窗体
- ✓ 可分割的窗体
- ✓ 多文档MDI界面设计

案例1 程序执行进度指示器



案例运行效果与操作

许多大的程序，如Office等，当程序因执行某项复杂的处理需要让用户等待很长一段时间时，最好给用户提供一个程序执行进度的指示标志，以提醒用户程序仍在执行以及还需要等待多长时间。这样，在日益强调用户界面的友好性的今天，无疑能够让你的产品更加受到用户的青睐，同时，你的用户界面也更加漂亮和人性化。本案例主要用于搜索C:\WINNT目录下的所有文件，包括子文件夹下的文件，然后将搜索的结果添加到ListBox列表框中。为了提醒用户当前应用程序处理的进度，应用Visual Basic.NET提供的ProgressBar进度条控件来显示当前文件的处理进度和整个搜索任务的处理进度。“程序执行进度指示器”界面如图1-1所示。

程序运行后，单击“Start Search Files”按钮，程序即开始搜索文件，同时在“当前文件”位置显示当前正在处理的文件，在“当前进度”位置显示当前文件处理的进度，在“总任务”位置显示总的搜索任务的处理进度。搜索完整个文件夹后，将所有文件添加到“搜索结果”位置的ListBox列表框中，搜索结果如图1-2所示。



制作要点

1. ProgressBar组件的应用；
2. ListBox组件的应用；
3. FileSystemObject对象的应用；
4. DirectoryInfo()类的应用；
5. FileInfo()类的应用；
6. Label组件的应用；
7. Button按钮组件的应用。





图1-1 “程序执行进度指示器”界面



图1-2 程序搜索结果



步骤详解

1. 在Microsoft Visual Basic.NET开发环境中，单击菜单“文件”►“新建”►“项目”，系统将弹出如图1-3所示的“新建项目”对话框。在该对话框中，在“项目类型”选择框中选定“Visual Basic项目”，然后在“模版”选择框中选定“Windows应用程序”，然后在窗体下方的“名称”文本框中输入应用程序的名称，默认为“WindowsApplication1”，但是为了便于记忆，请给应用程序取一个有意义的名字，如本案例为了标记明确，取名为“案例1”，然后还可以自定义保存项目的位置，默认为“C:\Documents and Settings\用户名\My Documents\Visual Studio Projects”目录下，如果需要更改存储位置，请单击“浏览”按钮进行更改。当一切设定好后，请单击“确定”按钮。

注意：为了简化叙述，在以后的案例中将不再具体叙述上面的步骤，请读者参照上面的步骤进行学习。

2. 在系统自动创建的空白Windows窗体Form1中（如图1-4所示）分别添加7个Label标签组件、2个ProgressBar进度条组件、1个Button按钮组件和1个ListBox组件，各组件属性如表1-1所示。

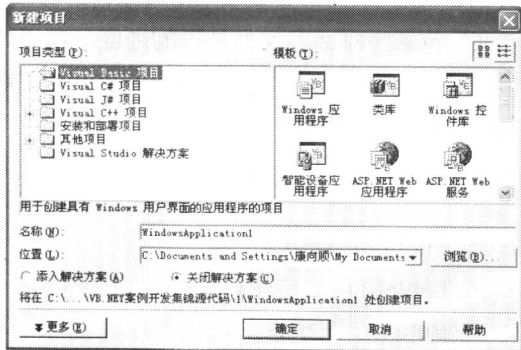


图1-3 “新建项目”对话框

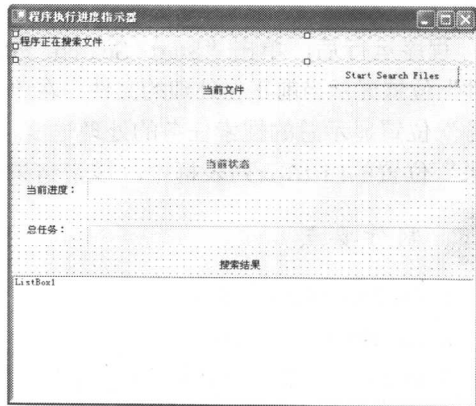


图1-4 程序设计界面



表1-1 窗口中各组件的属性设置

对象名	属性	属性值	功能
Form1	text	“程序执行进度指示器”	窗体标题
Lable1	text	“程序正在搜索文件...”	显示用户提示信息
Lable2	text	“当前文件”	提示信息
Lable3	text	空	显示当前正在处理的文件名
Lable4	text	“当前状态”	提示信息
Lable5	text	“当前进度”	提示信息
Lable6	text	“总任务”	提示信息
Lable7	text	“搜索结果”	提示信息
Button1	text	“Start Search Files”	开始搜索
ProgressBar1 ~ProgressBar2		程序中动态设定	显示程序执行的进度
ListBox1		程序中动态设定	显示搜索结果



代码添加与解释

1. 双击窗体进入代码编辑器，在程序的最前面引用后面程序中将要使用的名字空间，然后再在Public Class Form1类声明中定义两个全局变量：

```
Imports System.IO
Public Class Form1
    Inherits System.Windows.Forms.Form
    '声明全局变量
    Dim count As Integer '计算文件个数
    Dim filesize As Integer '计算文件大小
    Dim fso, fol
```

注意：全局变量一定要定义在类声明“Public Class Form1 Inherits System.Windows.Forms.Form”语句的后面，而且必须在类构造器函数“Public Sub New()”的前面，否则，后面将无法找到这些变量。

约定：以后的案例中所定义的全局变量都将在此位置，届时就不再专门强调，请读者一定记住。

2. 在代码编辑器中添加自定义函数，用于计算整个文件夹的大小：

```
'声明一个函数，返回文件夹大小
Public Function getfoldersize(ByVal dir As String) As Integer
    Dim fso, fol
    Dim sizes As Integer
    '创建filesystemobject对象和Folder对象
    fso = CreateObject("scripting.filesystemobject")
    fol = fso.GetFolder(dir)
```





```
sizes = fol.size  
Return sizes  
End Function
```

这个函数必须写在“#End Region”标记后面。但是读者可以运行一下，看看程序执行的效果，在用户单击了“Start Search Files”按钮后，程序并不是即时执行，而是稍有停顿，问题的根源就是程序需要计算文件的大小。读者可以将计算文件大小的代码写在构造函数里，构造函数的语句为：

```
InitializeComponent( )
```

在InitializeComponent()调用之后可以添加任何初始化，而这些处理动作是在窗口还没有显示之前完成的，所以看来用户根本不知道。

3. 声明一个自定义过程函数，用于递归搜索文件：

声明一个过程函数，用于循环遍历指定目录下的文件和子目录

```
Public Sub displaytree(ByVal dir As String)
```

声明DirectoryInfo类型的变量，以便创建该对象的实例

```
Dim dirinfos As DirectoryInfo
```

声明DirectoryInfo类型的变量数组

```
Dim dirinfo1 As DirectoryInfo( )
```

```
Dim subdirinfo As DirectoryInfo
```

声明FileInfo类型的数组变量

```
Dim fileinfos As FileInfo( )
```

```
Dim fileinfo As FileInfo
```

创建DirectoryInfo对象的实例，该对象要求传递一个字符串类型的目录参数

```
dirinfos = New DirectoryInfo(dir)
```

首先取得指定目录下的所有文件

```
fileinfos = dirinfos.GetFiles("*.")
```

准备更新ListBox对象

```
Me.ListBox1.BeginUpdate( )
```

遍历每一个FileInfo()数组成员

```
For Each fileinfo In fileinfos
```

动态设置ProgressBar进度组件的最大值和最小值，其中，最大值为当前文件的大小

```
Me.ProgressBar2.Maximum = fileinfo.Length
```

```
Me.ProgressBar2.Minimum = 0
```

设置ProgressBar对象的步长，该对象将按照该步长前进

```
Me.ProgressBar2.Step = 1
```

统计当前已经搜索的文件的总的大小

```
filesize = filesize + fileinfo.Length
```

统计当前已经搜索到的文件数量

```
count = count + 1
```

动态显示当前搜索的文件名称

```
Me.Label3.Text = fileinfo.FullName
```

刷新Label3组件，否则不能显示

```
Me.Label3.Refresh( )
```



```

        '驱动ProgressBar对象按照设置的步长前进
        Me.ProgressBar2.PerformStep( )
        '设置代表总进度的ProgressBar对象的当前值
        Me.ProgressBar1.Value = filesize
        '将搜索的文件加入ListBox列表对象
        Me.ListBox1.Items.Add(fileinfo.FullName)
    Next
    '当c:\winnt下的文件搜索完毕后, 继续搜索子目录
    dirinfo1 = dirinfos.GetDirectories
    '遍历子目录下的文件
    For Each subdirinfo In dirinfo1
        '递归调用该函数来搜索该子目录下的文件, 这时传递的参数为子目录
        displaytree(subdirinfo.FullName)
    Next
    '完成ListBox对象的更新
    Me.ListBox1.EndUpdate( )
    '显示搜索结果
    Me.Label1.Text = "搜索完毕, 共搜索到" & count.ToString & "个文件, 文件总的
大小为: " & Int(filesize / 1024).ToString & "KB"
End Sub

```

4. 双击按钮“Start Search Files”, 然后在Click用户单击事件里添加如下处理代码:

```

'处理按钮单击事件
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button1.Click
    '初始化组件对象, 必须这样, 否则当再次按下按钮时会出错, 而且不能放到Load事件中
    Me.count = 0
    Me.filesize = 0
    Me.ProgressBar1.Maximum = getfoldersize(System.IO.Directory.GetCurrentDirectory)
    Me.ProgressBar1.Minimum = 0
    Me.Label3.Text = ""
    Me.ListBox1.Items.Clear( )
    '调用自定义函数
    displaytree(System.IO.Directory.GetCurrentDirectory) '浏览指定的目录
End Sub

```



程序源代码与解释

```

Imports System.IO
Public Class Form1
    Inherits System.Windows.Forms.Form
    '声明全局变量
    Dim count As Integer '计算文件个数
    Dim filesize As Integer '计算文件大小
    Dim fso, fol

```





#Region " Windows 窗体设计器生成的代码 "

'省略系统自动生成的代码

#End Region

'声明一个函数，返回文件夹大小

Public Function getfoldersize(ByVal dir As String) As Integer

Dim fso, fol

Dim sizes As Integer

'创建filesystemobject对象和Folder对象

fso = CreateObject("scripting.filesystemobject")

fol = fso.GetFolder(dir)

sizes = fol.size

Return sizes

End Function

'声明一个过程函数，用于循环遍历指定目录下的文件和子目录

Public Sub displaytree(ByVal dir As String)

'声明DirectoryInfo类型的变量，以便创建该对象的实例

Dim dirinfos As DirectoryInfo

'声明DirectoryInfo类型的变量数组

Dim dirinfo1 As DirectoryInfo()

Dim subdirinfo As DirectoryInfo

'声明FileInfo类型的数组变量

Dim fileinfos As FileInfo()

Dim fileinfo As FileInfo

'创建DirectoryInfo对象的实例，该对象要求传递一个字符串类型的目录参数

dirinfos = New DirectoryInfo(dir)

'首先取得指定目录下的所有文件

fileinfos = dirinfos.GetFiles("*. *")

'准备更新ListBox对象

Me.ListBox1.BeginUpdate()

'遍历每一个FileInfo()数组成员

For Each fileinfo In fileinfos

'动态设置ProgressBar进度组件的最大值和最小值，其中，最大值为当前文件的

大小

Me.ProgressBar2.Maximum = fileinfo.Length

Me.ProgressBar2.Minimum = 0

'设置ProgressBar对象的步长，该对象将按照该步长前进

Me.ProgressBar2.Step = 1

'统计当前已经搜索的文件的总的大小

filesize = filesize + fileinfo.Length

'统计当前已经搜索到的文件数量

count = count + 1

'动态显示当前搜索的文件名称

Me.Label3.Text = fileinfo.FullName

'刷新Label3组件，否则不能显示

Me.Label3.Refresh()



```

        '驱动ProgressBar对象按照设置的步长前进
        Me.ProgressBar2.PerformStep( )
        '设置代表总进度的ProgressBar对象的当前值为当前已经搜索到的文件的总大小
        Me.ProgressBar1.Value = filesize
        '将搜索的文件加入ListBox列表对象
        Me.ListBox1.Items.Add(fileinfo.FullName)
    Next
    '当c:\winnt下的文件搜索完毕后, 继续搜索子目录
    dirinfo1 = dirinfos.GetDirectories
    '遍历子目录下的文件
    For Each subdirinfo In dirinfo1
        '递归调用该函数来搜索该子目录下的文件, 这时传递的参数为子目录
        displaytree(subdirinfo.FullName)
    Next
    '完成ListBox对象的更新
    Me.ListBox1.EndUpdate( )
    '显示搜索结果
    Me.Label1.Text = "搜索完毕, 共搜索到" & count.ToString & "个文件, 文件总的
大小为: " & Int(filesize / 1024).ToString & "KB"
End Sub

'处理按钮单击事件
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    '初始化组件对象, 必须这样, 否则当再次按下按钮时会出错, 而且不能放到Load事
件中
    Me.count = 0
    Me.filesize = 0
    Me.ProgressBar1.Maximum = getfoldersize(System.IO.Directory.GetCurrentDirectory)
    Me.ProgressBar1.Minimum = 0
    Me.Label3.Text = ""
    Me.ListBox1.Items.Clear( )
    '调用自定义函数
    displaytree(System.IO.Directory.GetCurrentDirectory) '浏览指定的目录
End Sub

```

案例2 动态菜单



案例运行效果与操作

许多大的程序, 如Visual Studio .NET等, 程序提供的菜单能够根据当前的项目类型自动产生; 而有些安全性要求比较高的程序, 如银行结算系统、图书管理系统等, 会根据登录用户的类型动态产生相应权限的菜单, 非权限内的操作菜单要么被隐藏, 要么处于无效状态。本案例主要演示了一个图书管理系统的窗口菜单, 它能够根据登录用户的类型产生相应的操





作菜单，而且在用户类型菜单项中选定相应的用户类型。程序执行后，首先显示一个“用户登录”对话框（如图1-5所示），要求输入用户的工号。

输入用户工号后，单击“确定”按钮，程序将弹出主窗口，产生属于该工号类型的系统操作菜单，如图1-6所示。

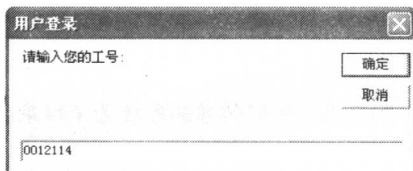


图1-5 “用户登录”对话框

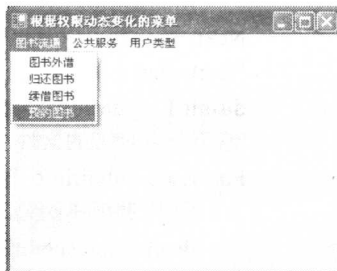


图1-6 图书流通人员的操作菜单

登录人员为图书采编类型的操作界面如图1-7所示。

注意，这时已经没有“图书流通”菜单项了。当登录人员为“参考咨询”类型时，用户操作菜单如图1-8所示。

注意，这时菜单项由“公共服务”变为了“参考咨询”。系统中默认类型如下：

- 工号左边两位号码为00——图书流通人员
- 工号左边两位号码为01——图书采编人员
- 工号左边两位号码为02——参考咨询人员

如果输入的工号不符合上述要求，那么系统将不提供任何菜单，而且在“用户类型”菜单项中选中“无权用户”，如图1-9所示。

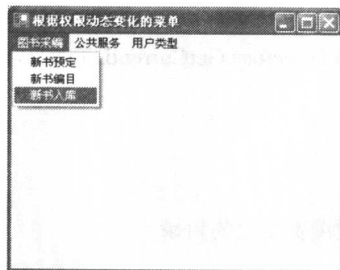


图1-7 图书采编人员的操作菜单

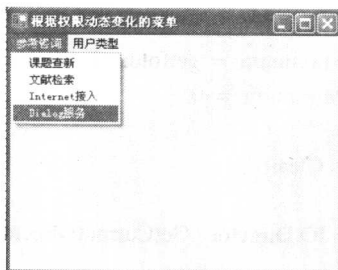


图1-8 参考咨询人员的操作菜单

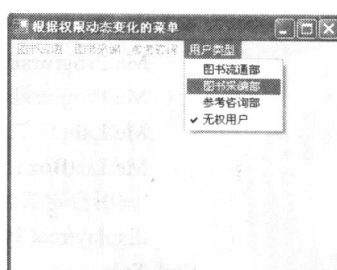


图1-9 无权用户菜单



制作要点

1. MainMenu菜单组件的应用；
2. InputBox输入对话框静态类的应用。

