您的潜力，我们的动力

Microsoft
微软（中国）有限公司

Visual Basic 2005开发技巧系列课程

在Visual Basic 2005中使用.NET Framework 2.0新增功能

讲师：施凡

msdn

MSDN Webcasts

摘要

- 基础类库增强
- 数据访问增强
- Windows Forms增强
- 总结

基础类库增强

- 控制台
- 驱动器信息
- 文件访问控制权限表
- 压缩流
- FTP
- 高精度计时器

控制台

- Console类主要增强：
- 增加颜色设置功能
- 可配置控制台标题和窗口大小
- 可复制显示缓冲区内的字符
- 可控制光标的大小、隐藏与位置
- 控制蜂鸣器发出声音

Console类的新增静态成员

- 新增属性设置颜色和控制台窗口属性

```
Console.BackgroundColor = ConsoleColor.Green
Console.WindowHeight = 40
Console.CursorSize = 50
```

- 新增方法用于复制缓冲区及控制蜂鸣器

```
‘ 将矩形(0, 0, 20, 10)中的字符移动到位置(5, 5)
Console.MoveBufferArea(0, 0, 20, 10, 5, 5)
‘ 发出600Hz的声音，持续1000毫秒
Console.Beep(600, 1000)
```

驱动器信息

- 在以前的Framework中，没有关于驱动器信息的类。
- 需要调用平台API才能获取驱动器类型、盘符、文件系统和剩余空间等信息
- 现在可通过新增System.IO.DriveInfo类来获取驱动器信息

驱动器信息

- 使用DriveInfo的属性

属性	描述
Name	驱动器名称A-Z
TotalFreeSpace	全部剩余空间
AvailableFreeSpace	可用剩余空间（配额限制）
TotalSpace	全部空间总额
DriveType	驱动器类型
DriveFormat	文件系统等

访问控制权限表（ACL）

- 您一定使用过Windows的文件/文件夹访问权限设定功能
- 现在可以用托管代码获取或修改访问控制权限
- 仅限NTFS文件系统

访问控制权限表（ACL）

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- 用File或Directory等类获取或设置访问控制权限表
- 用System.Security.AccessControl中的相关类型进行访问控制的设置

```
‘ 获取设置
Dim fSecurity As FileSecurity = File.GetAccessControl(FileName)
‘ 增加一条访问控制规则
fSecurity.AddAccessRule(New FileSystemAccessRule(Account, Rights,
ControlType))
‘ 将访问控制添加回文件
File.SetAccessControl(FileName, fSecurity)
```

压缩流

- 在进行带宽/存储空间有限的任务时，压缩数据能起到很大作用
- 新增System.IO.Compression命名空间，提供数据压缩功能
- 内置Deflate压缩算法（LZ77和哈夫曼编码）
- 内置GZipStream类以提供gzip数据格式，可扩展压缩算法

压缩流

- 压缩/解压流是管道流，必须用另一个流作为输出（解压亦然）

```
‘ 用一个文件流作为压缩流的输出
destinationStream = New FileStream(destinationFile,
FileMode.OpenOrCreate, FileAccess.Write)

‘ 创建压缩流，并指向目标文件流
compressedStream = New GZipStream(destinationStream,
CompressionMode.Compress, True)

‘ 向压缩流写入数据，目标流就会得到压缩的数据
compressedStream.Write(buffer, 0, buffer.Length)
```

FTP

- 根据用户需求，新增FtpWebRequest和FtpWebResponse类
- 可从WebRequest类直接创建和使用

```
Dim ftpRequest As FtpWebRequest =
CType(WebRequest.Create(downloadUri), FtpWebRequest)

ftpRequest.Credentials = New NetworkCredential(Me.m_userName,
Me.m_password)
ftpRequest.Method = WebRequestMethods.Ftp.DownloadFile

Dim ftpResponse As FtpWebResponse =
CType(ftpRequest.GetResponse, FtpWebResponse)

Dim stream As Stream = ftpResponse.GetResponseStream
```

StopWatch 高精度计时

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- DateTime.Now的精度有时不能满足需求
- 新增System.Diagnostics.StopWatch
- 提供比DateTime.Now更精确的计时功能
- 用于性能测量和优化
- 并非所有系统都支持高精度计时，在不支持高精度计时的系统上，自动采用和DateTime.Now相同的计时原理。

StopWatch 高精度计时

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

```
Dim sw As New System.Diagnostics.StopWatch

sw.Start() ‘用Start开始计时
‘这里开始要测量的代码
For i As Integer = 1 To 10000
        SomeFunction()
Next

sw.Stop() ‘停止计时

‘获得以毫秒为单位的时间间隔
Dim elapsed_ms As Long = sw.ElapsedMilliseconds
‘获得TimeSpan类型的时间间隔
Dim elapsed As TimeSpan = sw.Elapsed
```

数据访问增强

- 性能提升
- 异步查询
- 批量更新
- SQL Server大块数据复制
- 与SQL Server 2005紧密集成

数据访问相关的类库性能提升

- XML相关类库性能有很大提升
- DataTable、DataSet性能提升
- DataGridView数据显示控件比DataGrid性能有所提升

异步查询

- 两种方法实现异步查询
  - Polling
  - Callback
- 利用Command对象新增方法 BeginExecuteXXX, EndExecuteXXX

异步查询 Polling

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- 返回IAsyncResult
- 通过反复查询IsCompleted属性判断结束

```
Dim myCommand As SqlCommand = New SqlCommand(myQuery,
myConnection)
myConnection.Open()
‘开始异步查询
Dim myResult As IAsyncResult = myCommand.BeginExecuteReader()
‘不断检查IsCompleted
While Not myResult.IsCompleted
        Application.DoEvents() ‘浪费时间或者显示进度条
End While
‘获取查询结果
Dim myReader As SqlDataReader =
myCommand.EndExecuteReader(myResult)
```

异步查询 Callback

- 用回调函数接受查询结束的信号
- 在回调函数内取得查询结果

```
Dim myCallBack As AsyncCallback = AddressOf HandleCallback
‘将回调函数传递给BeginExecuteReader方法
‘第二个参数将Command自身传递给回调函数
myCommand.BeginExecuteReader(myCallBack, myCommand)
……
‘在回调函数HandleCallback中
‘可从参数中获取我们传入的原始Command对象
Dim myCommand As SqlCommand = myResult.AsyncState
‘获取查询结果
Dim myReader As SqlDataReader =
myCommand.EndExecuteReader(myResult)
```

批量更新

- 以前版本的DataAdapter每次只能更新单条记录
- 如果要更新的数量很大，单条更新的性能可能较差
- 现在只需设置一个属性，就能进行批量更新

```
myDataAdapter.UpdateBatchSize = 50
```

批量更新使用提示

- 将UpdateBatchSize设置为1，就表示禁止批量更新（回到.NET 1.1）
- 将UpdateBatchSize设置为0，就表示按数据库支持的一次更新最大值进行批量更新
- 并非越大越好，具体性能应根据自己的环境进行测量

SQL Server大块数据复制

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- SQL Server的bcp或DTS可以高性能地在数据库内部或者数据库间复制大量数据
- 新增System.Data.SqlClient.SqlBulkCopy类，可以让您用VB程序进行类似操作
- 性能比使用DataAdapter逐条复制高很多倍

SQL Server大块数据复制

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- 可选择从DataTable获取数据进行复制，也可以从IDataReader直接读取数据
- 目标只能是SQL Server，数据源不受限制

```
Dim myBulkCopy As SqlClient.SqlBulkCopy = New
SqlClient.SqlBulkCopy(myConnection)
‘设置提醒事件和发生条件
AddHandler myBulkCopy.SqlRowsCopied, AddressOf onSqlRowsCopied
myBulkCopy.NotifyAfter = 10
‘必须指定表名
myBulkCopy.DestinationTableName = "Data2"
‘开始写数据，本例从DataTable中获取数据
myBulkCopy.WriteToServer(myDataSet.Tables("MyTable"))
myBulkCopy.Close()
```

与SQL Server 2005紧密集成

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- SQL Server 2005对CLR有深层次支持
- SQL Server中可以使用CLR的所有类型
- 可以用VB或C#编写存储过程
- 具体内容可参考相关课程

Windows Forms 增强

- BackgroundWorker
- 新窗口布局控件
- Strip系列控件

您的潜力，我们的动力
Microsoft
微软（中国）有限公司
msdn
MSDN Webcasts

BackgroundWorker

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- 执行后台任务专用组件
- 可安全更新窗口界面元素
- 无需手动开发多线程和处理异步调用问题

DEMO

- BackgroundWorker实现安全异步调用

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

窗体布局控件

- 更容易适应用户不同的分辨率设置
- 更容易控制改变窗口大小后的布局
- 有FlowLayout和TableLayout两种布局

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

Strip系列控件

- 很容易给菜单加入小图标
- 容易制作Office 2003风格的工具栏
- 工具栏可停靠在窗体周围
- 效果比Toolbar或者Menu更专业

DEMO

- 窗体布局控件和Strip系列控件

总结

- .NET Framework 2.0在许多方面都有改进
- 为VB开发者提供更大方便

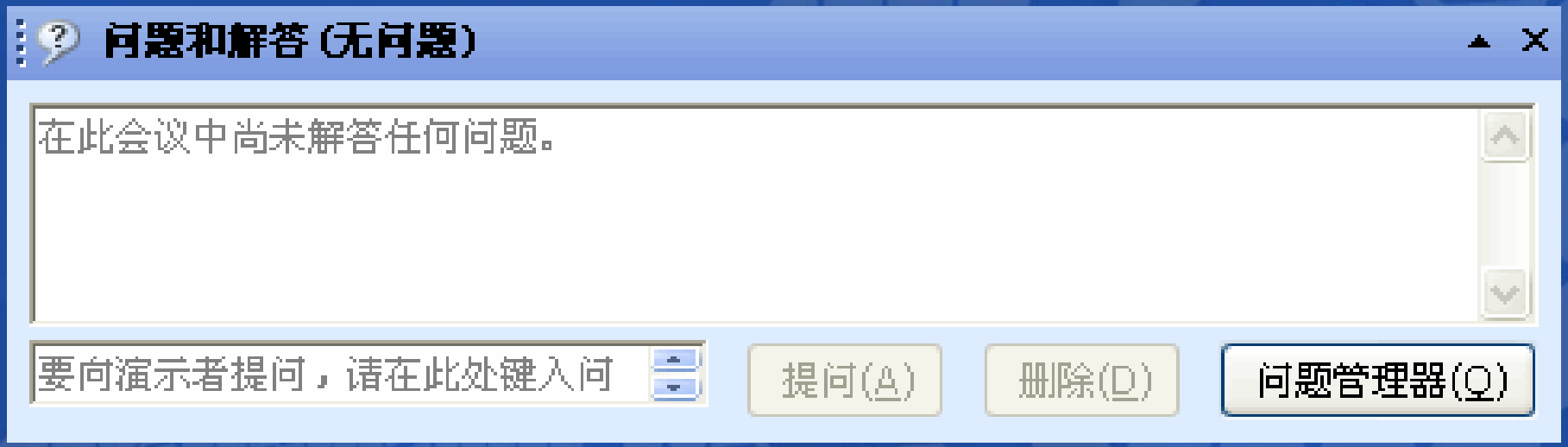
资源

- 101 Sample ——自学.NET 2.0最佳教材
- http://msdn.microsoft.com/vbasic/downloads/2005/code/101samples/

您的潜力，我们的动力
Microsoft
微软(中国)有限公司

Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。




您的潜力，我们的动力
Microsoft
微软（中国）有限公司
Microsoft
msdn
MSDN Webcasts