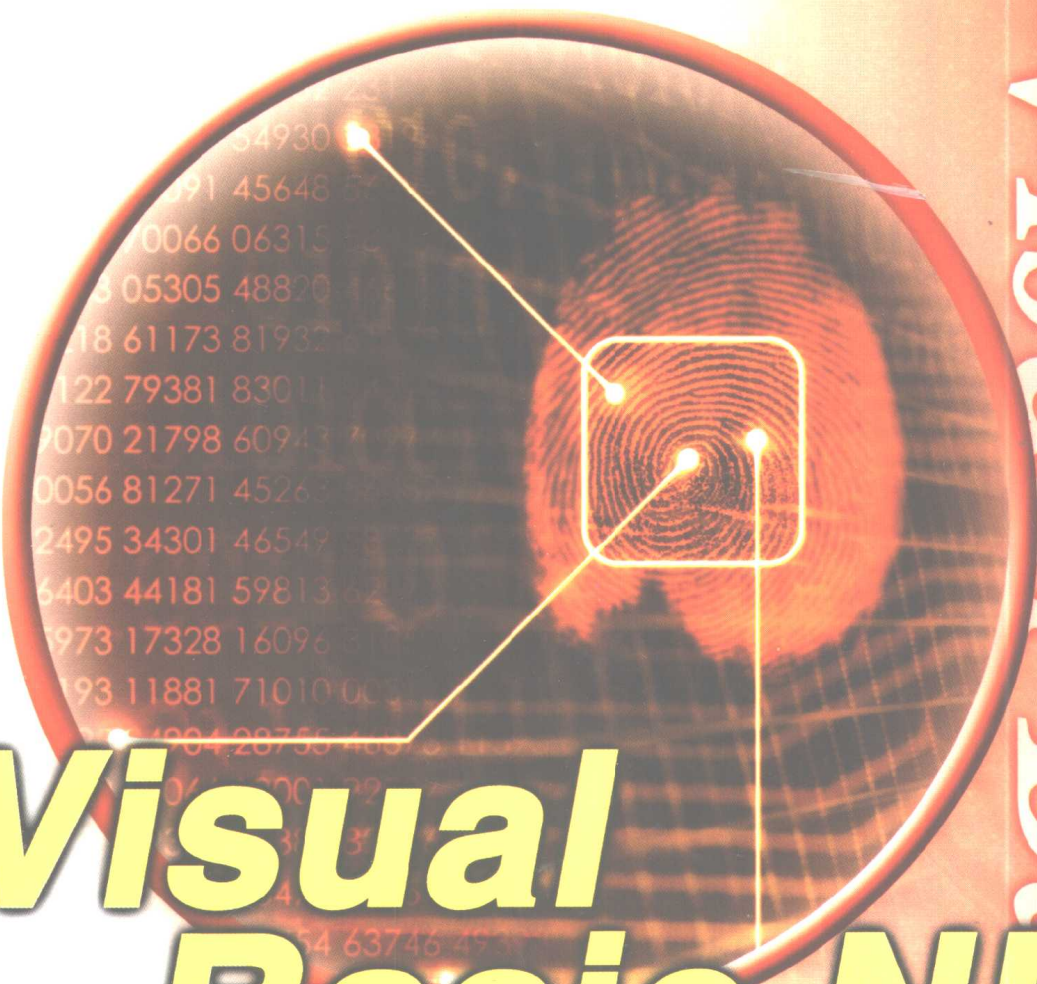


开发技术整合应用丛书



Visual Basic.NET

网络数据库 整合应用

● 精锐创作组 编著

人民邮电出版社
www.pptph.com.cn

开发技术整合应用丛书

Visual Basic.NET 网络数据库 整合应用

精锐创作组 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Visual Basic.NET 网络数据库整合应用/精锐创作组编著. —北京: 人民邮电出版社, 2001.11

(开发技术整合应用丛书)

ISBN 7-115-09802-6

I.V... II.精... III.BASIC 语言—数据库系统—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2001) 第 073962 号

内 容 提 要

本书从 Visual Basic .NET 的数据库编程基础开始, 以示例的形式由浅入深地介绍了在 Visual Basic .NET 中开发网络数据库应用程序的技术, 包括 ADO 技术、DAO 技术、SQL 语言、Web Application 等。

本书通俗易懂, 讲解细致, 对大量示例进行了透彻的分析, 适合中级程序开发人员学习使用。对于从事 Visual Basic .NET 网络数据库开发与应用的广大科研人员和高校相关专业的师生, 也不失为一本有价值的自学读物或教学参考书。

开发技术整合应用丛书

Visual Basic.NET 网络数据库整合应用

- ◆ 编 著 精锐创作组
责任编辑 张瑞喜
执行编辑 牛 磊
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线: 010-67129212 010-67129211 (传真)
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 23.5
字数: 576 千字 2001 年 11 月第 1 版
印数: 1—5 000 册 2001 年 11 月北京第 1 次印刷

ISBN 7-115-09802-6/TP · 2551

定价: 34.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

Visual Basic .NET 是运行于 Windows 平台上的交互式的可视化集成开发环境，它是美国 Microsoft 公司开发的 Microsoft Visual Studio .net 套件的一部分。像其他的可视化集成开发环境（如 Visual C++、Delphi、C++ Builder）一样，Visual Basic .NET（为了叙述方便，以下简称 VB）集程序的代码编辑、编译、连接和调试等功能于一体，给编程人员提供了一个完整方便的开发界面和许多有效的辅助开发工具。VB 的应用程序向导可以为很大一部分类型的程序提供框架代码，用户不用书写程序代码，只需点击几个按钮就可以生成一些完整的可以运行的程序。

本书通过丰富的示例程序向读者介绍如何使用 VB 开发网络数据库应用程序，这些示例程序都是作者根据自己使用和开发 VB 程序时的体会精心编写的，并针对程序员在开发过程中最迫切需要、使用频率最高的内容特意定制，可以说比较贴切地符合了初级和中级程序员的需求。另外，本书中所有示例程序的代码都经过了严格的调试和测试，读者只要按照书中的步骤往下做，最终一定能够圆满地完成程序。

本书内容结构安排如下：

第 1 章全面地介绍了 ADO 技术以及 ADO 技术在 VB 中的使用方法。本章首先对 ODBC API，以及从 DAO、RDO 到 ADO 的发展过程进行了简单的介绍，然后全面地介绍了 ADO 对象和集合，包括 Connection 对象、Command 对象和 Recordset 对象以及各对象重要的属性和方法，Errors 集合、Parameters 集合和 Fields 集合。最后还介绍了数据在网页上的显示、自定义查询等。

第 2 章详细地介绍了 DAO（数据访问对象）技术，包括 DAO 的对象、方法，并且每一种方法都有示例演示其功能，从而使读者对各种方法有一个直观的印象。在本章的最后，以示例程序充分地利用 DAO 技术的各个方面的，实现浏览任意一个 Access 数据库中任意一个表的所有记录，并且可以对表添加新的 Field（字段）和 Index（索引）。

第 3 章介绍了 SQL 语言，包括通用 SQL 查询语句、更新查询语句、表之间的连接和数据定义语言，以及 Transact-SQL 语句。

第 4 章介绍了在查询中可以使用的运算符和函数，从而可以实现比较复杂的查询功能。

Windows 提供了许多 API 函数来支持 Internet 的编程，但是这需要较多的网络知识，因此，本书在第 5 章着重介绍了 Internet 编程的方法，并举例说明了如何建立自己的网络应用程序。

为了方便读者学习，我们在本书中设计了两个小图标，它们的含义分别是：



表示注意和提示。



表示技巧。

参与本书编写的人员有尹立民、赵万军、齐明睿、许静、邓小南、胡鞍钢、徐伟宣、于景元、牛文元、方新、叶小梁、叶文虎、孙枢、刘钝、刘毅、汪寿阳、李扬、金铎、林毅夫、曹效业、薛澜、张丽新、向正贵、张玉彬、胡佩锋等。

由于编写时间紧，作者水平有限，书中难免存在一些不足之处，恳请读者批评指正。

E-mail: pptph@263.net

编 者

目 录

第 1 章 ADO 技术在 Visual Basic.NET 中的应用	1
1.1 ADO 技术在 Visual Basic.NET 中的应用	2
1.2 ADO 的 Connection 对象.....	3
1.2.1 Connection 对象的属性	3
1.2.2 Connection 对象的方法	10
1.3 ADO 的 Command 对象	18
1.3.1 Command 对象的属性	19
1.3.2 Command 对象的方法	21
1.4 ADO 的 Recordset 对象.....	22
1.4.1 Recordset 对象简介	23
1.4.2 Recordset 对象的属性.....	24
1.4.3 Recordset 对象的方法.....	31
1.5 ADO 集合	38
1.5.1 Errors 集合	38
1.5.2 Parameters 集合	39
1.5.3 Fields 集合	40
1.5.4 Properties 集合	41
1.6 列表显示数据.....	41
1.7 在网页中显示数据.....	45
1.8 数据显示控制.....	58
1.9 自定义查询.....	77
1.10 小结	101
 第 2 章 DAO 技术.....	103
2.1 DAO 对象	104
2.1.1 DBEngine 对象.....	105
2.1.2 WorksPace 对象.....	106
2.1.3 Database 对象	106
2.1.4 TableDef 对象	107
2.1.5 QueryDef 对象.....	108
2.1.6 Recordset 对象.....	109
2.1.7 Field 对象.....	111
2.1.8 Index 对象.....	112
2.1.9 Parameter 对象	113



2.1.10	User 对象	113
2.1.11	Group 对象	114
2.1.12	Relation 对象	114
2.1.13	Property 对象	115
2.1.14	Containers 集合	115
2.1.15	Document 对象	116
2.2	DAO 方法	116
2.2.1	AddNew 方法	116
2.2.2	Appendchunk 方法	117
2.2.3	BeginTrans、CommitTrans 以及 Rollback 方法	117
2.2.4	Clone 方法	118
2.2.5	Close 方法	119
2.2.6	CompactDatabase 方法	119
2.2.7	CreateDatabase 方法	120
2.2.8	CreateField 方法	121
2.2.9	CreateGroup 方法	122
2.2.10	CreateIndex 方法	122
2.2.11	CreateProperty 方法	122
2.2.12	CreateQueryDef 方法	123
2.2.13	DAO 应用示例	132
2.3	小结	170
第 3 章	简单查询的应用	171
3.1	通用查询语句	172
3.1.1	SELECT 语句	172
3.1.2	条件从句	181
3.1.3	分组从句	192
3.1.4	排序从句	195
3.1.5	分组后的条件从句	198
3.1.6	相关的函数	198
3.1.7	创建表查询	201
3.1.8	追加查询	201
3.1.9	更新查询	203
3.1.10	删除语句	203
3.2	Transact-SQL 语句	204
3.2.1	语法规则	205
3.2.2	数据类型	206



3.2.3	BACKUP.....	211
3.2.4	BEGIN..END	219
3.2.5	BREAK	220
3.2.6	BULK INSERT	220
3.2.7	CREATE DATABASE.....	224
3.2.8	CREATE INDEX	233
3.2.9	CREATE TABLE.....	246
3.2.10	IF..ELSE.....	252
3.3	小结	254
第 4 章	高级查询的应用.....	255
4.1	运算符	256
4.1.1	常用运算符.....	256
4.1.2	其他运算符.....	273
4.2	函数.....	276
4.2.1	函数确定性.....	276
4.2.2	函数排序规则.....	277
4.2.3	行集函数.....	277
4.2.4	聚合函数.....	292
4.2.5	标量函数.....	309
4.3	小结	336
第 5 章	开发 Web Application	337
5.1	服务器.....	338
5.2	接收数据	338
5.2.1	从文本框中取得表单信息.....	339
5.2.2	从复选按钮中取得表单信息.....	343
5.2.3	使用超级链接传递参数.....	349
5.2.4	使用表单的 Action 参数传递参数.....	351
5.2.5	数据集合	352
5.3	输出数据	355
5.3.1	Write 方法	356
5.3.2	Redirect 方法	356
5.3.3	缓冲输出和缓存.....	357
5.4	收集信息.....	359
5.4.1	创建 Cookie	360



5.4.2	读取 Cookie	360
5.4.3	设置 Cookie 期限	360
5.4.4	清除 Cookie	360
5.5	用户管理	360
5.5.1	站点数据共享	361
5.5.2	Application 对象	361
5.5.3	Session 对象	362
5.5.4	Global.asax 文件	363
5.6	小结	367

A decorative vertical line on the left side of the page, composed of two parallel columns of black footprints. The left column has 12 footprints and the right column has 11 footprints, all pointing downwards.

第 1 章

ADO 技术在 Visual Basic.NET 中的应用



1.1

ADO 技术在 Visual Basic.NET 中的应用

如果想采用 Visual Basic.NET 开发数据库应用程序,那么通过 ADO (ActiveX Data Objects) 技术可以很容易地存取和控制数据库中的数据,下面就来介绍 Visual Basic.NET 中的 ADO 技术。

ADO (ActiveX Data Objects) 是微软公司处理关系数据库和非关系数据库中信息的最新技术 (关系数据库管理系统用表来操纵信息,但并非所有的数据源都遵从这一模式)。ADO 没有完全取代现有的 DAO (Data Access Object),但是它确实把 DAO 的编程扩展到了新的领域。

ADO 基于微软公司最新的 OLE DB 数据访问模式,是专门为了给大范围的商业数据源提供访问而设计的,包括传统的关系数据库表、电子邮件系统、图形格式、Internet 资源等。ADO 比 DAO 所需的内存更少,所以它更适合于大流量和大事务量的网络计算机系统。

ADO 方式是 DAO 和 RDO 方式的继承者,它也有一个类似的对象模式。在 ADO 方式中,可编程对象展示了你的计算机上所有可获取的本地和远程数据源。在 Visual Basic 6.0 专业版中,使用新的 ADO 控件、把数据对象绑定到内置控件和 ActiveX 控件、创建 DHTML 应用程序以及使用 DED 等方法,都可以访问这些可编程数据对象。

ADO 对象模型如图 1-1 所示。

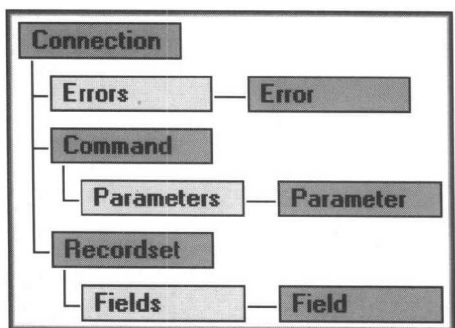


图 1-1 ADO 对象模型

ADO 顶层有 3 个对象 Connection、Command、Recordset:

- Connection 对象可以指定数据源,建立与数据源的连接、隔离特定的连接。
- Command 对象对数据源执行指定的命令。它可以接受 SQL 语句、表的名称和存储过程的名称,执行 SQL 查询、更新数据、插入记录等。
- Recordset 对象表示的是来自表或命令执行结果的记录全集,操作来自提供者的几乎所有数据。



该模型也反映出了 ADO 的 4 个集合：

- 每个对象都有 Properties（属性）集合。
- Errors（错误）集合包含连接数据库失败时产生的所有 Error 对象。
- Parameters（参数）集合包含 Command 对象的所有 Parameter 对象。
- Fields（字段）集合包含 Recordset 对象的所有 Field 对象。

为了在程序中调用 ADO，需要把 ADO 的 Reference 加入到程序中，如图 1-2 所示。

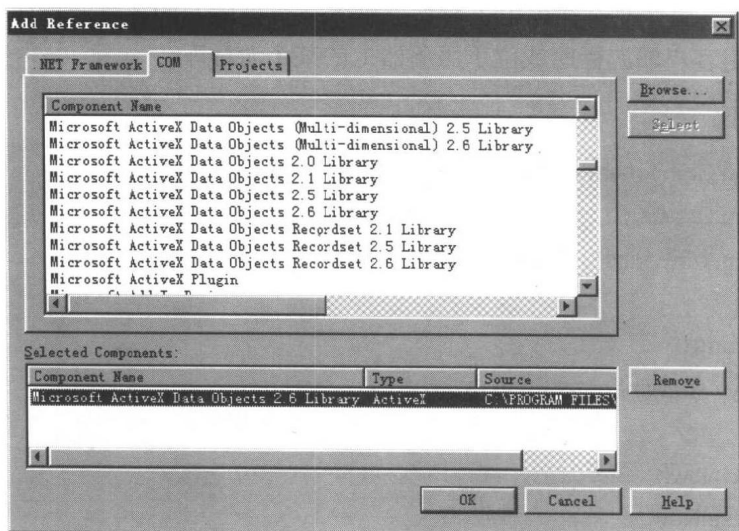


图 1-2 Add Reference 对话框

1.2

ADO 的 Connection 对象

Connection 对象是主要的顶层 ADO 组件，必须成功地打开一个连接到数据源的 Connection 对象，才有可能使用相关的 Command 对象或 Recordset 对象。

1.2.1 Connection 对象的属性

通常在连接数据库之前首先对 Connection 对象的属性进行设置，以便对连接进行配置，下面分别介绍 Connection 对象的属性。

CursorLocation 属性

CursorLocation 属性设置游标类型，通常有两种选择：



- **adUseClient:** 使用由本地游标库提供的客户端游标
本地游标服务允许使用的许多功能可能是驱动程序提供的游标无法使用的，因此使用该设置对于那些将要启用的功能是有好处的。
- **adUseServer:** 使用数据提供者或驱动程序提供的游标。这些游标有时非常灵活，对于其他用户对数据源所做的更改具有额外的敏感性。

CursorLocation 属性仅对设置属性后才建立的连接有影响，更改 CursorLocation 属性不会影响现有的连接。

CursorLocation 属性允许在可用于数据提供者（数据库）的各种游标库中进行选择。通常，可以选择使用客户端游标库或位于服务器上的某个游标库。

示例代码如下：

```
Imports System.ComponentModel
Imports System.Drawing
Imports System.WinForms

Public Class Form1
    Inherits System.WinForms.Form

    Public Sub New()
        MyBase.New
        Form1 = Me
        'This call is required by the Win Form Designer.
        InitializeComponent

        'TODO: Add any initialization after the InitializeComponent() call
    End Sub

    'Form overrides dispose to clean up the component list.
    Overrides Public Sub Dispose()
        MyBase.Dispose
        components.Dispose
    End Sub

    #Region " Windows Form Designer generated code "
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.Container
```



```
Dim WithEvents Form1 As System.Windows.Forms.Form
'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
Private Sub InitializeComponent()
    Me.components = New System.ComponentModel.Container()
    '@design Me.TrayHeight = 0
    '@design Me.TrayLargelcon = False
    '@design Me.TrayAutoArrange = True
    Me.Text = "Form1"
    Me.AutoScaleBaseSize = New System.Drawing.Size(6, 14)
End Sub
#End Region

Public Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Form1.Paint
    Dim MyCNN As ADODB.Connection
    Dim s As String

    MyCNN = New ADODB.Connection()
    MyCNN.CursorLocation = ADODB.CursorLocationEnum.adUseClient
    MyCNN = Nothing
End Sub
End Class
```

Mode 属性

Mode 属性只能在关闭 Connection 对象时设置，它设置对数据的可用权限，可返回当前连接正在使用的访问权限。返回值通常是下列常量之一：

- adModeUnknown: 默认值，表明权限尚未设置或无法确定。
- adModeRead: 表明权限为只读。
- adModeWrite: 表明权限为只写。
- adModeReadWrite: 表明权限为读/写。
- adModeShareDenyRead: 防止其他用户使用读权限打开连接。
- adModeShareDenyWrite: 防止其他用户使用写权限打开连接。



- **adModeShareExclusive**: 防止其他用户打开连接。
 - **adModeShareDenyNone**: 防止其他用户使用任何权限打开连接。
- 当用于客户端 **Connection** 对象时, **Mode** 属性只能设置为 **adModeUnknown**。

State 属性

State 属性是只读的, 其返回对象状态是打开或是关闭, 返回值是下列常量之一:

- **adStateClosed**: 默认, 指示对象是关闭的。
- **adStateOpen**: 指示对象是打开的。
- **adStateConnecting**: 指示 **Recordset** 对象正在连接。
- **adStateExecuting**: 指示 **Recordset** 对象正在执行命令。
- **adStateFetching**: 指示 **Recordset** 对象的行正在被读取。

可以随时使用 **State** 属性确定指定对象的当前状态, 它不仅仅是 **Connection** 对象的属性, 也是 **Command** 对象、**Recordset** 对象的属性。

示例代码如下:

```
Imports System.ComponentModel.  
Imports System.Drawing  
Imports System.WinForms  
Public Class Form1  
    Inherits System.WinForms.Form  
  
    Public Sub New()  
        MyBase.New()  
        Form1 = Me  
        'This call is required by the Win Form Designer.  
        InitializeComponent()  
        'TODO: Add any initialization after the InitializeComponent() call  
    End Sub  
  
    'Form overrides dispose to clean up the component list.  
    Public Overrides Sub Dispose()  
        MyBase.Dispose()  
        components.Dispose()  
    End Sub
```



```
#Region " Windows Form Designer generated code "
```

```
'Required by the Windows Form Designer
```

```
Private components As System.ComponentModel.Container
```

```
Dim WithEvents Form1 As System.Windows.Forms.Form
```

```
'NOTE: The following procedure is required by the Windows Form Designer
```

```
'It can be modified using the Windows Form Designer.
```

```
'Do not modify it using the code editor.
```

```
Private Sub InitializeComponent()
```

```
    Me.components = New System.ComponentModel.Container()
```

```
    '@design Me.TrayHeight = 0
```

```
    '@design Me.TrayLargelcon = False
```

```
    '@design Me.TrayAutoArrange = True
```

```
    Me.Text = "Form1"
```

```
    Me.AutoScaleBaseSize = New System.Drawing.Size(6, 14)
```

```
End Sub
```

```
#End Region
```

```
Public Sub Form1_Paint(ByVal sender As Object, ByVal e As  
System.Windows.Forms.PaintEventArgs) Handles Form1.Paint
```

```
    Dim MyCNN As ADODB.Connection
```

```
    Dim s As String
```

```
    MyCNN = New ADODB.Connection()
```

```
    Me.Text = MyCNN.State.ToString
```

```
    MyCNN = Nothing
```

```
End Sub
```

```
End Class
```

ConnectionTimeout 属性

ConnectionTimeout 属性设置在终止尝试、产生错误或建立连接时所等待的时间，默认值为 15s。



Version 属性

Version 属性返回 ADO 执行的版本号。

DefaultDatabase 属性

DefaultDatabase 属性设置连接的默认数据库，如果没有指定连接的数据库名称，就使用 DefaultDatabase 属性指定的数据库名称。对 SQL Server 而言，其值通常是 pubs。

Provider 属性

Provider 属性指示 Connection 对象提供者的名称，如果没有指定提供者，该属性将默认为 MSDASQL(Microsoft OLE DB Provider for ODBC)。

关闭连接时 Provider 属性为读/写，打开连接时该属性为只读。打开 Connection 对象或访问 Connection 对象的 Properties 集合后该设置才生效；如果该设置无效，将产生错误。

也可以通过下面介绍 ConnectionString 属性或 Open 方法的 ConnectionString 参数设置该属性。



最好不要在多处指定提供者，否则调用 Open 方法连接数据库时可能会产生无法预料的后果。

下面的示例通过打开 3 个使用不同提供者的 Connection 对象演示了 Provider 属性的使用，并且使用 DefaultDatabase 属性设置 Microsoft ODBC 提供者的默认数据库。

```
Public Sub ProviderX()  
    Dim cnn1 As ADODB.Connection  
    Dim cnn2 As ADODB.Connection  
    Dim cnn3 As ADODB.Connection  
    '定义三个 Connection 对象变量  
    cnn1 = New ADODB.Connection  
    cnn1.ConnectionString = "driver={SQL Server};" & _  
        "server=bigsmile;uid=sa;pwd=pwd"  
    '设置 ConnectionString 属性  
    cnn1.DefaultDatabase = "pubs"  
    '设置默认数据库  
    cnn1.Open (strCnn)  
  
    cnn2 = New ADODB.Connection  
    cnn2.Provider = "Microsoft.Jet.OLEDB.3.51"
```