本书含光盘

Visual Basic.NET

二次开发AutoCAD范例精解

张晋西 编著

清华大学出版社

Visual Basic.NET 二次开发 AutoCAD 范例精解

张晋西　编著

清华大学出版社
北　京

内 容 简 介

本书以实例形式介绍用最新的 Visual Basic.NET 中文版二次开发 AutoCAD 的技术。全书包括工程设计中 42 个典型应用实例，从创新、实用、扩展 AutoCAD 功能的角度对实例的构思、解决方法进行了详细的分析说明，内容深入浅出，易读易懂，有助于创新设计和解决工作中的实际问题。相信读者会对各实例的功能和效果产生强烈的兴趣。

为方便初学者，书中也介绍了有关的基础知识。书中所有实例均给出了完整的程序源代码及详细的注释，还可以通过随书附赠的光盘运行所有的程序。读者可以对这些源代码程序任意进行修改、使用。

由于本书对各实例的分析十分详细，因此，对不熟悉 Visual Basic.NET 的读者，也可以轻松地改用其他语言来完成编程。

本书内容新颖，实用性强，既适用于 AutoCAD 二次开发的初学者，也适用于资深的程序开发人员，并可以作为高校计算机辅助设计类课程的教材或供 CAD 技术人员使用。

图书在版编目 (CIP) 数据

Visual Basic.NET 二次开发 AutoCAD 范例精解/张晋西编著. —北京：清华大学出版社，2004.1
ISBN 7-302-07605-7

Ⅰ. V… Ⅱ. 张… Ⅲ. ①BASIC 语言-程序设计②计算机辅助设计-应用软件，AutoCAD
Ⅳ. TP312②TP391.72

中国版本图书馆 CIP 数据核字（2003）第 103829 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
责任编辑：魏江江
封面设计：杨　兮
印 刷 者：北京密云胶印厂
装 订 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所\清华大学出版社出版发行
开　　本：185×260　**印张**：22.75　**字数**：565 千字
版　　次：2004 年 1 月第 1 版　2004 年 1 月第 1 次印刷
书　　号：ISBN 7-302-07605-7/TP・5597
印　　数：1～4000
定　　价：36.00 元（附光盘 1 张）

前　言

Visual Basic 自问世以来，已成为目前开发 Windows 应用程序最为迅速、简捷的程序设计语言，全世界数以百万计的设计人员正在用它开发各种类型的软件。我国一些高校已经把 Visual Basic 语言列入教学计划。Visual Basic.NET 又称 Visual Basic 7.0，是 Visual Basic 的最新版本。

自 AutoDesk 公司开发的 AutoCAD 工程图形处理软件面世以来，以其完善的绘图功能，易学易用的特点，受到了广大工程技术人员的普遍欢迎，AutoCAD 及其图形格式已成为一种事实上的国际工业标准。

用 Visual Basic.NET 二次开发 AutoCAD，是基于新的 ActiveX 自动化界面技术（ActiveX Automation Interface）的。ActiveX 技术使得我们可以通过 AutoCAD 显示出来的信息，通过其他计算机语言编程，从 AutoCAD 的内部或外部来控制、操纵它们。

就 AutoCAD 软件本身来说，虽然在其自身的领域具备强大的功能，但是，在工程设计中，常常需要结合专业情况，对仅靠 AutoCAD 自身不能或不易做到的功能进行二次开发。例如，通过二次开发，可以用程序写出齿轮轮廓曲线的方程，根据需要绘制不同数目的点来精确模拟轮廓曲线；可以对复杂的机械运动，例如机器人，进行参数化三维动态模拟；进一步还可以根据自己专业特点，进行有价值的理论研究，开发出解决实际工程问题的有价值的软件。

不论是对编程爱好者、工程技术人员还是科研工作者来说，AutoCAD 二次开发均是一门值得高度关注的技术。借助 AutoCAD 软件本身的强大功能，可以花较小的力气，得到可观的成果。例如，参数化三维实体建模，既可用于制造工程，也可用于三维图形动态效果设计，与其他的方法相比，从编程的角度来说，用新推出的 Visual Basic.NET 进行 AutoCAD 二次开发，可能是最简单、最方便的了。其次，借助 AutoCAD 广泛的应用群体，也容易使自己的劳动成果得到承认。总之，这就好比站在巨人肩膀上，可以看得更远，更容易获得成功。

目前有关用 Visual Basic 进行 AutoCAD 二次开发的书，大多只介绍基础知识。但是，在实践中将会发现，要用先进的手段，创造性地解决问题，仅有基础知识是不够的。

本书的着力点在创新、实用以及 AutoCAD 功能扩展三个方面。书中 42 个实例在设计应用、AutoCAD 功能的扩展、解决问题的技巧等方面，均有所创新，并对其思路做了详细的分析和说明。希望通过本书，能够帮助读者提高在工作、学习中善于发现问题与分析问题的能力，能够找出创新点，解决用一般方法不能解决的实际问题。

书中一些例题，采用了作者多年教学、科研以及获奖的成果。所选例题，力求做到既具有较高的实用与创新价值，含有一定的编程技巧，又深入浅出，易学易懂。

由于作者水平有限，疏漏和错误之处在所难免，恳请读者批评指正。作者电子邮箱：zjx2002cq@sina.com。

张晋西

2003 年 11 月于重庆工学院

目　录

实例 1　连接 AutoCAD

实例功能

- 创建新的 Visual Basic.NET 项目
- 引用 AutoCAD 对象库
- 创建 AutoCAD 对象变量，启动运行 AutoCAD

分析说明

用 Visual Basic.NET 进行 AutoCAD 二次开发，是基于新的 ActiveX 自动化界面技术（ActiveX Automation Interface）的。AutoCAD ActiveX 技术使得我们可以通过 AutoCAD 显示出来的信息，用其他的应用程序（如 Visual Basic.NET）通过编程从 AutoCAD 内部或外部来控制、操纵 AutoCAD。我们可以用 Visual Basic.NET 将 AutoCAD 当成 Visual Basic.NET 程序中的一个图形窗口，对其进行打开、绘图、编辑、打印、关闭等操作。

本实例介绍怎样创建一个新的 Visual Basic.NET 项目；Visual Basic.NET 如何连接 AutoCAD，包括引用 AutoCAD 对象库、创建 AutoCAD 对象变量、启动运行 AutoCAD。本实例的内容是二次开发的基础，以后各实例都要用到其中的一些代码。

编程步骤

1．创建一个新的 Visual Basic.NET 项目

运行 Visual Basic.NET，选择【文件】/【新建】/【项目】，出现“新建项目”对话框，如图 1.1 所示。在“项目类型”栏选择【Visual Basic 项目】，在“模板”栏选择【Windows

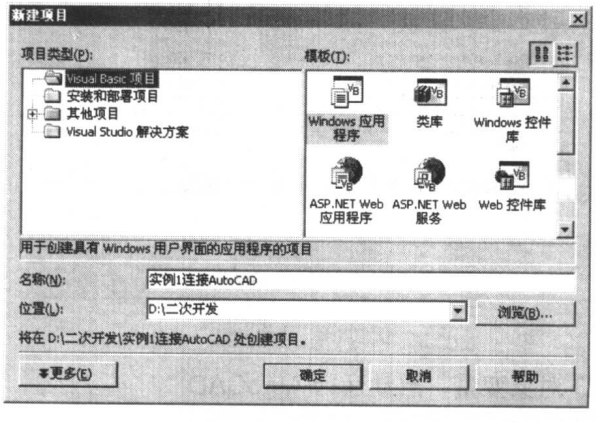

图　1.1

应用程序】，然后在下面“名称”栏输入要创建的项目的名称“实例 1 连接 AutoCAD”，在“位置”栏输入项目文件所在的路径“D:\二次开发”，或者从列表中选择一个位置。默认情况下，新项目将创建于 Visual Basic.NET 菜单【工具】/【选项】/【环境】/【项目和解决方案】中设置的位置。单击【确定】按钮，得到如图 1.2 所示 Visual Basic.NET 编程环境。

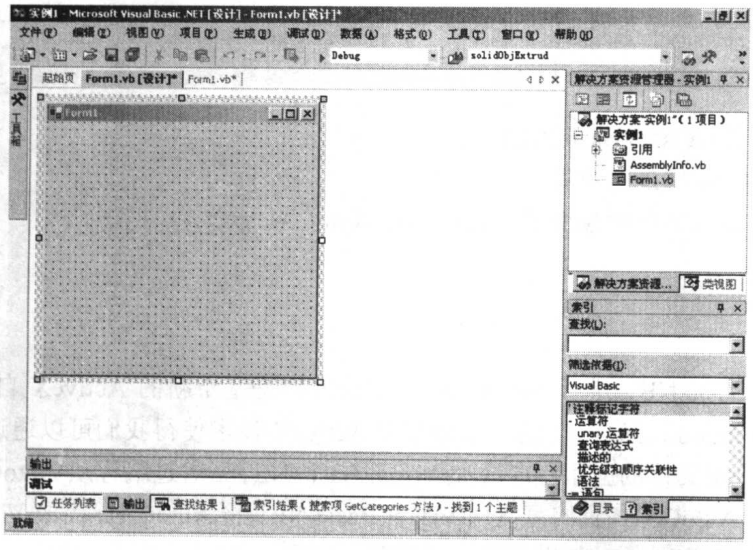

图 1.2

2. 引用 AutoCAD 对象库

在编写 Visual Basic.NET 代码前，需要在 Visual Basic.NET 编程环境中引用 AutoCAD 对象库。

在 Visual Basic.NET 编程环境中选择菜单【项目】/【添加引用】/COM，再选择 AutoCAD2000 Type Library，如图 1.3 所示，然后单击【选择】和【确定】按钮。

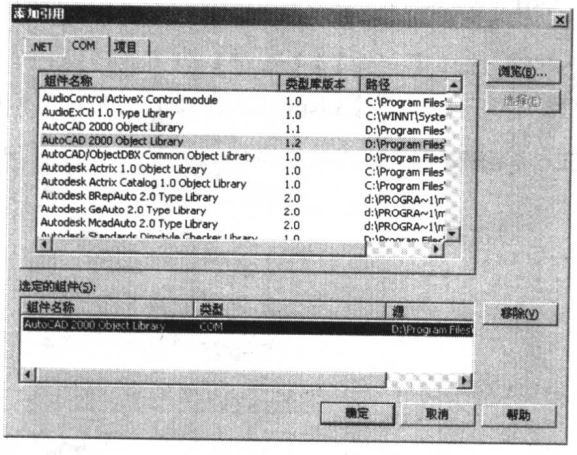

图 1.3

3. 创建 AutoCAD 对象变量，启动运行 AutoCAD

在窗体中添加一个命令按钮 Button1，如图 1.4 所示，双击 Button1，在出现的代码窗

体的 Button1_Click 事件过程中输入下面代码（见程序清单 1.1），调用“连接 AutoCAD”过程。若只有一个窗体要使用 AutoCAD 对象变量，还应在代码窗体中所有过程前定义 AutoCAD 对象变量 AcadApp，这样，变量 AcadApp 在该窗体的所有过程或事件中均有效。

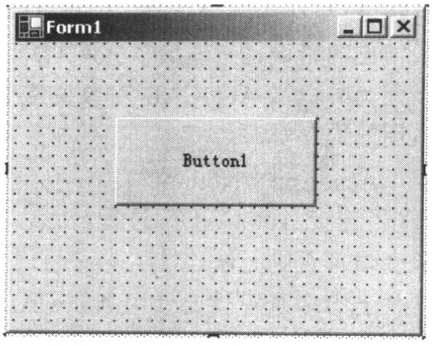

图　1.4

程序清单 1.1：定义 AutoCAD 对象变量，调用“连接 AutoCAD”过程（Button1 按钮）

```
Dim AcadApp As AutoCAD.AcadApplication
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Call 连接AutoCAD()
End Sub
```

创建 AutoCAD 对象变量 AcadApp、启动运行 AutoCAD 由“连接 AutoCAD”过程实现，见程序清单 1.2。

程序清单 1.2：创建 AutoCAD 对象变量，启动运行 AutoCAD（Sub 连接 AutoCAD）

```
Sub 连接AutoCAD()
On Error Resume Next
AcadApp = GetObject(, "AutoCAD.Application")
If Err.Number Then
    Err.Clear()
    AcadApp = CreateObject("AutoCAD.Application")
    If Err.Number Then
        MsgBox("不能运行AutoCAD,请检查是否安装了AutoCAD")
        Exit Sub
    End If
End If
AcadApp.Visible = True'界面可视
AcadApp.WindowState = AutoCAD.AcWindowState.acMax '界面最大化
AppActivate(AcadApp.Caption) '显示AutoCAD界面
End Sub
```

上面代码将 AutoCAD 对象引用赋给变量 AcadApp，创建 AutoCAD 对象变量。如果 AutoCAD 已经启动运行，则 GetObject 函数返回对 AutoCAD 应用程序对象的引用，否则发生一个错误，使 Err.Number 非零（为真），用 Err.Clear 方法清除错误信息，然后用 CreateObject 函数创建 AutoCAD 应用程序对象的引用，如果成功，则 AutoCAD 被启动，否则用 MsgBox 函数给出一个错误信息提示框，然后退出程序。这里使用了比较复杂的条件语句，目的是确保只启动运行一次 AutoCAD，避免重复启动运行 AutoCAD。

AutoCAD 不能被启动的原因有多种，如 AutoCAD 在系统注册表中有错误，系统资源

紧张等，可以用消息框与 Err 对象的 Description 属性具体显示，如 MsgBox(Err.Description)。最常见的错误是未安装 AutoCAD，因此这里提示“检查是否安装了 AutoCAD”。

另外，当 AutoCAD 对象变量 AcadApp 不再使用时，应该用下面语句释放该对象变量占用的内存资源：

```
AcadApp = Nothing
```

“连接 AutoCAD”过程应该在用到 AcadApp 变量之前，先行调用、创建 AcadApp 对象变量，否则无法连接 AutoCAD。

若有多个窗体要使用 AutoCAD 对象变量，可以选择【项目】/【添加新项】/【本地项目】/【模块】，在项目中添加一全局模块 Module1，用下面语句定义 AutoCAD 对象变量 AcadApp：

```
Public AcadApp As AcadApplication
```

然后在 Module1 代码窗体输入上面“连接 AutoCAD”过程代码，就可以在任何地方调用该过程了。

注意，本书后面所有实例运行时，均要先调用“连接 AutoCAD”过程，创建 AutoCAD 对象变量。

也可以简单地用下面语句创建一个 AutoCAD 对象变量：

```
AcadApp = New AutoCAD.AcadApplication()
```

由于每执行一遍该语句，就会加载一次 AutoCAD，这样会浪费时间和系统资源，因此不推荐使用。

Visual Basic.NET 与 AutoCAD 连接后，就可以利用该 AutoCAD 对象及其下级对象的属性、方法等，完成用 Visual Basic.NET 语言在 AutoCAD 环境中的图形绘制、编辑等操作。

程序运行结果

按下 F5 键，运行程序，单击图 1.4 中【Button1】按钮，将会看到 AutoCAD 界面出现。如图 1.5 所示，完成 AutoCAD 的连接与启动。

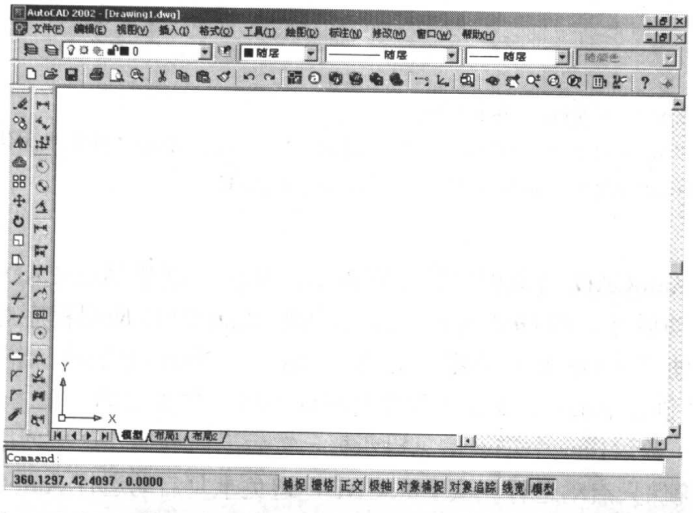

图 1.5

实例 2　用 Access 数据库管理图形

实例功能

- 编程实现 Access 数据库的创建、录入、显示、查询
- 参数化绘图
- 鼠标单击选择表格显示的数据库记录，根据记录内容绘图
- 在 AutoCAD 中拾取实体并将其信息录入数据库

分析说明

AutoCAD 绘图软件与 Access 数据库目前均拥有十分广泛的用户。本实例以一个三维圆筒实体图形为例，说明用 Access 数据库来管理 AutoCAD 图形，实现用数据库的信息绘图，或将 AutoCAD 中已经完成的图形信息录入数据库。对其他用户或管理者来说，只需提供要绘制的图形信息数据库，就可以快速完成绘图；对绘图技术人员来说，可以将已完成的图形信息用数据库形式保存。

本例以简便实用的方式介绍了数据访问对象 DAO（Database Access Object）编程以及最新的由 ADO（ActiveX Data Object）升级而来的 ADO.NET 数据库访问对象编程，完成了 Access 数据库的创建、录入、显示、查询的程序设计。

数据查询采用 SQL 语句。结构化查询语言 SQL（Structure Query Language）功能强大，应用广泛，内容也较多，但常用的语句较简单，易于掌握应用。通过本例的学习，可做到举一反三。

在界面的方便性和实用性方面，采用 DataGrid 控件以表格显示数据库内容或查询结果，只需用鼠标单击表格某行，就可以根据该记录的数据绘图；用鼠标单击选中 AutoCAD 图形，就可以将其数据录入数据库存储起来。

编程步骤

1．创建一个新项目

首先，创建一个新项目，名为“实例 2Access 数据库管理图形”，然后选择菜单【项目】/【添加引用】/COM，选择 AutoCAD2000 Type Library，单击【选择】、【确定】按钮，引用 AutoCAD 对象库。

给窗体添加控件，如图 2.1 所示。然后定义窗体级变量，在 Form1_Load 事件中给控件赋初值。见程序清单 2.1。

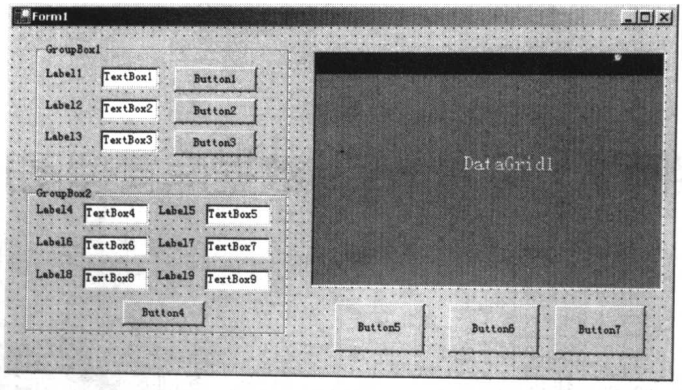

图 2.1

程序清单 2.1：定义窗体级变量，赋初值（Form1_Load 事件）

```
Dim AcadApp As AutoCAD.AcadApplication
Dim R1, R2, H As Double '圆筒外径、内径、高
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.TextBox1.Text = 20
    Me.TextBox2.Text = 10
    Me.TextBox3.Text = 30
    Me.TextBox4.Text = 1
    Me.TextBox5.Text = 100
    Me.TextBox6.Text = 1
    Me.TextBox7.Text = 100
    Me.TextBox8.Text = 1
    Me.TextBox9.Text = 100

    Me.Text = "实例2 Access数据库管理图形"
    Me.GroupBox1.Text = "创建数据"
    Me.GroupBox2.Text = "数据查询"

    Me.Label1.Text = "外径"
    Me.Label2.Text = "内径"
    Me.Label3.Text = "高"

    Me.Label4.Text = "外径介于"
    Me.Label5.Text = "与"
    Me.Label6.Text = "内径介于"
    Me.Label7.Text = "与"
    Me.Label8.Text = "高介于"
    Me.Label9.Text = "与"

    Me.Button1.Text = "录入数据库"
    Me.Button2.Text = "绘图"
    Me.Button3.Text = "新建数据库"

    Me.Button4.Text = "数据查询"
```

```
    Me.Button5.Text = "选择记录并绘图"
    Me.Button6.Text = "拾取实体并添加入数据库"
    Me.Button7.Text = "结束"

    Me.DataGrid1.CaptionText = "数据库显示"
End Sub
```

2. 创建数据库

本例采用 Access 数据库。这里介绍两种创建方法，一种采用 Access 数据库向导创建，另一种用编程创建。本实例采用编程创建数据库。

方法 1：采用 Access 数据库向导创建数据库

用 Microsoft Access2000 创建数据库。运行 Access，在出现的对话框中选择“空 Access 数据库”，如图 2.2 所示。在“文件新建数据库”对话框中，在“保存位置”栏选择启动应用程序的可执行文件的路径，它在当前项目文件所在文件夹下面的 bin 文件夹内（这里为 D:\二次开发\实例 2Access 数据库管理图形\bin），在“文件名”栏填入将要建立的数据库名，这里取名“圆筒数据库”。然后，在“数据库” 对话框中选择【使用设计器创建表】，表设计器如图 2.3 所示，录入格式如图 2.4 所示，表名为“圆筒”。该表包括 ID、“外径”、“内径”、“高”字段，ID 字段数据类型设置为长整型，其余为双精度型。

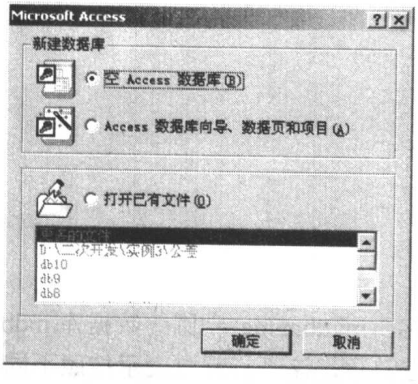

图　2.2

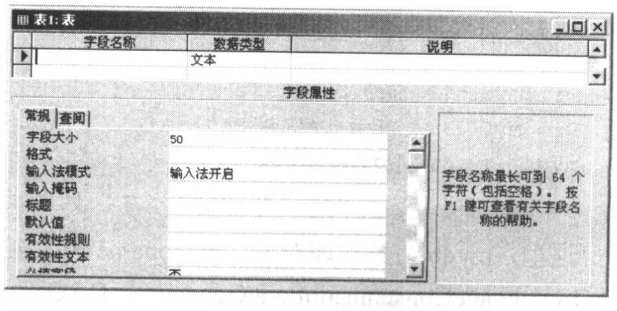

图　2.3

图　2.4

方法 2：编程创建数据库

用程序代码创建一个数据库，包括创建数据库对象、创建表、给表添加字段、添加记

录等步骤。

■ 创建数据库对象

由于使用面向数据库对象编程，而数据库对象包含在 DAO 库中，Visual Basic.NET 不会在程序中自动应用 DAO，因此在程序设计阶段要将 DAO 库包含到项目中。

首先，选择【项目】/【添加引用】/COM，选择组件名称为 Microsoft DAO 3.6 Object Library 的 DAO 对象库（Access2000 已经将 DAO 升级为 3.6），如图 2.5 所示。

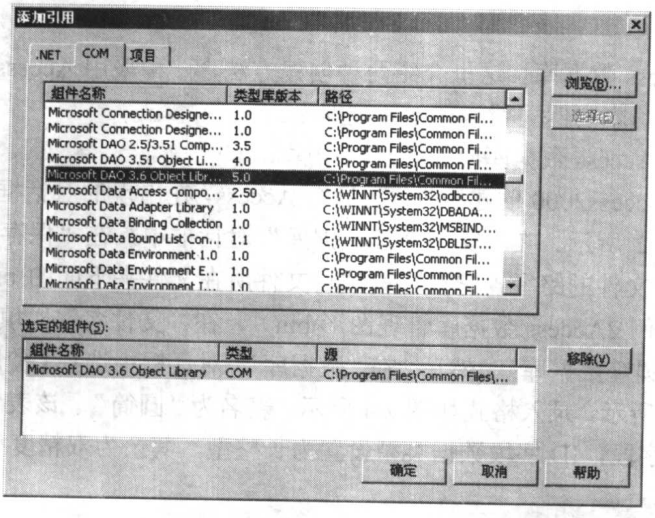

图 2.5

然后，用下面语句创建数据库对象：

```
Dim mydb As DAO.Database
Dim myDBE As New DAO.DBEngine()
mydb = myDBE.Workspaces(0).CreateDatabase(Application.StartupPath + "\" +
mydbname, DAO.LanguageConstants.dbLangGeneral)
```

这里，mydb 为数据库对象，mydbname 为数据库名，mydbname = "圆筒数据库.mdb"，DAO.LanguageConstants.dbLangGeneral 为 DAO 的语言常数，该语言常数表示按照不同的语言习惯设置数据库的排列顺序。在此选择 dbLangGeneral，表示常用的英、德、法等语言设置。Application.StartupPath 为启动应用程序的可执行文件的路径。

■ 获取启动应用程序的可执行文件的路径

下面介绍在 Visual Basic.NET 中获取文件路径的方法，这些知识在数据的读写中要常常用到。

用 Application.StartupPath 属性可以获取启动应用程序的可执行文件的路径。

若要获得当前项目文件所在路径，用下面语句实现：

```
mypath=Application.StartupPath.Remove(Application.StartupPath.Length-4, 4)
```

对本例而言，Application.StartupPath 返回应用程序的可执行文件的路径字符串“D:\二次开发\实例 2Access 数据库管理图形\bin”，最后四个字符为“\bin”，Remove 方法从此路径中的指定位置开始删除指定数目的字符。因此当前项目文件所在路径 mypath 值为“D:\

二次开发\实例 2Access 数据库管理图形”。

本例将数据库文件“圆筒数据库.mdb”建立在应用程序的可执行文件的路径“D:\二次开发\实例 2Access 数据库管理图形\bin”下。

■　创建表

用下面语句创建数据库的一张表：

```
Dim mytbl As DAO.TableDef
mytablename = "圆筒"
mytbl = mydb.CreateTableDef(mytablename)
```

■　给表添加字段

用下面语句给表添加字段：

```
myfd = mytbl.CreateField("外径", DAO.DataTypeEnum.dbDouble)
mytbl.Fields.Append(myfd)
```

这里，外径为字段名，DAO.DataTypeEnum.dbDouble 表示数据类型为双精度型，然后用下面语句将该表附加给数据库的表集合：

```
mydb.TableDefs.Append(mytbl)
```

■　给表添加记录

用下面语句给表添加记录。

打开数据库：

```
mydb = myDBE.Workspaces(0).OpenDatabase(Application.StartupPath  + "\" +
mydbname)
mytablename = "圆筒"
```

添加记录：

```
  Dim myrec As DAO.Recordset
myrec = mydb.OpenRecordset(mytablename, DAO.RecordsetTypeEnum.dbOpenDynaset)
  myrec.AddNew()
  myrec.Fields(0).Value = myrec.RecordCount + 1
  myrec.Fields(1).Value = R1
  myrec.Fields(2).Value = R2
  myrec.Fields(3).Value = H
```

这里将一条记录“ID（标号），R1（外径），R2（内径），H（高）”的值分别添加到了该表的第 1 到第 4 个字段。

下面的程序清单 2.2 实现创建数据库、创建表、给表添加字段功能；程序清单 2.3 实现给表添加记录功能；程序清单 2.4 实现将文本框中输入的数据录入数据库功能。

程序清单 2.2：创建数据库、表、字段（【新建数据库】按钮）

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button3.Click
    If MsgBox("将删除原来数据库,继续?", MsgBoxStyle.YesNo) = MsgBoxResult.No
    Then
        Exit Sub
    End If

    Dim mydb As DAO.Database '数据库对象
    Dim mydbname As String '数据库名

    mydbname = "圆筒数据库.mdb"

    If Dir(Application.StartupPath + "\" + mydbname) <> "" Then Kill
    (Application.StartupPath + "\" + mydbname)

    Dim myDBE As New DAO.DBEngine()

    '创建数据库对象
    mydb = myDBE.Workspaces(0).CreateDatabase(Application.StartupPath + "\" +
    mydbname, DAO.LanguageConstants.dbLangGeneral)

    Dim mytbl As DAO.TableDef '表对象
    Dim mytablename As String '表名
    Dim myfd As DAO.Field '字段对象
    mytablename = "圆筒"
    '创建表()
    mytbl = mydb.CreateTableDef(mytablename)

    '添加字段()
    myfd = mytbl.CreateField("ID", DAO.DataTypeEnum.dbLong)
    mytbl.Fields.Append(myfd)

    myfd = mytbl.CreateField("外径", DAO.DataTypeEnum.dbDouble)
    mytbl.Fields.Append(myfd)

    myfd = mytbl.CreateField("内径", DAO.DataTypeEnum.dbDouble)
    mytbl.Fields.Append(myfd)

    myfd = mytbl.CreateField("高", DAO.DataTypeEnum.dbDouble)
    mytbl.Fields.Append(myfd)

    mydb.TableDefs.Append(mytbl)

    mydb.Close()

End Sub
```

程序清单 2.3： 给表添加记录,在 DataGrid 控件中显示表的记录（Sub 添加记录）

```
Sub 添加记录()

    Dim mydb As DAO.Database '数据库对象
    Dim mydbname As String '数据库名
```

```
    mydbname = "圆筒数据库.mdb"
    Dim myDBE As DAO.DBEngine
    myDBE = New DAO.DBEngine()

    On Error GoTo ErrorHandler '防止当前应用程序文件夹内无数据库文件
    '打开数据库()
    mydb = myDBE.Workspaces(0).OpenDatabase(Application.StartupPath + "\"
    + mydbname)

    Dim mytbl As DAO.TableDef '表对象
    Dim mytablename As String '表名
    Dim myfd As DAO.Field '字段对象

    mytablename = "圆筒"

    '添加记录
    Dim myrec As DAO.Recordset '记录
    myrec = mydb.OpenRecordset(mytablename,DAO.RecordsetTypeEnum.dbOpenDynaset)

    myrec.AddNew()
    myrec.Fields(0).Value = myrec.RecordCount + 1 '第I+1条记录
    myrec.Fields(1).Value = R1
    myrec.Fields(2).Value = R2
    myrec.Fields(3).Value = H
    myrec.Update()

    mydb.Close()

    '在DataGrid中显示表的所有记录
    Dim mySQL As String 'SQL查询字符串
    'SQL查询语句
    mySQL = "SELECT * FROM " & mytablename

    Dim myadocmd As System.Data.OleDb.OleDbDataAdapter
    myadocmd = New Data.OleDb.OleDbDataAdapter(mySQL, "Provider=Microsoft.
    jet.OLEDB.4.0; Data Source=" & Application.StartupPath + "\圆筒数据库.mdb")
    Dim myds As DataSet = New DataSet()
    myadocmd.Fill(myds, "圆筒")
    Me.DataGrid1.DataSource = myds.Tables.Item(0)
    Me.DataGrid1.Update()
    Exit Sub
ErrorHandler:
    MsgBox("请检查是否已建有数据库，否则请先按新建数据库按钮")

End Sub
```

程序清单 2.4：将文本框中输入的数据录入数据库（【录入数据库】按钮）

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
```