如何用 VB 在 C/S 模式下实现远程数据访问

沈晓兵 （哈尔滨工程大学自动化学院 150001）

VISUAL BASIC 作为一种流行的开发工具，在开发 C/S 应用程序方面有强大的功能。概括的说在 VB 中有四种编程模型可以实现远程数据访问，它们是数据访问对象(DAO)和数据控件(DATA)、远程数据对象（RDO）和远程数据控件（RDC）、开放数据库互连（ODBC）API 和 VBSQL API。本文将结合一个简单的例子，主要讨论如何使用前两种编程模型，对后两种仅做简要介绍。

假设服务器的操作系统是 WINDOWS NT 4.0，服务器名为 402，并且在服务器上已有一名为 example 的 SQL SERVER 6.5 数据库，其中有一名为 student 的表，表中存放着姓名、年龄等信息。下面将要讨论如何在 WINDOWS 95 客户机上访问该数据库。由于篇幅所限，示例将仅打开连接，其他内容读者可自行编写。

一、使用 DAO 和 DATA 控件

1. 通过 Microsoft Jet 引擎使用 DAO

首先在窗体的公共声明部分加入下面的声明：

```
Dim demomdb As Database
Dim demoquery As QueryDef
Dim demorecord As Recordset
```

下面将要建立与服务器的连接，为了叙述方便，假设在窗体的加载事件中建立连接，在窗体的卸载事件中取消连接，至于记录集的显示，读者可自行选择编程，下面的各段程序都将遵守这个假设。

```
Private Sub Form-Load()
    ' 打开一个现有的数据库，用来创建 QueryDef 对象
    Set demomdb = OpenDatabase("c:\use.mdb")
    ' 创建一个查询，用于从 Microsoft SQL Server 数据库返回数据
    Set demoquery = demomdb.CreateQueryDef("")
    demoquery.Connect = "ODBC;DATABASE=example;UID=sa;PWD=;DSN=example"
    demoquery.SQL = "SELECT * FROM student"
    '打开记录集(Dynaset 型)
    Set demorecord = demoquery.OpenRecordset()
    demorecord.MoveFirst
End Sub
Private Sub Form-Unload(Cancel As Integer)
    demomdb.QueryDefs.Delete ""
    demomdb.Close
End Sub
```

注 1：若在工程中没有 Database 等 DAO 类型定义，请按如下操作并在 VB 中打开"工程"菜单，单击"引用"菜单，选中 Microsoft DAO 3.5 Object Library，单击"确定"按钮。

注 2：若尚未建立 DSN，请按如下操作并单击"开始"按钮，选中"设置"、"控制面板"，双击"32 位 ODBC"，在"用户 DSN"下，单击"添加"按钮，选择"SQL SERVER"单击"完成"按钮，在"Data Source Name"框中输入 example，在"server"框中输入 402，单击"options"按钮，在"Database Name"框中输入 example，单击"OK"按钮，单击"确定"按钮。

2. 通过 ODBCDirect 使用 DAO

首先在窗体的公共声明部分加入下面的声明：

```
Dim demorecord As Recordset
Dim demowrk As Workspace
Dim demoodbc As Connection
```

实现连接的程序如下：

```
Private Sub Form-Load()
'创建 ODBCDirec Workspace 对象
Set demowrk = CreateWorkspace("NewODBCWorkspace", "sa", "", dbUseODBC)
'打开 Connection 对象
Set demoodbc = demowrk.OpenConnection("Connection1", , , -
"ODBC;DATABASE=example;UID=sa;PWD=;DSN=example")
'打开记录集(Dynaset 型)
Set demorecord = demoodbc.OpenRecordset("select * from student"-
```

```
, dbOpenDynamic)
End Sub
Private Sub Form-Unload(Cancel As Integer)
demorecord.Close
demoodbc.Close
demowrk.Close
End Sub
```

3. 使用 DATA 控件

确定在窗体中已添加一个 DATA 控件，实现连接的程序如下：

```
Private Sub Form-Load()
'设置 DATA 控件的几个必须属性
Data1. Connect = " ODBC; DATABASE = example; UID = sa; PWD = ; DSN = example"
Data1.RecordSource = "SELECT * FROM student"
Data1.Refresh
End Sub
```

二、使用 RDO 和 MSRDC 控件

1. 使用 RDO OpenConnection 方法

首先在窗体的公共声明部分加入下面的声明：

```
Dim democn As rdoConnection
Dim demoen As rdoEnvironment
Dim demorecord As rdoResultset
```

实现连接的程序如下：

```
Private Sub Form-Load()
'创建 rdoEnvironment 对象
Set demoen = rdoEnvironments(0)
'建立连接，建立 rdoConnection 对象
Set democn = demoen. OpenConnection ( dsName: =""", Prompt: = rdDriverNoPrompt, -Connect: = "uid = sa; pwd = ; driver = {SQL Server};" & server = 402; database = example;")
'返回结果记录集(与 Recordset 类似)
Set demorecord = democn. OpenResultset("select * from student"-, rdOpenDynamic)
End Sub
Private Sub Form-Unload(Cancel As Integer)
democn. Close
demoen. Close
End Sub
```

注 3：若在工程中没有 rdoconnection 等 RDO 类型定义，请按如下操作并在 VB 中打开"工程"菜单，单击"引用"菜单，选中 Microsoft Remote Data Object 2.0，单击"确定"按钮。

2. 使用 RDO EstablishConnection 方法

首先在窗体的公共声明部分加入下面的声明：

```
Dim democn As New rdoConnection
Dim demoqd As New rdoQuery
Dim demorecord As rdoResultset
```

实现连接的程序如下：

```
Private Sub Form-Load()
democn. Connect = "uid = sa; pwd = ; DSN = example;"
democn. CursorDriver = rdUseOdbc
democn. EstablishConnection rdDriverNoPrompt
'返回结果记录集(与 Recordset 类似)
Set demorecord = democn. OpenResultset("select * from student"-
, rdOpenDynamic)
End Sub
Private Sub Form-Unload(Cancel As Integer)
democn. Close
demoqd. Close
End Sub
```

3. 使用 MSRDC 控件

首先需要在工程中添加 MSRDC 控件，打开"工程"菜单，单击"部件"菜单，选中 Microsoft RemoteData Control 2.0，单击"确定"按钮。然后将 MSRDC 控件加入窗体，按如下设置其属性：单击"自定义"，在"数据源"框中选择 example，在"用户名"框中输入 sa，密码框空白(此设置是由服务器上 SQL SERVER 决定的，读者应根据实际情况具体调整)，在"SQL"框中输入 select * from student，单击"确定"按钮，这样不用编写任何程序代码，连接就建立起来了，读者可简单地加入几个文本框通过设置 DataField 属性与之关联来观察结果。

三、使用 ODBC API

尽管仍然能够在 Visual Basic 中使用 ODBC API 创建应用程序，但是它的优点无法与其他对象模型接口相比，在 32 位平台上尤其如此。一般来说有两种使用 ODBC API 的方式：使用 ODBC API 的扩展 RDO，在

RDO 或 ODBCDirect 应用程序中可以使用 ODBC API 函数；直接利用 ODBC API 进行程序设计，在创建程序时，可以使用不依赖于 RDO 或其他对象风格接口的 ODBC API。尽管可以获得额外的速度、灵活性和内存，但是也应该考虑到所有 ODBC API 接口代码都是用 Visual Basic 编写的，这也会对程序大小和性能造成负面影响。

四、使用 VBSQL API

如果用户的客户/服务器应用程序只需要连接 Microsoft SQL Server，可以使用 Visual Basic Library for SQL Server（VBSQL）、VBSQL 控件和 DB－Library。它可用来打开一个或多个连接、提交查询、创建游标、处理结果集、执行存储过程以及执行成批复制操作。但是由于 VBSQL 所基于的 DB－Library 接口被认为是过时了的技术，因此 Microsoft 鼓励使用其他编程模型，例如远程数据对象等来代替它。

从上述讨论我们不难发现，四种程序设计模型最显著的不同在于它们编写代码利用数据识别绑定控件的方法。运用 DAO 或 RDO 程序设计模型，特别是当使用了 Visual Basic 或第三方控件时，将显著地减少开发和测试应用程序所花费的时间，开发者应根据实际情况具体选择，有关更多更具体的信息读者可参见 VB 的联机帮助。以上程序在 VB5.0 中文企业版中调试通过。

（来稿时间：1998 年 8 月）