

ODBC 技术在 Windows 下的应用

毛惊琼 (上海大学计算机学院 200072)

朱尚凑 (上海大学经济管理学院 200072)

摘要:本文首先探讨了 Visual Basic 中用 ODBC API 开发数据库应用程序的方法,然后简介了 Windows 系统剪贴板技术,最后给出了一个利用 ODBC API 直接编程技术将数据库数据读到 Windows 系统下的剪贴板上的主要模块代码,解决了一般应用程序不能共享数据库的难题。

关键词:ODBC API 剪贴板

1. 引言

ODBC(Open DataBase Connectivity, 开放数据库互连)允许数据库与应用程序之间实现数据交换,实现应用程序对不同 DBMS 的共享,应用工具如 Word、Excel 等都支持该功能。但尚有许多应用程序不具有该功能,因而无法利用 ODBC 良好的数据库独立性,无法共享数据库。

Windows 系统的剪贴板对于所有的应用程序都是有效的。因此,如将数据库数据读到剪贴板上就能解决一般应用程序不能共享数据库的问题。

2. Visual Basic 中用 ODBC API 开发应用程序

使用 ODBC 来编写 Visual Basic 程序的机制按编程的难易程度(从易到难)可以分为三类:使用数据库控制

项和绑定控制项,使用数据库对象变量进行编程及直接调用 ODBC API。

相对于前两种方法,直接调用 ODBC API 对数据库操作提供了最大程度的直接控制,因此该方法具有最好的性能。对于在复杂性、综合性及性能方面提出附加要求的重要数据库应用程序来说直接使用 ODBC API 函数是很适当的。

Visual Basic 调用 ODBC API 进行编程的一般过程是:

·声明将使用的 DLL 函数

例:Declare Function SQLAllocEnv Lib "odbc.dll" (env As Long) As Integer

- 对 ODBC 子系统进行初始化

例: Global henv As long

Dim rc As Integer

rc = SQLAllocEnv(henv)

- 建立用户所需的 ODBC 数据库连接

例: rc = SQLAllocConnect(henv, hdbc)

rc = SQLConnect(hdbc, "MyDataBase", 10, "I-mAUser", 7, "MyPassword", 10)

- 获取有关数据源驱动程序及数据源本身的信息

例: Dim DataSource As String, Description As String

Dim DataSourceLen As Integer, DescriptionLen As Integer

rc = SQLDataSources(henv, 2, DataSource, Len(DataSource), DataSourceLen, Description, Len(Description), DescriptionLen)

- 检索并建立用户所需的驱动程序选项

例: rc = SQLSetConnectOption(hdbc, SQL-ODBC-CURSOR, SQL-CUR-USE-ODBC)

- 准备并提交 SQL 语句

例: rc = SQLAllocStmt(hdbc, hstmt)

sqlstr = "SELECT * FROM Tablename"

rc = SQLExecDirect(hstmt, sqlstr, Len(sqlstr))

- 检索结果集及其有关信息

例: Dim strEmpName As String * 20

Dim lngNameLen As Long

rc = SQLFetch(hstmt)

rc = SQLGetData(hstmt, 1, SQL-C-CHAR, strEmpName, 20, lngNameLen)

- 中止 SQL 语句处理

例: rc = SQLFreeStmt(hstmt, SQL-DROP)

- 中断以前建立的数据源连接

例: rc = SQLDisconnect(hdbc)

rc = SQLFreeConnect(hdbc)

- 切断应用程序与 ODBC 子系统的关系

rc = SQLFreeEnv(henv)

3. 利用 ODBC API 编程的实例

下面代码是利用 ODBC API 在 VB 中实现将数据库中的数据读入 Windows 下的剪贴板中, 据此可以达到在

应用程序中共享数据库, 解决了某些应用程序不支持 ODBC 的难题。

'变量定义

Global henv As Long * 定义环境句柄变量

Global hdbc As Long * 定义连接句柄变量

Global hstmt As Long * 定义语句句柄变量

Global rc As Integer

Global tableSelect As Integer

Global TableName As String

Global SourceName As String

Dim szSqlStr As String

Dim strEmpName As String * 20

Dim lngNameLen As Long

Dim intCols As Integer

Dim i As Integer

Dim retcode As Integer

Dim msg As String

'ODBC API 声明

Declare Function SQLAllocEnv Lib "odbc.dll" (env As Long) As Integer

Declare Function SQLAllocConnect Lib "odbc.dll" (ByVal env As Long, dbc As Long) As Integer

Declare Function SQLAllocStmt Lib "odbc.dll" (ByVal dbc As Long, stmt As Long) As Integer

Declare Function SQLConnect Lib "odbc.dll" (ByVal dbc As Long, ByVal szDSN As String, ByVal cbDSN As Integer, ByVal szUID As String, ByVal cbUID As Integer, ByVal szAuthStr As String, ByVal cbAuthStr As Integer) As Integer

Declare Function SQLExecDirect Lib "odbc.dll" (ByVal stmt As Long, ByVal szSqlStr As String, ByVal cbSqlStr As Long) As Integer

Declare Function SQLDataSources Lib "odbc.dll" (ByVal henv As Long, ByVal fdir As Integer, ByVal szDSN As String, ByVal cbDSNMax As Integer, pcbDSN As Integer, ByVal szDesc As String, ByVal cbDescMax As Inte-

```

ger, pcbDesc As Integer) As Integer
Declare Function SQLFreeEnv Lib "odbc.dll" (ByVal env
As Long) As Integer
Declare Function SQLFreeConnect Lib "odbc.dll" (ByVal
dbc As Long) As Integer
Declare Function SQLFreeStmt Lib "odbc.dll" (ByVal stmt
As Long, ByVal fOption As Integer) As Integer
Declare Function SQLDisconnect Lib "odbc.dll" (ByVal dbc
As Long) As Integer
Declare Function SQLGetData Lib "odbc.dll" (ByVal stmt
As Long, ByVal icol As Integer, ByVal fCType As Integer,
ByVal rgbValue As String, ByVal cbValueMax As Long,
pcbValue As Long) As Integer
Declare Function SQLFetch Lib "odbc.dll" (ByVal stmt As
Long) As Integer
Declare Function SQLNumResultCols Lib "odbc.dll" (By-
Val stmt As Long, pccol As Integer) As Integer

```

'ODBC 系统初始化

```
rc = SQLAllocEnv(henv)
```

'获取数据源名称 SourceName

```
Call GetDataSources(List1)
```

'与数据源建立连接

```
rc = SQLAllocConnect(henv, hdbc)
```

```
rc = SQLConnect(hdbc, SourceName, -3, "", -3, "",
-3)
```

'准备 SQL 语句

```
rc = SQLAllocStmt(hdbc, hstmt)
```

'获取表名称 TableName

'执行 SQL 语句

```
szSqlStr = "select *. from " & TableName
```

```
rc = SQLExecDirect(hstmt, szSqlStr, Len(szSqlStr))
```

```
strEmpName = String$(20, 0)
```

'检索结果集,并将数据写入剪贴板

```
rc = SQLNumResultCols(hstmt, intCols)
```

```
clipboard.Clear
```

```
Do While (1)
```

```
rc = SQLFetch(hstmt)
```

```
If rc = 0 Or rc = 1 Then
```

```
For i = 1 To intCols
```

```
retcode = SQLGetData(hstmt, i, 1, ByVal strEmp-
Name, 20, lngNameLen)
```

```
msg = Space(5) & strEmpName
```

```
clipboard.SetText clipboard.GetText() & msg
```

```
strEmpName = String$(20, 0)
```

```
Next i
```

```
clipboard.SetText clipboard.GetText() & Chr(13)
```

```
Else
```

```
Exit Do
```

```
End If
```

```
Loop
```

'释放数据源

```
rc = SQLFreeStmt(hstmt, 1)
```

```
rc = SQLDisconnect(hdbc)
```

```
rc = SQLFreeConnect(hdbc)
```

```
rc = SQLFreeEnv(henv)
```

4. 结束语

VB 中用 ODBC 来编程的方法有多种,然而利用 ODBC API 直接编程的方法由于其编码的复杂性及要求程序员具有编制 API 调用的经验,因而例子并不多见,本文给出的利用 ODBC API 直接编程技术将数据库数据读到 Windows 系统下的剪贴板上的程序代码,在解决了一般应用程序不能共享数据库的问题的同时,对于如何利用 ODBC API 直接编程也具有借鉴意义。

参考文献

- [1] Microsoft 公司, Microsoft ODBC 2.0 程序员参考手册和 SDK 指南, 电子工业出版社, 1996.9.
- [2] JAMES I. CONGER, Microsoft Windows API 大全, 北京科海培训中心, 1996.2.

(来稿时间:1997年8月)