

在 VB 中 Oracle 大字段存取的实现

钱晓明 王宁生 吴含前 (南京航空航天大学 CIMS 工程研究中心)

摘要: 本文介绍了在 VB 中通过 ADO 实现对 Oracle 数据库大字段的存取, 并着重探讨了如何利用 RecordSet 对象的二进制读写函数 AppendChunk, GetChunk 实现文件与数据库之间的存取。

一、问题的提出

最近开发一个文档管理软件, 起先将所有文档都上载到 FTP 站点, 但文件放在 FTP 站点上, 有以下缺陷:

- (1) FTP 站点的安全性难以保证。
- (2) 文件存放在 FTP 站点上, 难以完成有效的备份。
- (3) 由于文件不能重名, 因而版本管理非常麻烦。

如果能将文件以二进制的形式存放在 Oracle 数据库中, 则以上的问题就可以迎刃而解。在 Oracle 数据库中存放文件有以下优点:

- (1) Oracle 数据库的安全性是比 FTP 站点要可靠。
- (2) Oracle 数据库有完整的自动备份系统, 可以保持数据的完整和有效性。
- (3) 可以有效的实现文件的版本管理与控制。

因此, 用 Oracle 数据库存放文件具有非常现实的意义, 尤其在文档管理应用方面, 为此开发了一个在 VB 下实现 Oracle 数据库大字段存储的程序, 适用于 txt, gif, dwg, exe, doc 等各种文件。

二、实现原理

在 Oracle 7.3 版本中有一个 Long Raw 字段, 在 Oracle 8 中叫 BLOB, 是用来存储二进制的, 最大可以 2G。定义数据库连接对象如下:

```
Set conn = New ADODB.Connection
```

```
Set rec = New ADODB.Recordset
conn.Open "pdm", "user", "psw" ' "pdm",
"user", "psw" 分别为 ODBC 的 DSN, Oracle 数据库的用户名和口令。
```

```
rec.Open "test", conn, adOpenKeyset,
adLockOptimistic, adCmdTable
```

“test”为表名, 由 FileName 和 FileContent 两个字组成, FileName 为 Varchar2(50), FileContent 为 Long Raw, 即大字段。

ADO 中 Recordset 对象有 AppendChunk 和 GetChunk 两个方法, 还有一个 ActualSize 属性。AppendChunk 用来向数据库写二进制, GetChunk 用来从数据库读二进制。实现存放文件主要用这两个方法。ActualSize 属性主要用来得到大字段中的字节数。

在实现过程中可以开一个缓冲区, 用来暂时存放分块读入/出的二进制数据, 每次读的大小可自己控制。其定义如下:

```
dim Chunk() as long '定义缓冲区
```

```
const ChunkSize = 16384 '定义每次读写二进制的大小
```

在 VB 中还提供了直接对文件读写的方法, 如 Get, Put, Seek, 还有一个 LOF() 函数, 用来获得文件的大小, 与前面提到的 ActualSize 相对应。首先利用 Open

以二进制形式(包括文本文件)打开一个文件,然后用Get方法,可以把每次从文件读取的二进制放在缓冲区Chunk()中,最后用AppendChunk将缓冲区Chunk()中的数据读入Long Raw字段中。反之,用Put方法可以把Long Raw字段中的数据读入文件。其定义如下:

```
dim Filename as string '文件名变量
dim IFileSize as long '获得文件大小的变量
dim iFileNumber, as Integer '文件号,是某个文件的标志
dim IRemainder as long
dim iChunkCount as integer '文件的块数
```

三、实现程序

在窗体Form1上放置三个命令按钮和一个CommonDialog1控件。

```
在Command1_Click事件中'打开一个文件
CommonDialog 1.Filter = "All Files|*.*"
CommonDialog 1.ShowOpen
FileName = CommonDialog 1..FileName
For l = Len(FileName) To 1 Step -1 '以下代码用来把文件的路径去掉,只剩文件名
If Mid(FileName, l, 1) = "\" Or Mid(FileName, l, 1) = "/" Then
Exit For
End If
Next
FileName = Right(FileName, Len(FileName) - 1)
```

```
在Command2_Click事件中'将文件存到读数据
iFileNumber=FreeFile '获得空闲文件号
Open Filename For Binary Access Read As iFileNumber
'以二进制形式打开由CommDialog1控件获得的文件。
Screen.MousePointer = 11 '显示沙漏光标
IFileSize=LOF(iFileNumber)'获得文件大小
iChunkCount= IFileSize\ ChunkSize '获得文件块数
IRemainder= IfileSize Mod ChunkSize '取整块后剩余的数据块
Redim Chunk(ChunkSize)'定义缓冲区大小为ChunkSize
For i = 1 To iChunkCount
Get iFileNumber, , Chunk()'从文件中读数据
```

```
rec(1).AppendChunk Chunk()'写到数据库中
Next i
ReDim Chunk(IRemainder)'处理剩余的数据块
Get iFileNumber, , Chunk()
rec(1).AppendChunk Chunk()
rec(0)=FileName
Close iFileNumber '关闭文件
Screen.MousePointer = 0 '光标恢复
在Command3_Click事件中'将文件从数据库读出
iFileNumber=FreeFile '获得空闲文件号
Open App.Path & "\" & rec(0) For Binary Access
Write As iFileNumber
'在当前目录下创建文件
iChunkCount = rec(1). ActualSize \ ChunkSize
IRemainder = rec(1). Actual Size Mod ChunkSize
For i = 1 To iChunkCount
ReDim Chunk(ChunkSize)
Chunk() = rec(1).GetChunk(ChunkSize)
Put iFileNumber, , Chunk()
Next i
ReDim Chunk(IRemainder)
Chunk() = rec(1).GetChunk(IRemainder)
Put iFileNumber, , Chunk()
Close iFileNumber
```

四、结论

通过Recordset对象的Appendchunk和Getchunk方法,顺利实现了对Oracle数据库大字段的存取,保证了文件的安全性,完整性和可靠性,同时也简化了版本管理,因而功效明显。对于一些应用需要浏览文件内容,程序中可以结合VB6.0提供的OLE控件来实现,具体代码实现此处从略。

上述程序在Visual Basic 6.0, Windows NT 4.0,Oracle 7.3中实现。■

参考文献

- 1 MSDN 在线帮助
- 2 <<计算机世界>>网站之“实用技巧”
- 3 <<Visual Basic 6.0 开发宝典>> 机械工业出版社