

WWW 数据库访问技术研究

昆明 云南大学信息学院 张国生

摘要: 本文对 ADO 的各个对象在 WWW 上访问数据库的技术进行了深入细致地研究,并结合开发实例阐述各个对象在远程数据库访问中的应用。

关键词: ADO 数据库 对象 属性

概述

Internet 浪潮席卷了全世界,人们可以轻松地坐在计算机前通过电脑浏览各种各样的信息,最初人们通过浏览器所看到的 WWW 服务器上的页面信息是静态的,如果需要变更就只能调用另外的页面,这样极大地占用了 Internet 网络的带宽,影响浏览信息的速度。后来采用 CGI (Common Gateway Interface) 技术产生动态页面,但开发较困难,Microsoft 公司的 ASP (Active Server Pages) 技术的出现,极大地简化了开发动态 Web 页面的难度,使人们通过 Web 页面很容易地访问各种类型的数据库。

ADO (ActiveX Data Object) 为动态 Web 页面开发者提供了轻松存取 Internet 数据库的崭新技术,并在客户端实现网上即时更新显示的 Web 内容,它将原本烦杂的数据库程序开发变成轻松容易的工作,ADO 是 Microsoft 替 Web 数据库量身订做的开发环境,除具有传统数据库开发环境的优点外,还增加了一些更为先进的功能;另外,ADO 几乎兼容各种数据库系统,无论是 Microsoft Access、FoxPro 甚至 Microsoft SQL Server、Oracle、Informix、Sybase 等,ADO 都提供相同的处理界面供程序设计人员使用。

利用 ADO 开发数据库应用程序的优点有:

- (1) 支持客户机/服务器结构(Client/Server)与 Web 系统开发设计;
- (2) 内部具有多个相互独立的对象模型;
- (3) 支持分批修改数据库内容;
- (4) 支持多种不同的数据控制指标形式;
- (5) 先进的 Recordset 数据高速缓存 (Cache) 管理功能;
- (6) 允许在程序中使用多个 Recordset 对象或多个分批修改区块传送等。

ADO 数据访问对象共包括六个对象: Connection、Error、Recordset、Command、Parameter 和 Field 对象,其对象模型如图 1 所示。ADO 应用程序操作数据的一般步骤为:

- ① 链接到数据源 (Connection 对象);
- ② 指定一个访问数据源的命令 (Command 对象),其中可包括任意的参量或操作 (Parameter 对象);
- ③ 执行命令 (Command、Connection 或 Recordset 对象);
- ④ 如果命令导致表中的数据按照行的形式返回,在一个高速缓冲存储器 (Recordset 对象) 中存储该记录,以便很容易地检查、操作或改变;
- ⑤ 如果合适,在高速缓冲存储器中更新数据 (Recordset 和 Field 对象);
- ⑥ 提供一个通用的手段检查通常在连接或执行命令的时候产生的错误 (Error 对象)。

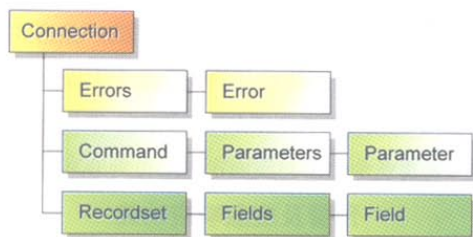


图 1 ADO 对象模型

ADO 各个对象在实际应用中并非都会用到,可根据实际情况合理使用,将会收到事半功倍的良好效果,下面以 ADO 访问 Microsoft SQL Server 7.0 数据库为例讲

述各个对象的使用(这里假设在 Windows NT 服务器控制面板的 ODBC 中所建数据源为:

DSN=sqldsn;UID=sa;PWD=;DATABASE=userbase)

Connection 对象

Connection 对象负责网页应用程序与数据库服务器实际的链接,是唯一与远程沟通的管道,在 ASP 中同时打开多个 Recordset 对象,仅需要一个 Connection 链接点与数据库沟通。

Connection 对象的常用属性有:ConnectionString 属性—指定 Connection 对象的数据库链接信息。

Connection 对象的常用方法有:

(1)Open 方法—打开一个新的 Connection 对象;

(2)Execute 方法—执行数据库操作命令,其格式为:

cn.Execute(CommandText,RecordAffected,Options)

其中 CommandText 参数是一个字符串,可以是某个存储过程名、表名或某个将被执行的 SQL 语句;RecordAffected 参数是执行命令所影响的记录的总数,比如用 Execute 方法执行一个删除或更改记录的 SQL 语句,将可以返回被影响的记录数,但 Select 语句或打开数据表的返回值为 -1,该参数是一个可选项;Options 参数用来指定 CommandText 参数的性质,也是一个可选项,但指定它会提高效率,避免服务器去判断 CommandText 参数的类型,Options 参数见表 1。

(3)Close 方法—关闭 Connection 对象与数据库的链接。

(4)BeginTrans 方法—开始事务处理。

(5)CommitTrans 方法—确认事务处理。

(6)RollbackTrans 方法—放弃事务处理。

表 1 Options 参数值表

名称值	功能
AdCmdText	CommandText 参数中是一个 SQL 命令
AdCmdTable	CommandText 参数中是一个数据表名
AdCmdStoreProc	CommandText 参数中是一个存储过程
AdCmdUnKnown	默认值, CommandText 参数类型不清楚

Connection 对象使用示例:

```
<!--#include virtual="adovbs.inc"-->
<%Response.Buffer=True
On Error Resume Next
...
```

```
Set cn=Server.CreateObject("ADODB.Connection")
cn.Open "DSN=sqldsn;UID=sa;PWD=;
DATABASE=userbase
cOutstockno=trim(Request.Form("txtOutstockno"))
cSerialno=trim(Request.Form("txtSerialno"))
cColor=trim(Request.Form("txtColor"))
cn.BeginTrans
sql="insert outstockqc(outstockno,serialno,color) values
('"&cOutstockno&"','"&cS
erialno&"','"&cColor&"')
cn.Execute(sql)
sql="update stockqc set outmark=1 where
serialno='"&cSerialno&"'
cn.Execute(sql,nCount,adCmdText)
if err.number<>0 then
Response.Clear
cn.RollbackTrans
Response.Write "数据更新失败,请核对后重试。"else
cn.CommitTrans
Response.Write "数据更新成功,祝贺您。"
Response.Write "记录更新数为: " & cstr(nCount)
end if
... %>
```

Recordset 对象

Recordset 对象提供了多种功能强大的对象接口供程序设计人员使用,在处理数据库服务器所返回的数据集合时,存在着一个游标随时指向服务器端正在指向的数据行位置,游标的起始状态在 Recordset 对象的第一个数据行位置。

Recordset 对象的常用属性有:

(1)absolutePage 属性—指定当前的绝对数据页位置;

(2)ActiveConnection 属性—表明与 Command、Connection 对象主动链接;

(3)BOF 属性—为记录集的开头边界;

(4)BookMark 属性—为游标位置的书签值;

(5)CursorType 属性—表示由服务器端数据库取回数据的类型(见表 2);

(6)EOF 属性—为记录集的结尾边界;

(7)LockType 属性—为数据更新锁定模式(见表 3);

(8)MaxRecords 属性—允许取回的最大记录数;

(9)PageCount 属性—为数据页总数;

(10)PageSize 属性—为每个数据页容纳的记录数;

(11)RecordCount 属性—为取回的记录总数;

(12)Source 属性—为执行命令的字符串。

表 2 Recordset 对象光标类型表

名称	值功能
AdOpenForwardOnly	只读且只向前移动游标类型, 其运行速度快, 默认为游标类型
AdOpenKeySet	允许看见其他用户所做的修改, 但不能看见其他增加或删除的记录
AdOpenDynamic	所有数据库操作都会立即反应到各客户端, 其功能最强大
AdOpenStatic	仅是数据的一幅快照, 无法看到其他用户对 Recordset 的任何改变

表 3 Recordset 对象数据更新锁定模式表

名称值	功能
AdLockReadOnly	只读模式, 缺省属性值
AdLockPessimistic	编辑记录时即锁定, 直到记录更新被提交才释放
AdLockOptimistic	在数据提交的瞬间锁定记录, 是最安全的锁定机制
AdLockBatchOptimistic	批量更新锁定

Recordset 对象的常用方法有:

(1)Addnew 方法—加入一个新的记录行;

(2)Close 方法—删除当前记录行;

(3)Move 方法—移动到指定的记录行位置;

(4)MoveNext 方法—后移一个记录行;

(5)MovePrevious 方法—前移一个记录行;

(6)MoveFirst 方法—移到首记录;

(7)MoveLast 方法—移到尾记录;

(8)Open 方法—打开记录集;

(9)Requery 方法—重复数据库查询;

(10)Update 方法—更新当前数据;

(11)UpdateBatch方法—在批量上传模式下更新数据。

Recordset 对象使用示例:

```
<!--#include virtual="adovbs.inc"-->
```

```
<% ...
```

```
if Request.Form("subOpen")=" 打开数据表 " then
```

```
set rs=server.CreateObject("adodb.recordset")
```

```
rs.Source="select nno,nal,uqty from ptnoty where
```

```
uqty>10.00"
```

```
rs.ActiveConnection=" DSN=sqldsn;UID=sa;
```

```
PWD=;DATABASE=userbase "
```

```
rs.Open „adOpenKeyset,adLockPessimistic,adCmdText
```

```
Response.Write rs.RecordCount
```

```
rs.PageSize=8
```

```
rs.AbsolutePage=1
```

```
set session("rsOpen")=rs
```

```
session("pageposition")=1
```

```
end if
```

```
if Request.Form("subLast")=" 上一页 " then
```

```
if session("Pageposition")>1 then
```

```
session("Pageposition")=session("Pageposition")-1
```

```
session("rsOpen").AbsolutePage=session  
("Pageposition")
```

```
session("rsOpen").movefirst
```

```
session("rsOpen").move (session("Pageposi-  
tion")-1)*session("rsOpen").pagesize
```

```
else
```

```
session("rsOpen").movefirst
```

```
session("rsOpen").move (session("Pageposi-  
tion")-1)*session("rsOpen").pagesize
```

```
end if
```

```
end if
```

```
if Request.Form("subNext")=" 下一页 " then
```

```
if session("Pageposition")<session("rsOpen")  
.PageCount then
```

```
session("Pageposition")=session("Pageposition")+1
```

```
session("rsOpen").AbsolutePage=session  
("Pageposition")
```

```
session("rsOpen").movefirst
```

```
session("rsOpen").move
```

```
(session("Pageposition")-1)*session("rsOpen")  
.pagesize
```

```
else
```

```
session("rsOpen").movefirst
```

```
session("rsOpen").move (session("Pageposi-  
tion")-1)*session("rsOpen").pagesize
```

```
end if
```

```
end if
```

```
... %>
```

Recordset Fields 数据集合

Fields是指Recordset对象当前位置记录的字段集合,可直接取得 Recordset 内的字段数据,主要属性有:

- (1)Count 属性—为可以获得的字段数;
- (2)Name 属性—为字段名称。

Fields 集合应用示例 (显示 Recordset 中的内容):

```
<!-- #include virtual="adovbs.inc" -->
<%
...
Set rs=Server.CreateObject("ADODB.Recordset")
rs.Open "select * from cj"," DSN=sqldsn;UID=
sa;PWD=;DATABASE=userbase ", adOpenDynamic,ad-
LockOptimistic, adCmdText

if not rs.bof then
Response.Write("<TABLE Border=2 Size=100>")
for i=0 to rs.Fields.Count-1' 显示表头
Response.Write("<TD>" & rs(i).Name & "</TD>")
next
While not rs.EOF
Response.Write("<TR>")
for i=0 to rs.Fields.Count-1' 显示字段内容
temp=rs(i)
if isnull(temp) then temp=""
Response.Write("<TD>" & CStr(temp) & "</TD>")
next
rs.MoveNext
Wend
Response.Write("</TABLE>")
Else
Response.Write " 没有满足条件的记录存在!
End If
...%>
```

Command 对象

Command对象表示一个可被数据源处理的命令,包括查询和存储过程,使用Command对象可以很容易地利用SQL语句对数据源的数据进行有效地处理。

Command对象的常用属性有:

- (1)ActiveConnection属性—建立与Connection对象的链接关系;
- (2)CommandText 属性—为查询命令;
- (3)CommandType属性—为查询命令类型(见表1)。

Command 对象常用方法:

- (1>CreateParameter 方法—建立一个新的参数对象;
- (2)Execute 方法—对数据库提出数据查询。

Command对象应用示例(建立参数化SQL添加命令):

```
<!-- #include virtual="adovbs.inc" -->
<%
...
Set cm=Server.CreateObject("ADODB.Command")
cm.ActiveConnection=" DSN=sqldsn;UID=sa;
PWD=;DATABASE=userbase "
cm.CommandText="insert prnoty(nno,na1,ty,unit,
note) values(?,?,?,?)"
cm.CommandType=adCmdtext
set bjparam=cm.CreateParameter(adChar,,len("
cx1001"),"cx1001")
cm.Parameters.Append objparam
set objparam=cm.CreateParameter(adChar,,len("
前刹车_鼓"),"前刹车鼓")
cm.Parameters.Append objparam
set objparam=cm.CreateParameter(adChar,,len(
("23.46"),"23.46")
cm.Parameters.Append objparam
set objparam=cm.CreateParameter(adChar,,len("
个"),"个")
cm.Parameters.Append objparam
set objparam=cm.CreateParameter(adChar,,len(
("1111"),"1111")
cm.Parameters.Append objparam
set rs=cm.Execute
... %>
```

ADO所涵盖的内容十分丰富,其应用多种多样,本文对ADO的各个对象进行了深入细致地应用研究,结合实际开发给出了相应的示例,相信大家对 Connection、Recordset 和 Command 对象将会有更深刻地理解,并结合实际灵活运用,一定能体会到 Microsoft ADO 技术给开发者带来的巨大便利。■

参考文献

- 1 赵昊彤,杨珏等编著《VBScript 制作实例》,人民邮电出版社
- 2 李世杰编著《Active Server Pages(ASP) 2.0 网页设计手册》,清华大学出版社