

# 基于 OLE DB 的 ADO 数据访问技术\*

王 平,覃理矜

(重庆邮电学院,重庆 400065)

**摘 要:**介绍了基于 OLE DB 的 ADO 数据访问技术,并详细介绍了 OLE DB/ADO 的概念及对象模型和工作原理,还举例说明了 ADO 数据访问技术的一般过程和实现方法。

**关键词:**OLE DB;数据访问;对象;数据访问接口

**中图分类号:**TN919.1      **文献标识码:**A

## ADO Data Access Technology with Basis of OLE DB

WANG Ping, QIN Li-jin

*(Institute of Automation, Chongqing University of  
Posts and Communications, Chongqing, 400065, China)*

**Abstract:** Based on OLE DB, this paper introduces the technology of ADO Data Access, introduces the concept, object model and working mechanism of OLE DB/ADO in detail, and then gives an example to depict the general process and realizing method of using the technology of ADO Data Access.

**Key words:** OLE DB; data access; object; data access interface

## 0 引 言

ADO(ActiveX Data Objects)是由微软公司推出的以 ActiveX 技术为基础的数据存取方法,是一组基于 OLE DB 数据的高级自动化(Automation)应用层接口。它继承了 DAO/RDO,功能上与 RDO 更相似,两者之间存在一定的映射关系。ADO 扩展了 DAO 和 RDO 所使用的对象模型,包含较少的对象、更多的属性、方法(和参数)和事件模型。ADO 支持建立基于 COM 的桌面和客户/服务器模式与基于 Web 的应用程序,具有远程数据服务 RDS(Remote Data Service)的特性,通过 RDS 能够在一次往返中将服务器端的数据传送到客户端的应用程序

或 Web 页面中,并在客户端对数据进行处理后,立即更新服务器端的数据。ADO 是为微软公司新的数据访问接口 OLE DB 而设计的,通过它可以方便快速地通过统一的数据接口 API(Application Programming Interface),对关系或非关系数据库、电子邮件和文件系统、文本和图形、自定义和业务对象等进行访问和存取,形成一个高性能的轻量接口,是数据库访问技术的一个主要发展方向。

## 1 OLE DB 的概念和对象模型

微软提出的 ODBC(开放数据库互联技术)采用分层结构,通过提供一标准的基于 SQL 查询的 API 函数,提供了一致的数据库访问接口,但是它的对象

\* 收稿日期:2000-08-22

作者简介:王平(1963-),男,重庆市黔江人,博士,副教授,从事智能化邮政技术、网络控制技术、计算机应用开发方面的研究;覃理矜,硕士研究生,从事系统集成、地理信息系统开发和数据库技术研究。

是基于 SQL 的数据源,对于不同地方、不同格式的数据源包括操作系统中的文件、顺序索引文件、桌面数据库、电子邮件、目录服务、多媒体数据、空间数据等却显得无能为力,此时微软的 UDA(一致数据访问)则能很好地解决这个问题。其层次结构如图 1 所示。

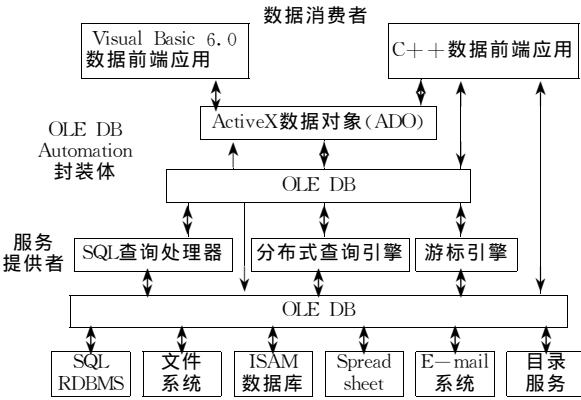


图1 Universal Data Access 结构

从图 1 中可看出,UDA 包括两层软件接口:OLE DB 和 ADO,分别对应于不同层次的应用开发。OLE DB 是 UDA 的核心,在系统建立了数据访问的一组标准 COM 接口。OLE DB 标准的具体实施是一组符合 COM 标准、基于对象的 C++ API 函数。使用 OLE DB API,可以编写能够访问符合 OLE DB 标准的任何数据源的应用程序,也可以编写针对某种特定数据存储的查询处理程序和游标引擎,因此,OLE DB 标准实际上是规定了数据消费者和数据提供者之间的一种应用层协议。OLE DB 的对象模型如图 2 所示。

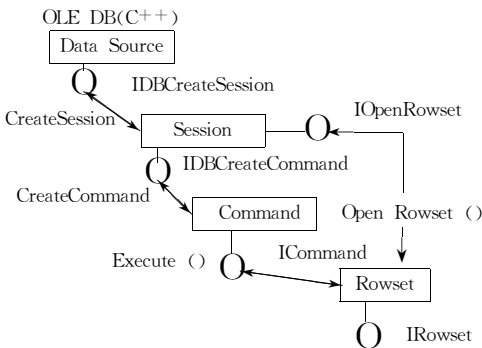


图2 OLE DB 的对象模型

从图 2 中可看出,OLE DB 规范一组层次型数据对象间的接口,这些 COM 对象主要是:

① Data Source 对象 它提供了一组函数用以标识特定的 OLE DB 数据提供者,负责管理用户权

限、建立与数据源的连接等初始化操作;

② Session 对象(会话) 在数据源连接的基础上建立会话对象,提供创建 Rowset 和隔离事务的环境;

③ Command 对象(命令) 数据消费者对各种数据库进行操作的处理查询的集合,这些查询通常是 SQL 语句或存储过程名称;

④ Rowset 对象(行集) 可由 Session 对象直接产生或由 Command 对象的执行而产生,通过 Irowset 接口把数据传给数据消费者,它是应用程序主要的操作对象。

## 2 ADO 工作原理

尽管 OLE DB 这个系统级的数据库编程接口提供了一系列有关数据库管理系统服务的统一的 COM 接口,能很好地解决对异构的关联和非关联信息的存取,但是其接口较为复杂,而且其接口 API 函数是 C++ API,只能提供 C++ 语言调用接口,而不能直接用于其它高级编程语言。于是,ADO 应运而生,它可适用于多种编程语言,自适应访问任何 OLE DB 的数据源,而且允许 C++ 程序员直接访问底层 OLE DB,ADO 的对象模型如图 3 所示。

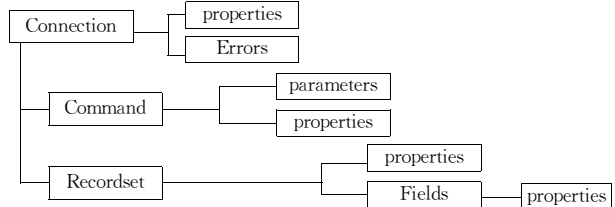


图3 ADO对象模型

由图 3 可看出,ADO 对象模型包括 7 个对象,但主体对象只有 3 个:Connection,Command 和 Recordset,可被独立创建和释放。而 Parameters 对象虽然可独立于 Command 对象创建,但不能独立于一个 Command 对象使用。Field,Error 和 Property 对象则不能独立创建,只能依附于它们的父对象而存在。

① 数据连接(Connection)。Connection 建立一个数据交换环境,应用程序通过 Connection 访问数据。应用程序直接或通过诸如 IIS(Internet Information Server)这样的系统间接访问数据源。Connec-

tion 对象用于规定特定的数据提供者和附加参数。由 Connection 发生的一系列数据存取操作形成一个事务处理过程, ADO 确保事务过程成功操作的结果传给数据源, 如果其中某个操作失败或用户取消事务过程, 数据源中的数据就会维持原先的状态。对象模型并不明确地具体表示一个事务过程, 而通过一套 Connection 对象方法实现。

② 命令(Command)。Command 通过已建立的 Connection 发布。典型的 Command 有增加、删除和更新、恢复表中的记录等。

③ 记录集(Recordset)。Recordset 存储查询命令返回的结果, 是检查和修改记录的主要手段, 在 Recordset 对象中可以检查记录, 在记录中浏览, 浏览记录的次序, 增加修改或删除记录、更新记录已变化的数据源等。

④ 错误(Error)。任何涉及 ADO 对象的操作都会生成一个或多个提供者错误, 此时一个或多个 Error 对象被放到 Connection 对象的错误集合中, 新的 Error 将替换先前的 Error 对象。

⑤ 属性(Property)。ADO 有内置和动态 2 种属性, 前者是 ADO 固有的, 始终生效; 后者是数据提供者添加到 ADO 对象的 Property 集合中, 仅当使用该数据使用者时才有效。

⑥ 字段(Field)。记录集的每条记录由一个或多个字段组成, 每个字段包括自己的名称、类型和数据值, 可以通过修改记录集中的字段对象值, 最终传递到数据源中。

### 3 ADO/OLE DB 的基本应用方法

假设所要设计的应用程序是从多个不同格式和地方的数据源(如 ISAM 数据仓库、Oracle 关系数据库和 SQL Server)等获取数据, 则处理方法如下:

首先, 使用 OLE DB 数据提供程序与每一个数据库连接, 建立统一的接口, 并使用 ADO 从 OLE DB 获取数据; 其次, 向应用加入查询处理器, 该处理器是 OLE DB 的一个服务组件。当应用与一个支持查询功能的数据提供程序对话时, 如一个关系数据库, 该应用可直接通过 ADO 访问 SQL 数据。但当应用要与 ISAM 数据对话时, 不需要 SQL 查询处

理器就可以通过 ADO 与其对话。作为一般的策略, 我们可向应用中插入多种查询处理器, 以支持不同的查询语法和功能。

(1) 基本的 ADO 原型访问顺序:

- 连接数据源;
- 指定数据源的存取命令;
- 激活命令;
- 将返回的数据存入表的记录缓冲区中便于检查修改;
- 更新数据的变化。

以上步骤都会用到, 但不是必须的, ADO 可以灵活地使用原型中的一部分进行工作。

可以直接从一个文件中把数据存到表的记录缓冲区, 利用 ADO 方法检查数据。

(2) 采用 VB 6.0 创建一个数据访问应用的基本步骤

#### ① 连接数据源

使用 ADO 的 Connection. Open 或 Recordset. Open 连接, 其语法为:

```
Connection. Open ConnectionString, UserID, Password, OpenOptions
```

```
Recordset. Open Source, ActiveConnection, CursorType, LockType, Options
```

其中 ActiveConnection 或 Connection 中的 ConnectionString 的属性一般设置为:

```
"PROVIDER=MSDataShape;Data PROVIDER=MS-DASQL;DSN=Postrans;UID=user;PWD=password"
```

这样, 数据源的名称及位置、用户 ID 和密码都被设定了。另外还可使用 ADO Data Control 控件或采用数据环境设计器(DED)进行快速高效的连接。

#### ② 创建 SQL 查询命令对象

典型的命令用 SQL 语句组成, 查询的结果包含所请求内容的记录集。它的使用格式通常是, 先创建对象, 再使用其属性和方法, 具体如下:

```
Command cmd = New ADO.DB.Command
```

```
cmd.CommandText = "Select * from PostalCenter"
```

上面的 SQL 语句的内容是固定的, 但可以通过创建“参数化”命令, 即使用占位符“?”来指定参数化命令字符串, 这样, 当命令执行时占位符将被参数所取代。如下面这个例子可查询四川省的中心局及其代号:

```
cmd.CommandText = "Select * from PostalCenter
where province=?"
cmd.Prepared=True
Parameter prm = New ADODB.Parameter
Prm.Type=adVarChar
Prm.Direction= adInput
Prm.size=40
Prm.Value="四川"
cmd.Parameters.Append prm
```

### ③ 执行命令

在创建命令之后,紧接着就是执行它。命令的结果是从数据源返回所需要的信息数据,如果命令按行返回数据,则把行存储在存储对象 Recordset 中。随后对 Recordset 对象进行操作,便达到对查询的数据的操作。

通过 ADO 查询命令返回 Recordset 的方法有 3 种:Connection.Execute、Command.Execute 以及 Recordset.Open。以下是它们的 VB 语法:

```
connection.Execute(CommandText,RecordsAffected,
Options)
command.Execute(RecordsAffected,Parameters,Options)
recordset.Open(Source,ActiveConnection,CursorType,LockType,Options)。
```

### ④ 处理数据

执行命令之后,就可以对其进行处理,具体的数据处理操作有:

- 可把 Recordset 看作行数组,在任意给定的时间可进行测试和操作的行“当前行”,在 Recordset 中的位置为“当前行位置”;
- 使用 Move 或 Find 方法或设置 BookMark 属性来对记录集进行移动或定位;
- 可使用 Filter 属性筛选记录集,用 Sort 属性进行排序;
- 记录集有 Fields 集合,可使用其中 Field(字段)对象的 Value 属性为字段赋值或检索数据。

### ⑤ 更新数据源

上面的数据处理只是在 Recordset 对象中进行了其中数据的定位、检查和编辑。为了能真正把数据源中的数据值改变,则要使用存储对象中的变更对数据源进行更新,也可选择在事务中嵌入更新。

ADO 有 2 种途径用于添加、删除和修改数据源的行。

其一是更改不是立即对 Recordset 进行,而是对“复制缓冲区”进行,不想更改时,复制缓冲区中的改动将被放弃;如果想保留更改,复制缓冲区中的改动将应用于 Recordset。前者非常适合于数据操作出现异常,如中途断电、网络故障等,它可确保几个数据操作的同步。

其二是将更改立即传给数据源。这种“立即模式”下,用户确定数据处理完成时直接把单个记录,或成批把一系列记录的更新传送给数据源,通过 CursorLocation 属性和 LockType 属性进行管理。当前记录的移动或 Update 方法保存数据修改到记录集,必须用 UpdateBatch 方法将更新传送给数据源,它与打开记录集的方式有关。

### ⑥ 结束更新数据源

在进行数据源的更新时,并不是每次都能成功地完成任务,特别是在事务之后,一旦数据源接受或拒绝对数据源的所做的更改,都将结束对数据源的更新操作。为此,ADO 结束更新数据源的过程分两部分进行:

其一,如果更新过程中没有错误发生,则数据源接受事务对数据的更新,随后结束更新数据源的过程;

其二,如果更新数据源时发生了错误,则该错误将会被错误处理例程捕获。随之可使用 adFilterConflictingRecords 常数对 Recordset 进行筛选,将冲突行显示出来。为了纠正错误需要把有关字段的信息打印出来,然后回卷事务,放弃对数据源的更新操作,由此结束更新数据源过程。

```
.....
conn.CommitTrans
.....
On Error
Rs.Filter=adFilterConflictingRecords
Rs.MoveFirst
Do While Not Rs.EOF
    Debug.print "局名: " & Rs.Fields("局名")
    Rs.MoveNext
Loop
Conn.Rollback
```

(下转 94 页)

Resume Next

.....

以上几个步骤是 ADO/OLE DB 数据访问的基本使用方法,对于不同的程序设计语言,都有相似的过程。此外,借助 VB 6.0 的 ADO 数据访问控件进行快速连接,之后把其它 ADO 数据绑定控件 DataList,DataCombo 和 MSHFlexGrid 绑定在该数据控件上,对数据源进行操作。

## 4 结束语

作为一致的数据访问接口,由于 OLE DB/ADO 对象模型的简单性、可扩充性、数据访问的一致性以及易用性,它将逐步代替老的 DAO 和 RDO 数据访问接口,成为新的一种标准的数据访问接口,它的应

用将会越来越广。

## 参 考 文 献

- [1] OLE DB/ADO: Making Universal Data Access a Reality, Microsoft Corporation [S]. 1998.
- [2] MSDN, Microsoft Coporation [S]. 1998.
- [3] 莫卫东. Visual Basic 6.0 高级编程技巧—ADO 数据访问篇 [M]. 西安市: 西安交通大学出版社, 1999.
- [4] ROGER JENNINGS. Visual Basic 6.0 数据库开发人员指南 [M]. 北京: 机械工业出版社, 2000.