活动数据对象 ADO 及在 Internet/Intranet 环境下的具体使用方法

杨莉萍　(山东财政学院信息系　250014)

摘要:ADO 是 Microsoft 推出的通用数据存取(UDA)中面向高层编程的一个组件,它独立于开发工具和编程语言,为各种数据源操作提供了一致的、高效执行的数据存储方法。本文对 ADO 进行了简要概述,介绍了 ADO 的主要性能及内部对象模型,最后给出了它在 Internet/Intranet 开发环境下的具体使用方法。

关键词: ADO　通用数据存储 UDA　Internet/Intranet　ASP

一、引言

企业级应用面临各种类型的数据源处理,其中有关系型和非关系型。以前由于缺乏对各种数据源进行存取的统一界面,所选择的数据存取方案以及应用编程接口很可能不适应新的环境,常常让开发人员陷入困境。正是在这样一种情况下,Microsoft 推出了它的通用数据存取(UDA, Universal Data Access)解决方案。

UDA 可以看作是应用程序和各种数据源之间的一个中间层(如图 1),UDA 一方面可以对各种类型的数据源进行高效存取,同时又提供了一个独立于编程语言、开发工具的统一编程界面,这样企业可以选用他们所熟悉的、易于使用的开发工具,把分散的、完全不同的各类数据源集成起来,创建容易维护、功能强大的应用程序。

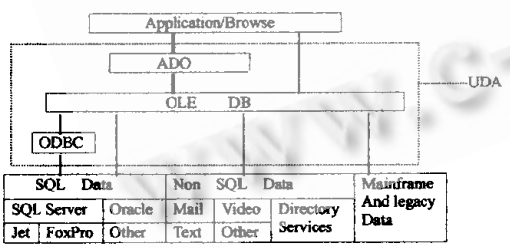

图 1　UDA 的体系结构

UDA 实际上是由 Microsoft 的一些数据存取组件组成,其中最重要的是 OLE DB 和 ADO。OLE DB 提供了对底层各种数据源的存取界面,面向数据提供者;ADO 则向应用程序提供了统一的数据存取界面,面向应用程序开发人员。

对于配制了相应 OLE 数据提供者(Data Provider)的各种数据源,均可以通过 ADO 编写应用来存取和操纵其中的数据。因此对应用开发人员来说,UDA 就是 ADO,只需使用 ADO 就可以统一地存取各种数据源。

二、ADO 支持的主要性能

ADO(ActiveX Data Object)是基于 OLE DB 上的高层编程界面,它提供了一致的、高执行效率的数据存取方法。它具有高速、易于使用、内存开销低、占用磁盘少等优点。

针对建立客户/服务器和 Web 应用,ADO 支持下列主要性能:

1. 独立创建对象。与 DAO(Data Access Object)和 RDO(Remote Data Object)不同,你不必再通过层次体系引导来创建对象,因为大多数 ADO 对象可以被独立创建,允许你只创建和使用你需要的对象,从而产生更少的 ADO 对象和工作集。

2. 成批修改,在本地缓存变动数据,用一次改变将结果集返写回服务器,增进了系统性能。

3. 支持对二进制大数据量的数据存取。

4. 支持带有输入输出参数及返回值的存储过程。

5. 支持多种游标类型,使用灵活。

6. 支持对返回行数及其他查询目标进行限制。

7. 支持存储过程和批处理语句返回的多个结果集。

8. 为高效 Web 服务器应用提供自由线程(free-thread)对象。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

三、ADO 内部的对象模型

使用 ADO 进行数据存取，是通过调用其内部对象提供的方法和属性来完成的。因此使用 ADO，首先要了解其内部提供的各个对象。图 2 就是 ADO 的对象模型：

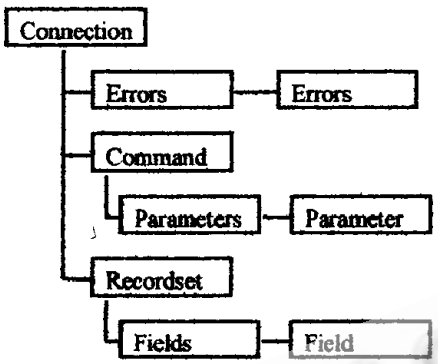

图 2

此外，每个 Connection、Command、Recordset 和 Field 对象还都具有一个 Prooerties 集：

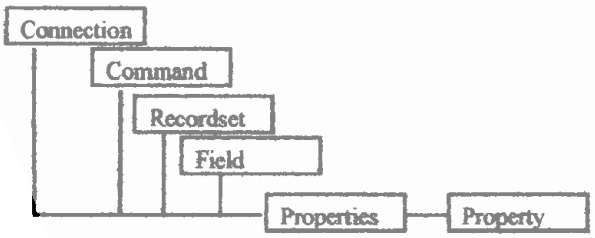

图 3

总共有 7 个对象：Connection、Command、Recordset、Field、Patameter、Property、Error，4 个集合：Fields、Patameters、Propertys、Errors。它们之间关系见图 4。

其中对象 Connection、Command、Recordset 为 ADO 的主体对象，Connection 对象主要用来建立与数据库服务器的连接；Command 主要用来执行对数据库的操作命令，象查询、数据修改等；而 Recordset 对象则用于观察和操作返回的数据集。至于 Command 对象使用的命令语言，依赖于数据库的基础提供者（underlying provider），如果是关系型数据库，命令语言一般为 SQL。还有 Command 对象并非必须使用，对一些非关系型的数据查询就不需要它。

此外，Property 对象及对象集是用来描述 Connection、Command、Recordset 等对象在进行数据操作时相应的操作环境属性，因而隶属于上述对象；而 Parameter 对象和对象集是用于描述 Command 对象在执行带参数的数据操纵命令时需要的一些参数，它隶属于 Command 对象；Error 对象及对象集用于描述在对数据库连接或操纵发生错误时产生的错误信息，它只能通过 Connection 对象检索到，隶属于 Connection 对象；Field 对象和对象集则用于描述 Recordset 对象返回的数据字段，可借助于 Field 对象来访问 Recordset 对象中返回的数据，Field 对象隶属于 Recordset 对象。

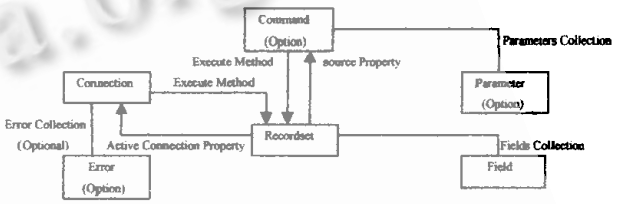

图 4

以上对象除 Error、Field、Property 外均可被创建，ADO 为每个对象均提供了丰富的属性和方法供用户调用，借助于它们完成对各种数据源的处理。

四、ADO 在 Internet/Intranet 环境下的具体使用方法

不论是 Internet 还是 Intranet，它们包含了各类信息源的复杂组合，所有信息都要能通过浏览器浏览，以前在 Internet/Intranet 上开发数据库应用比较困难，现在通过 ADO 可以方便地进行开发。下面就以 Internet/Intranet 开发环境为例，介绍 ADO 的具体使用方法。

在 Internet/Intranet 上通过 ADO 进行数据存取一般采用下述方式：将浏览器对数据的存取要求经过网络传到相应站点的 IIS（Internet Information Server）服务器，然后由 IIS 服务器使用 ADO 来完成对各类数据源的数据存取。这实际上就是 Microsoft 大力推广的活动服务页面技术（ASP，ActiveX Server Page）。所谓 ASP 技术，简单地说就是对 Microsoft 的 IIS 在服务器端的一种扩展，它允许基于 Web 的应用程序的开发人员编写服务器端的脚本以扩展基于 Web 的应用程序的能力。服务器端的脚本可以用 VB Script 来写。这些用高级语言编写的

© 中国科学院软件研究所 http://www.c-s-a.org.cn

脚本在服务器端运行,所以它们不依赖于任何客户端的操作系统或浏览器。

采用这种方式使用 ADO,开发 Internet/Intranet 数据库应用是在服务器端进行的,具体地是通过建立调用 ADO 对象的 ASP 页面来实现。在 ASP 页面中调用 ADO 对象进行数据处理的一般步骤为:

·建立 ADO 对象实例。

·打开与相应数据库的连接。

·调用 ADO 对象实例的方法和属性,完成相应的数据操作,建立结果集、对结果集中的数据进行操作。

·关闭连接。

在这里建立 ADO 对象实例有两种方式:

·一种是使用 ASP 内嵌对象 Server 的 CreateObject 方法,在 VB Script 中的具体格式为:

<% SET 对象实例名 = Server.CreateObject("相应 ADO 对象的注册名") %>

其中 ADO 对象 Connection、Command、Recordset、Field、Patameter、Property、Error 的注册名(PROGID)分别为 ADODB.Connection、ADODB.Command、ADODB.Recordset、ADODB.Field、ADODB.Patameter、ADODB.Property、ADODB.Error。

·一种是使用 HTML 语句的<Object>标志,格式为:

<Object RUNAT = Server ID = 对象实例名 PROGID = "相应 ADO 对象的注册名">

或 <Object RUNAT = Server ID = 对象实例名 CLSID = "相应 ADO 对象的注册号">

下面即为具体应用实例:

该例为一个产品浏览页面,其中的产品条目就是从数据库中获取的:

```
<html>
<head>
<title>产品浏览页面</title>
</head>
<body>
<% rem 实例化一个 Recordset 对象
    set rds = Server.CreateObject("ADODB.Recordset")
    rem 连接相应的数据库,同时执行检索命令,建立结果集,打开对应游标
    rds.open "select mc,gg,jj,jg from cplb ", "UID=sa;DSN=Mydate" %>
<% rem 将结果集以表格的形式列出 %>
<% rem 通过 HTML 语言输出标题栏 %>
<table>
  <tr>
    <td>产品名称</td>
    <td>规格</td>
    <td>简介</td>
    <td>价格</td>
  </tr>
<% rem 逐行输出结果集中的内容
  do while not rds.eof %>
    <tr>
      <td><% = rds.fields(0) %>
      </td>
      <td><% = rds.fields(1) %>
      </td>
      <td><% = rds.fields(2) %>
      </td>
      <td><% = rds.fields(3) %>
      </td>
  </tr>
<% rem 游标指针下移
    rds.movenext %>
<% loop %>
<% rem 关闭连接
    rds.close %>
</table>
</body>
</html>
```

上例只是一个简单的查询,一个查询命令即可完成。而实际上有的查询要复杂得多,不仅要涉及多个数据表、参数,还要使用多个数据库操作命令才能完成,在这种情况下,为了提高系统性能,最好使用存储过程,下面就是调用存储过程来提取查询结果的一段实例:

```
<% rem 将上一页提交的参数通过 ASP 的内置对象 Request 提取出来分别赋给变量 date1、date2
    date1 = Request.Form("t1")
    date2 = Request.Form("t2")
    rem 实例化一个 Command 对象
    set cmd = Server.CreateObject("ADODB.command")
    rem 建立它与相应数据库的连接
    cmd.activeconnection = "UID=sa;DSN=Mydata"
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn

```
rem 将要执行的命令,即调用存储过程 aa,通过 commandtext 属性告诉它
cmd.commandtext = "{call aa(?,?)}"
rem 由于命令中涉及两个输入参数,建立两个参数的描述,两个参数均为日期时间型
cmd.parameters.append cmd.createparameter("dt1",135)
cmd.parameters.append cmd.createparameter("dt2",135)
rem 为两个参数赋值
cmd("dt1") = date1
cmd("dt2") = date2
rem 执行命令同时将结果集赋给一个 Recordset 对象实例
set rds = cmd.execute %>
```

五、小结

ADO 不仅提供了一致的、独立于工具和编程语言的高级数据界面,统一了 Client/Server、Web 和各种其他的数据存储和读取技术,支持用户以统一的方式结构化和非结构化进行存储,而且效率高,易于使用。对于应用程序开发人员来说,ADO 将成为他们存取各类格式信息的强有力工具。采用 ADO 开发新的应用程序,会大大降低应用程序开发成本,提高应用程序的可维护性。

(来稿时间:1998 年 9 月)

© 中国科学院软件研究所 http://www.c-s-a.org.cn