



SOM 脚本对象模型对数据库的访问

黄金生 吴进 (南京军区空军网络中心 210018)

摘要

本文描绘了目前主要的几种基于web的数据库访问技术基本原理、功能、编程方式,以及相互联系与不同。着重阐述了SOM脚本对象模型和DTC设计时间控件的特点和在visual interdev中的使用,对每一种数据访问方式给予简单的举例说明,以利于读者的理解和深入学习。

关键词

ADO ActiveX SOM DTC RDS

1 ADO 访问数据库

1.1 ADO(ActiveX Data Object)是建立在OLE DB基础上的访问

开放数据库连接(ODBC)提供了一种标准方法来访问关系型数据库中的数据,它是建立Web应用的强大工具。然而信息也存在于非关系型数据源中,例如,电子信箱、文本文件,目录数据和多媒体文件也在企业环境中经常使用。

为了提供一致的方法来访问数据库,微软开发了OLE DB,它是ODBC之后的又一次革新,它允许应用程序访问各种类型的信息。OLE DB定义了一系列COM界面,将各种数据库管理服务封装在其中。COM就是组件对象模型,它是分布式对象的基础。OLE DB界面既可用于数据的产生者,也可用于数据的消费者。当OLE DB技术广泛使用时,人们期望能提供对各种不同类型数据源的OLE DB供应器。在OLE DB和ODBC之间有一个桥梁,使得你可以用现存的ODBC驱动程序来访问关联数据库中的信息。

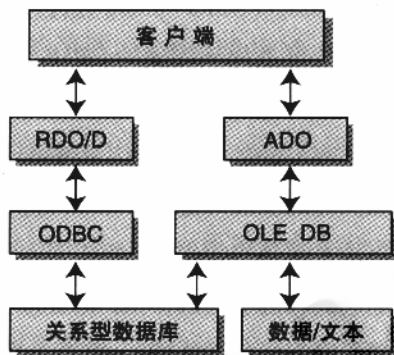


图1 ADO和OLE DB的关系

Active Data Objects (ADO)使得对OLE DB的使用变得容易了,ADO是一系列对象,封装了OLE DB中最常使用的一些特性,ADO对象可以在Visual Basic或Visual C++中使用,也可以在服务器端脚本中使用。ADO和OLE DB的关系如图1所示,ADO处于OLE DB之上并为应用程序提供一个对OLE DB的容易使用的应用程序接口(API),图中有多个OLE DB供应器来访问各种类型的数据源,OLE DB和ODBC之间的桥梁使得可以用现存的ODBC驱动程序来访问数据库。

1.2 ADO与ASP技术的结合,是目前主要的访问方法

微软的ASP技术是一种运行于服务器端的脚本编写模型,它使开发者可以使用几乎所有的脚本语言(VBScript、Jscript或Perl等)编写脚本,这些脚本可以执行应用程序逻辑,并能够调用ActiveX组件执行特殊任务,如数据库查询、文件输入输出等。

ADO是ActiveX组件中最重要的一个,是ASP技术的核心之一,它集中体现了ASP技术丰富而灵活的数据库访问功能。图2显示ASP中,ADO是如何完成数据访问的。

用ADO访问数据库更类似于编写数据库应用程序,ADO把绝大部分的数据库操作封装在七个对象中,在ASP页面中编程调用这些对象执行相应的数据库操作。ADO建立了基于Web方式访问数据库的脚本编写模型,它不仅支持任何大型数据库的核心功能,而且支持许多数据库所特有的特性。ADO使用本机数据源,通过ODBC访问数据库。这些数据库可以

是关系型数据库、文本型数据库、层次型数据库或者任何支持ODBC的数据库。

ADO的主要优点是易用、高速、占用内存和磁盘空间少,所以非常适合于作为服务器端的数据库访问技术。相对于访问数据库的CGI程序而言,它是多线程的,在出现大量并发请求时,也同样可以保持服务器的运行效率,并且通过连接池(Connection Pool)技术以及对数据库连接资源的完全控制,提供与远程数据库的高效连接与访问,同时它还支持事务处理(Transaction),以开发高效率、高可靠性的数据库应用程序。

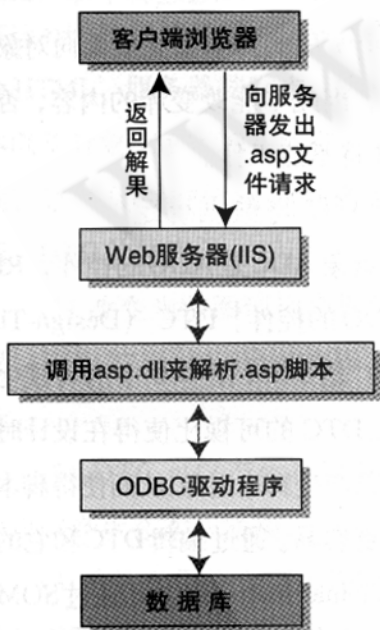


图2 ASP的工作原理

2 远程数据服务 RDS

2.1 RDS 是一种“动态”的访问方式

RDS是IIS 4.0中新提出的概念,它是由ASP中原来的Advanced Data Connector(ADC)发展而来的。在IIS 4.0中,RDS与ADO集成到一起,使用同样的编程模型,提供访问远程数据库的功能。

ADO虽然能够提供非常强大的

数据库访问功能,但是它不支持数据远程操作(Data Remoting)。换句话说,ADO只能执行查询并返回数据库查询的结果,这种结果是静态的,服务器上的数据库与客户端看到的数据没有“活的连接关系”。假如,客户端需要修改数据库中的数据,就必须构造修改数据的SQL语句,执行相应的查询动作。

而RDS就比ADO更进一步,它支持数据远程操作。它不仅能执行查询并返回数据库查询结果,而且这种结果是“动态的”,服务器上的数据库与客户端看到的数据保持“活的连接关系”。即把服务器端的数据搬到客户端,在客户端修改数据后,调用一个数据库更新命令,就可以将客户端对数据的修改写回数据库,就象使用本地数据库一样。而不必象ASP中那样提交给服务器,由服务器来介绍脚本,将sql查询写回数据库。

RDS是一套组件,利用它,可以创建访问与ODBC兼容的数据库的web应用程序。RDS将记录集数据绑定到web页上的数据感知html组件或ActiveX控件上。使用RDS需要COM组件、html和客户端脚本,可以将RDS.DataControl对象添加到一个html文档中,并设置其属性以检索数据,然后设置html组件属性,以将数据绑定RDS.Control上,同时也可以编写客户端脚本,用来在运动中请求一组数据或将改变提交给数据库。

RDS使用客户端和服务器的组件,客户端组件包含在IE4.0中,服务器组件当最小或典型安装Internet Information Server 4.0以后就有了。

2.2 RDS的优点

(1) RDS是一种透明的基于客户端缓存的架构:rds能够将大量的结果集保留一份在客户端,客户端使用这

些缓存是完全透明的,从而避免了对数据库的反复查寻,大大提高了应用性能。

(2) 在客户端无须安装任何数据驱动:无论是哪一种具体的数据源,RDS都能够在客户端进行处理。

(3) 使用ODBC数据源:RDS支持所有具备ODBC连接的数据库,从而大大扩展了可应用的范围。

2.3 RDS的应用

由于RDS与ADO集成,RDS的底层是调用ADO来完成的,所以也可以将RDS理解为ADO的RDS,即ActiveX数据对象的远程数据服务。所以RDS同样具有ADO的易用性,区别仅在于RDS需要与数据绑定控件一同使用,比如Sheridan的ActiveX DataBound Grid控件。

下面是使用RDS提供客户端数据远程访问的例子:

```
<html>
<head>
<SCRIPT ID=clientEvent-
HandlersJS LANGUAGE=javascript>
<!--
function button1_onclick(){
name_rds.Server="http://
sserve_IP"
name_rds.Connect="DSN=test;l;
server=server_name"
name_rds.SQL="select * from
table1"
name_rds.ExecuteOptions=1
name_rds.Refresh
-->
</SCRIPT>
</head>
<body>
<OBJECT classid=ClsId:
BD96C556-65A3-11D0-983A-
00C04FC29E33 height=1
```

```
id=name_rds style="DISPLAY: none" width=1>
<PARAM NAME="ExecuteOptions" VALUE="2">
<PARAM NAME="FetchOptions" VALUE="3">
<PARAM NAME="SortDirection" VALUE="-1">
<PARAM NAME="InternetTimeout" VALUE="300000">
</object>
<INPUT type="button" value="Button" id=button1 name=button1 LANGUAGE=javascript onclick="return button1_onclick()">
<table DataSource="# name_rds " >
<thead>
<tr><td>字段名</td>
.....
</tr>
</thead>
<tbody>
<tr><td><span datafld=" 字段名 "></td>
.....
</tr>
</tbody>
</table>
```

为了表现出 RDS 与 ADO 的不同,这里使用了一个按钮,当按动按钮时,表格显示出对应字段的数据。所以,RDS 在 ADO 的基础上通过绑定的数据显示和操作控件,提供给客户端更强的数据表现力和远程数据操纵功能。

RDS 实现了将传统的客户机/服务器技术与 web 相结合,不再象过去那样是只读的、静态的,通过使用本地数据缓存,终端无需进行复杂的交互就可以完成对大量数据的操纵。

3 SOM 模型与 DTC

3.1 SOM 模型的特点与优点

SOM (The Scripting Object Model) 是微软在 Visual Interdev 中提供的一种面向对象的可视化脚本对象模型。通过 html 和脚本上的面向对象的编程模式,SOM 使得 web 上的应用开放更快更容易。它不仅能完成客户端的 DHTML 的功能,而且具备服务器端的脚本处理能力。在数据库访问上,以其可视化,结合 DTC 对象的数据绑定,而十分快捷实现数据访问和控制。并且功能可延伸到 RDS。

值得一提的是,SOM 在数据库访问上,仍然是建立在 ADO 的基础上,就象 ADC (ActiveX Data Control) 与 ADO 之间的关系,ADC 只是利用 ADO 技术来完成数据访问的一个可视化的控件。

DHTML 对象模型 (DOM) 将网页及其中的元素当作对象,客户端脚本可以使用这些对象,可以建立事件处理函数来处理相应的事件。然而在服务器脚本中定义的对象不可以被客户端脚本访问,要想在服务器上处理信息,必须将网页某个元素的内容提交给服务器,这可通过 HTTP GET 或 POST 来实现,Active Server Framework (ASF) 提供了 Request 对象来让服务器脚本获取窗体中的内容,Application 和 Session 对象用来保存由许多页组成的站点的状态信息。

但这种技术有几点不利之处,所有的窗体数据都必须提交给服务器,一个新页必须传送到客户端来更新显示,这增加了服务器的负担,你还必须编写额外的服务器脚本来保存应用和对话的状态。由 Visual InterDev 提供的脚本对象模型 (SOM) 解决了这些问题,下面列出了它的优点:

(1) SOM 允许你用同样的面向对

象技术来设计 Web 应用,就象其他的一些软件技术如 Visual Basic 一样。这种方法简化了用于处理一个网页的脚本。

(2) 网页中某个元素的事件处理函数可以用客户端或服务端脚本来建立,SOM 直接激活此函数,你不再需要编写提交窗体内容给服务器的代码了。

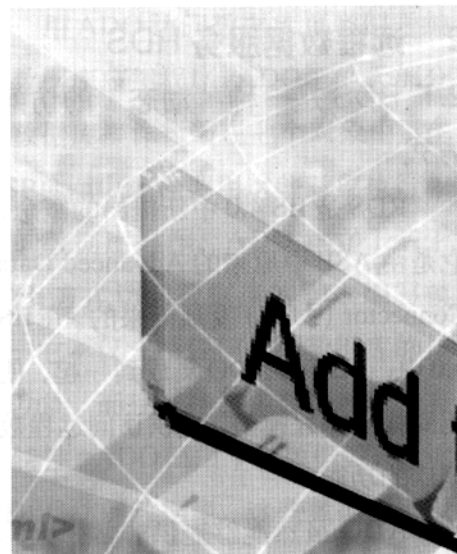
(3) 由多个网页组成的应用的状态信息自动保存,不再需要编写代码来保存这些信息。

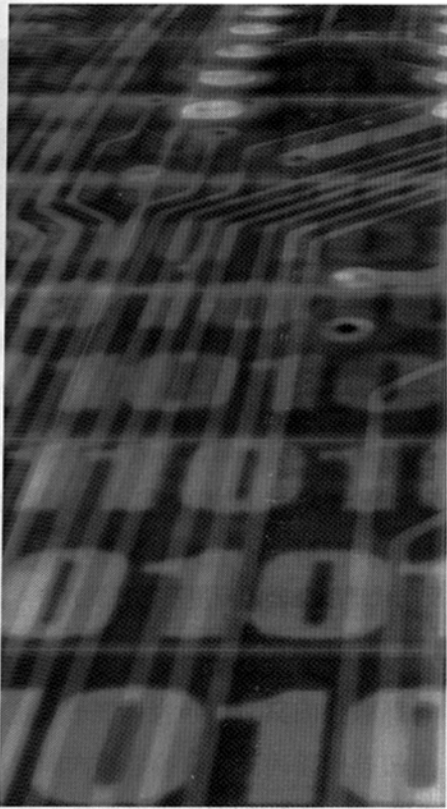
(4) 组成 Web 应用的每个网页都可以是一个对象,你可以为此对象取名并定义属性和方法,客户端脚本可以使用这些属性和方法。

Note: SOM 通过脚本库 (Script Library) 里的脚本来完成面向对象的编程。因此不能改变库的内容,否则 SOM 将不能工作。

3.2 DTC 与数据绑定

就象 ADC 是 ADO 的控件,RDC 是 RDO 的控件,DTC (Design-Time 控件) 是 SOM 在 web 中的可视化的应用。DTC 的可视化使得在设计时间就可以产生脚本对象,从而使得脚本的编写更容易。通过编辑 DTC 和它的属性,Visual InterDev 自动通过 SOM 产生运行时的代码。实际上,可以认为 DTC 是一个脚本产生器。Design-Time





端的DTC (或者说html中DTC) 则又变成RDS方式对数据库的访问。相比ADO、RDS, DTC以可视化的特点使得编程更容易更少出错。

DTC访问数据库, 使用Recordset DTC 控件和数据绑定控件。Recordset DTC 为其他数据控件提供数据源。Recordset DTC 提供了一个数据连接、数据库对象和一些属性, 其使用与ADO中的Recordset 极其相似。下面通过此控件和Grid 数据控件来实现上两例中不同的访问方式。(visual InterDev 环境)

首先, 在服务器端访问数据库的实现

① 建立projct1 工程, 利用本机 D S N 建立与数据库的连接 Connection1。此连接将自动改变 global.asa 文件。

② 新建 test.asp 文件, 将DTC 工具栏中的 Recordset DTC 放入 web 页中。在页属性中设置SOM有效, 脚本平台在 Server 端。

③ 设置 Recordset1 对象使用 Connection1 连接, 数据表或 sql 查询 (select * from table1) 等属性。

④ 将 Grid DTC 放入 web 中, 在其属性中绑定数据。Data 中设置数据源为 Recordset1, 将欲输出的字段勾选。

至此, 一个服务器端访问数据库的 asp 程序完成。全过程不需要写一个代码, 完全可视化编程。而且, 可以改变Grid DTC的属性, 得到不同的输出样式。

其次 在客户端访问数据库的实现。方法步骤同上, 不同之处只是: 建立test.html 文件, 脚本平台在Client 端。

Note: 无论是在服务器端还是在客户端访问数据, 只要是在远端开发

web 程序, 还需要修改 Recordset 和 global 中 dsn 为服务器中使用的DSN。这是因为远端的 dsn 只是为了能在设计时间对服务器端数据库控制而设的数据源, web 页真正产生后, 是放在服务器端的, 不管是 asp 还是html, 其对数据库的访问应当使用其所在主机的DSN。相反, 如果在服务器端开发 web 程序, 则不会产生这样的问题, 因为设计时间的 DSN 和运行时间的 DSN 是一样的。

通过这个例子, 可以充分看到 SOM访问数据库的快捷和方便。而且在客户端访问时, 可以动态的获得数据, 不需频繁的象 asp 中那样提交服务器, 在html中不用刷新页面, 就能访问到数据。有利于更好的控制页面。

本文以三种不同的方法来完成同样的数据访问, 并以表格显示, 相对而言, 各有千秋。ADO是ASP中推荐使用的方式, 它功能强大, 使用方便, 是ASP的核心技术之一, 但是它在提供用户远程操作数据库的功能时, 比较复杂, 实现起来有一定的难度。而RDS是基于ADO的, 并提供远程操作数据库的较好方法, 但在程序控制上不易掌握, 特别是其数据绑定控件较少, 对 web 开发造成影响。SOM对数据库的访问简单好用, 功能广泛, 数据控件丰富, 输出样式灵活。并且具有 ADO 和 RDS 两种访问方式。其可视化的编程方法是目前网页开发的流行方法。在数据库访问的可视化方面, Visual InterDev 的SOM的确提供一个很好的模型。■

参考文献

- 1 杨威, asp3.0网络开发技术, 人民邮电出版。
- 2 微软, MSDN Library Visual Studio 6.0 版。

控件 (DTC) 只在设计时执行并产生在运行时执行的代码, 这些代码中包含HTML、服务器端脚本、客户端脚本以及对象。每个DTC都有一系列属性, 这些属性的值将影响产生的代码。DTC 的特点:

(1) 改变当前图形用户界面的设置, 可以控制运行时间的脚本的行为。

(2) DTC 可以产生在服务器端, 也可以产生在客户端。

(3) 允许数据绑定到远程数据库。使用 DTC 需要指明脚本平台, 即此 DTC 运行在客户端还是服务器端。也就是说, 产生的脚本对象是客户端对象还是服务器端对象, 如果不指明, 则 html 文档中的 DTC 默认 (inherit from page) 为客户端对象, asp 文档 DTC 默认为服务器端对象。服务器端的 DTC 需要 SOM 的支持, 因此在添加 DTC 到 asp 网页时, 必须在页属性中确定 SOM 可用。

在访问数据库中, 服务器端的 DTC (或者说 asp 中 DTC) 实际上就是 ADO 方式对数据库的访问。而客户