

Intranet 连接数据库的两种技术

童晓阳 (成都西南交通大学模拟中心 610031)

摘要:本文介绍了 Intranet 连接数据库的基本工作机制,剖析了用 IDC 和 Visual J++ 的 RDO 对象连接数据库的工作过程,并给出实例,最后对这两种技术进行了比较。

关键词:Intranet 数据库 IDC Java

Intranet 是采用 Internet 技术在企业组织内部建立的局域网,它与外界世界安全隔离。利用它在内部网上进行 Web 信息发布,员工通过浏览器,可共享图文并茂的 HTTP 信息,如果再与企业内部的数据库连接起来,就可获得更加丰富的数据资料,提高工作效率。Intranet 连接数据库建立在 ODBC 的基础上,可连接各种数据库,本文介绍了两种连接数据库的技术。

1. Intranet 连接数据库的基本工作机制

Intranet 连接数据库的过程属于客户/服务器模式,客户机从浏览器的网页里提出查询数据库的请求,此请求传到 Web Server 服务器的 IIS(Internet Information Server),经过接口应用程序对用户事先编好的动态应用程序处理,将其中的 SQL 请求提交给 ODBC,ODBC 接口再根据系统数据源的类型,调用相应的数据库驱动程序,从数据库中提取数据,经服务器的接口应用程序,将查询的数据集返回给客户机的浏览器,以某种形式显示出来。客户请求在客户机提出,数据的处理在服务器完成,网络线上流动的只有客户请求和查到返回的数据,传输数据少,工作效率高。

这里的动态应用程序要放到服务器的专门发布 Web 的目录中,此目录的任务就是存储向局域网发布公共信息的文件。动态应用程序可由两种技术来实现,IDC(Internet Database Connector 数据库连接器)和 Java 的连接数据库技术,而用 Java 连接数据库有 Microsoft 公司的 Visual J++ 的 RDO 对象和 Sun 公司的 JDBC 类。本文采用 Visual J++ 的 RDO 对象。

2. IDC 技术

IDC 的工作过程由六部分组成,如图 1 所示:

(1) 由 Internet Information Server 接收 URL。由 Web 服务器发送 URL。

(2) Internet Information Server 装载 Httpodbc.dll,并提供其 URL 中保存的信息。Idc 文件映射到 Httpodbc.

dll。Httpodbc.dll 装载并获得来自 URL 并传递给 Internet Information Server 的 Internet 数据库连接器文件名(及其他条目)。

(3) Httpodbc.dll 读取 Internet 数据库连接器文件。Internet 数据库连接器文件含有几个如下格式的项 field: value, Datasource: ODBC 数据源, Template: HTML 扩展文件(用于合并查询结果)等项。

(4) IDC 连接到 ODBC 数据源,并执行 Internet 数据库连接器文件内的 SQL 语句。IDC 生成到 ODBC 数据源的连接。在本示例中,装载 SQL Server 数据库的 ODBC 驱动程序,并连接在数据源的定义中指定的服务器。一旦生成了连接,Internet 数据库连接器文件中的 SQL 语句就被发送到 SQL Server ODBC 驱动程序,后者按顺序将其发送到 SQL Server。

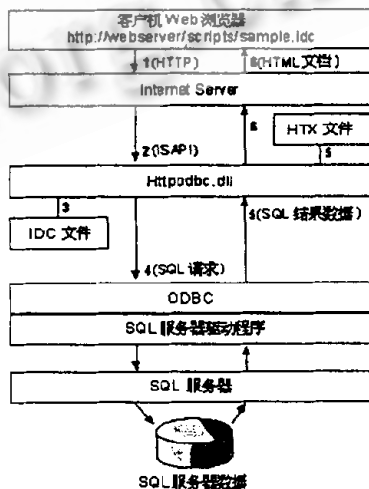


图 1

(5) IDC 从数据库中取出数据,并将其合并到

HTML 扩展文件。执行 SQL 语句之后, IDC 读取 Sample.idc 中指定的 HTML 扩展文件 Sample.htx。HTML 扩展(.htx)文件包含特别 HTML 标志, IDC 用标志控制从 SQL 语句中返回的数据合并入何处以及如何合并。

(6) IDC 将合并的文档发送回 Internet Information Server, 并返回到客户。数据合并入 Sample.htx 之后, 将完整的 HTML 文档发送回客户。

Idc 和 htx 文件如下:

Idc 文件:

```
Datasource: Web SQL //在控制面板 ODBC
                //设置的系统 DSN
Username: sa      //用户名
Template: sample.htx //HTML 扩展文件
SQLStatement: select * from employees
```

htx 文件:

```
.....
< %begindetail% >
Full Name: < %name% > //显示查询结果
E-mail Name: < %email% > .....
< %enddetail% > .....
```

3. Java 的 RDO 技术

Visual J++ 包括许多由 COM(Component Object Model)定义的类型库(Type libraries), 其中的 RDO 对象类型库——远程数据对象(Remote Data Object), 通过封装的一些类能处理存储在远程的 ODBC 数据源中的数据。RDO 控件能从结果集中的一行数据移到另一行, 显示和操作每行数据。RDO 控件提供了在远程数据和获得数据控件的接口。在 Java 文件开始要引入 RDO 和 COM, 加入 import msrdo20.* 和 import com.ms.com.*。Java 的 RDO 实现与数据库连接的过程如下:

- (1) 首先建立与 ODBC 数据源的连接;
- (2) 获取数据, 形成数据结果集(rdoResultset);
- (3) 将当前数据传递给数据显示控件来显示, 使每一列的数据对应视中的一个显示控件;
- (4) 允许用户获得当前记录的指针;
- (5) 对数据进行操作, 如增加、修改、删除等, 并把对数据的改变返回给数据源。

下面介绍如何连接数据库, 获得数据集, 以及四个数据控制按钮的事件处理方法:

```
rdoConnection Connection = new rdoConnection();//
打开与数据库的连接
```

```
m-IConnection = (-rdoConnection) Connection;
m-IConnection.putConnect("DSN = Sample Database;
UID = ; PWD = ; Database = Sample Database;"); //指出
数据库的来源
m-IConnection.EstablishConnection(v1, v2, v3); //
建立连接
v4.putInt(ResultsetTypeConstants.rdOpenKeyset);
//设置连接的参数
v5.putInt(LockTypeConstants.rdConcurReadOnly);
v6.putInt(0); //只读
m-IResultSet = (-rdoResultset) m-IConnection.Open-
ResultSet("SELECT * FROM Employees", v4, v5, v6); //
提交 SQL 查询
m-form = new SimpleForm(m-IResultSet); //形成
视, 显示查询结果
add("Center", m-form); //将视增加到 applet
m-IResultSet.MoveFirst(); //得到查询结果的第一
条记录
public boolean action(Event evt, Object what)
{ // 处理单击 "First" 按钮
if(evt.target == m-first)
{ m-IResultSet.MoveFirst();
m-form.showData(); //显示此记录
return true; }
// 处理单击 "Prev" 按钮
if (evt.target == m-prev)
{ m-IResultSet.MovePrevious();
// 检查是否超出头条记录
if (m-IResultSet.getBOF())
m-IResultSet.MoveFirst();
m-form.showData();
return true; }
// 处理单击 "Next" 按钮
if (evt.target == m-next)
{ m-IResultSet.MoveNext();
// 检查是否超出尾条记录
if (m-IResultSet.getEOF())
{ Variant v1 = new Variant();
m-IResultSet.MoveLast(v1);
m-form.showData();
return true; }
// 处理单击 "Last" 按钮
```

```
if (evt.target == m-last)
{ Variant v1 = new Variant();
  m-IResultSet.MoveLast(v1);
  m-form.showData();
  return true; }
return false; //其他不处理 }
```

4. 结论

(1) IDC 技术将查询的数据结果放在 htx 页面中显示, 而 htx 页面与发出查询请求的页面是不同页面。Java 可在当前页面显示结果, 它是通过编程刷新显示面板显示新数据。

(2) 从 IDC 的 htx 文件的显示语句可看出, 它将查询到的所有记录依次显示出来, 用户通过移动窗口的垂直滑块来查看不同的记录; Java 由四个按钮控件来查阅各条记录, 不须翻页, 使用很方便, 当然前提是要对按钮控

件的单击事件编程。

(3) 前面举例是最简单的 SQL 查询, 也可以从页面以表单(Form)输入查询条件。IDC 的 SQL 语句是事先写好在 IDC 文件中, 将表单输入的值代替 SQL 的 Where 所需值, 但由于 SQL 已定做好, 不可能改变; 而 Java 可动态生成 SQL 语句, 实现复杂的查询。

(4) IDC 只能得到数据项, Java 还能从数据库提取更多有关的信息, 如记录的总数, 当前记录的位置等等。

总之, IDC 技术不是采用一种编程的语言来制作应用程序, 它能编写较简单的交互式页面, 适用于查询中小型数据库数据, 便于掌握, 不需要页面制作者具有很高的编程能力。而 Java 对编程者的编程水平要求较高, 但它能为网页提供更强的和专门的功能。Intranet 与数据库的连接, 可根据实际情况采取不同的技术来实现。

(来稿时间: 1999 年 1 月)