ODP.NET 访问 ORACLE 的方法

To Access Oracle Data By ODP.NET

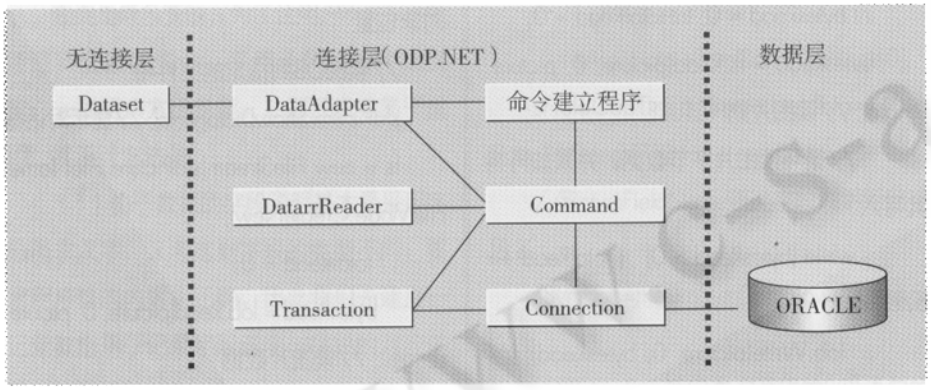
姚世军　柴育梅　（郑州市解放军信息工程大学理学院　450052）

摘　要：ODP.NET是专用的.NET访问ORACLE的高效方法。本文描述了ODP.NET的主要特性和结构并通过例子介绍了它的使用方法。

关键词：ODP.NET　ODBC　OLE DB

1 ODP.NET 简述

微软.NET结构通过OLE DB或ODBC对象可访问ORACLE数据库；但要想从ORACLE的应用中得到更多高级的数据库性能支持，应该选择专用的ORACLE连接方法而不是通用数据库连接。.NET编程语言没有专用的ORACLE连接程序。为此ORACLE公司为.NET编写了Oracle Data Provider(ODP.NET)，它是在.NET环境下访问ORACLE数据的API组成。ODP.NET的对象模型结构如下图：



ODP.NET通过Connection对象与数据库连接，用Command来修改数据，用DataReader读出结果集，Dataset是无连接的数据缓冲区，命令建立程序构思来自SELECT的更新命令。

ODP.NET提供了在.NET环境下访问ORACLE数据库的第三种方法,在这三种方法中，ODP.NET是与.NET环境关系最密切的，它不需要OLE DB或ODBC作为中间层,直接与ORACLE连接。可以在任何.NET编程语言中使用ODP.NET来访问ORACLE。

2 ODP.NET 的主要特性

2.1 XML 特性

ODP.NET提供了两种使用XML的方法：XML DB和System.XML服务。

XML DB是ORACLE的高性能XML存取和恢复技术，它能够管理和存取结构化和非结构化数据，它在XML和SQL之间提供了完全透明的互操作性。.NET客户可从ODP.NET中调用PL/SQL程序来激活XML DB的功能。

微软System XML用于处理来自.NET数据提供者的数据。ODP.NET与System XML的API一起通过ODP.NET的DataAdpater接口来传递数据。

XML DB为数据库服务器上的数据提供服务，而System XML处理客户端的XML。这样ODP.NET使程序员有更大的灵活性来选择适合的XML技术。

2.2 性能改进

ODP.NET有更好的性能，它不需要用数据访问中间件，而中间件是数据传输的附加层且会增加数据访问的不稳定性。ODP.NET也有高级的数据库访问功能。

2.3 处理内置数据类型

ODP.NET对恢复和处理ORACLE内部的数据类型，如LOB和REF游标有许多优化。使用ODP.NET中的TUNING特性可以提高.NET数据访问的性能。

.NET中为不同编程语言中引进了统一的数据类型。使用ODP.NET，ORACLE用户可以访问.NET的数据类型也可访问ORACLE的内置数据类型。在.NET程序中可完全处理ORACLE类型并可与.NET数据类型互相操作。支持ORACLE中如REF游标，各种LOB,LONG,RAW,LONG RAW及N系列高级数据类型。ODP.NET可以访问PL/SQL输出参数返回的REF游标，而用其他两种方法只能使用线性方式。

2.4 其他特性

ODP.NET可以完全执行数据库中的PL/SQL存储过程和函数，而不管PL/SQL是定义在包中与否。程序员可以从结构存储过程中返回多个结果集，在使用PL/SQL时有很大的灵活性。ODP.NET参入事务应用程序。在.NET环境中ODP.NET应用Microsoft Enterprise Services作为事务处理器。

3 用ODP.NET方法ORACLE数据库例子

3.1 使用 ODP.NET 的条件

只需在客户端或中间层机器中安装ODP.NET。ODP.NET可以与Oralce8以上的数据库服务器一起工作，但客户端必须用Oracle9i 9.2版以上；它不能与8i的客户端一起工作。

客户端系统可以是WINDOWS 98/NT/2000/XP；只要安装.NET结构和ORACLE网络。

3.2 使用 ODP.NET 的步骤

（1）安装ODP.NET软件；

（2）将ORACLE.DATAACCESS.DLL安装在Visual Studio's Solution Explorer;

（3）在C#类中，用USING增加两个名字空间在代码行前面：Oracle.DataAccess.Client; Oracle.DataAccess.Types。

3.3 用ODP.NET访问ORACLE的程序例子

在以上任务完成后，可按下面步骤编写程序来访问ORACLE：

（1）用OralceConnection对象来进行数据连接，有三个参数：数据源、用户名和口令。数据源是tnsnames.ora文件中的合法TNS名字。用该连接的OPEN方法来打开。

（2）用OracleCommand对象建立命令，第一个参数是将要执行的任何合法的SQL命令，第二个参数是连接数据库的OracleConnection对象。要改变SQL语句，只需改变OracleCommand对象的CommandText特性。

（3）用方法ExecuteNonQuery()来执行建立的命令，当不需要INSERT、UPDATE或DELETE返回的行时，调用该方法，它取SQL语句做为参数，并将它传给数据库执行，返回影响的行数。对SELECT语句用ExecuteReader()。

下面是一个用C#编写的通过ODP.NET访问ORACLE数据库的例子程序。这里假定表中已有所需的数据记录及其他对象。表名为EMPLOYEES，有四个字段：id number(10) not null； name varchar2(32) not null； photo blob；bignumber number(38, 37)。程序的主要功能是演示通过ODP.NET对表的插入、删除和更新以及LOB字段的处理操作。

```
using System; using System.Drawing; using System.Collections;
using System.ComponentModel; using System.Windows.Forms;
using System.Data; using System.Data.OleDb; using Oracle.DataAccess.Client;
using Oracle.DataAccess.Types; using System.IO;
namespace Test
{
public class frmTest : System.Windows.Forms.Form
{ private void ClearDatabase() //清空数据库的方法
{ try {OracleConnection dbConn = new OracleConnection("Data Source=
" + txtDataSource.Text + ";" + "User Id=" + txtUsername.Text + ";"
+ "Password=" + txtPassword.Text + ";"); //新建数据源连接
dbConn.Open(); //打开数据库连接
OracleCommand cmd = new OracleCommand(
"delete from employee", dbConn); //新建SQL命令，删除表的所有行
cmd.ExecuteNonQuery(); //执行SQL命令
bConn.Close(); //关闭连接 }
//测试连接的方法
private void btnTestConnect_Click(object sender, System.EventArgs e)
{try {OracleConnection dbConn = new OracleConnection(
"Data Source=" + txtDataSource.Text + ";" + "User Id=" + xtUsername.Text + ";"
+ "Password=" + txtPassword.Text + ";");
dbConn.Open(); dbConn.Close(); //打开、建立、关闭连接
MessageBox.Show(this, "Connection open/close sucessful");}
//读数据库表
private void btnReadWrite_Click(object sender, System.EventArgs e)
{ ClearDatabase(); //清除数据库
try { //用户名和口令建立数据库连接
OracleConnection dbConn = new OracleConnection("Data Source=" +
txtDataSource.Text + ";" + "User Id=" + txtUsername.Text + ";" +
"Password=" + txtPassword.Text + ";");
dbConn.Open(); //打开连接
OracleCommand cmd = new OracleCommand("insert into employee
values " + "(1, '张三立', empty_blob(), " + "1)", dbConn); //建立新SQL命令
int rows = cmd.ExecuteNonQuery(); //执行SQL命令
cmd.CommandText = "select * from employee where " + "id = 1";
OracleDataReader reader = cmd.ExecuteReader(); //执行SQL语句
reader.Read(); //读数据集
dbConn.Close();} }
//处理大对象的方法
private void btnLob_Click(object sender, System.EventArgs e)
{ ClearDatabase();//调用清空数据库方法
try {OracleConnection dbConn = new OracleConnection(
"Data Source=" + txtDataSource.Text + ";" +
"User Id=" +
txtUsername.Text + ";" +"Password=" + txtPassword.Text + ";");//建立连接
dbConn.Open(); //打开连接
//插入记录
OracleCommand cmd = new
```

下转第 51 页 >>

```
OracleCommand("insert into employee values "
    + "(1, '李存根', empty_blob(), " + "1)",
dbConn);//新建SQL命令
    cmd.ExecuteNonQuery();//执行SQL
    // 打开更新的记录，更新时需要事务
    OracleTransaction trans = dbConn.
BeginTransaction(); //开始事务处理
    cmd.CommandText = "select * from em-
ployee where " + "id = 1 for update";
    OracleDataReader reader = cmd.
ExecuteReader();//执行SQL语句
    reader.Read(); //读数据集
    OracleBlob lob = reader.GetOracleBlob
(2); // BLOB的序号
    ofdPicture.ShowDialog(this);
    // 打开图片并保存为字节流
    FileStream fs = new FileStream(ofdPicture.
FileName, FileMode.Open); //建文件流
    byte[] picture = new byte[4096];
    int bytesRead = 0; int totalRead = 0;
    bytesRead = fs.Read(picture, 0, picture.
Length); //读图片到PICTURE
    //循环读图片字节数据，并累加再将
其写入
    while (bytesRead > 0) { totalRead +=
bytesRead;
    lob.Write(picture, 0, bytesRead); //
写入LOB
    bytesRead = fs.Read(picture, 0, picture.
Length); } //从文件读图片数据
    trans.Commit(); //提交事务
    trans = dbConn.BeginTransaction(); //开
始事务
    cmd.CommandText = "select * from em-
ployee where " + "id = 1";
    reader = cmd.ExecuteReader();reader.Read
();
    lob = reader.GetOracleBlob(2);//得到大
对象位置
    //读BLOB到指定的文件中
    sfdPicture.ShowDialog(this); //显示对话框
    fs = new FileStream(sfdPicture.FileName,
FileMode.CreateNew);
    totalRead = 0;
    bytesRead = lob.Read(picture, 0, picture.
Length); //读图片数据
    while (bytesRead > 0)
    { totalRead += bytesRead; //字
节统计
    fs.Write(picture, 0, bytesRead); //
文件写入
    bytesRead = lob.Read(picture, 0,
picture.Length); //读图片数据
    }
    dbConn.Close();//关闭数据库连接
    fs.Close(); //关闭文件
    } } }
```