使用数据库操作引擎ACE的OLEDB

jet和ace引擎的区别：jet可以操作97-2003版本的excel，而且不能操作打开的excel；但是ace不仅可以操作97-2003，也可以操作2007及以后的excel，而且ace可以在打开excel状态下操作excel。

**Microsoft.Jet.OLEDB.4.0和Microsoft.ACE.OLEDB.12.0的区别**

在通过ADO对Excel对象进行连接时（此时Excel则认为是一个数据源），需要配置对Excel数据源对应的连接串，这个连接串中包括了Provider信息（其实类似对**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)**进行连接操作时，都需要指定连接字符串），以下是一行连接串源代码：

strConnString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & strExcelFilePath & ";Extended Properties=Excel 12.0"

这里的Provider使用了Microsoft.ACE.OLEDB.12.0，其实除了Microsoft.ACE.OLEDB.12.0，还有Microsoft.Jet.OLEDB.4.0，它们俩者之间有什么联系和区别呢：

共同点：都是做为连接Excel对象的接口引擎

不同点：

对于不同的Excel版本，有两种接口可供选择：Microsoft.Jet.OLEDB.4.0（以下简称 Jet 引擎）和Microsoft.ACE.OLEDB.12.0（以下简称 ACE 引擎）。

Jet 引擎，可以访问 Office 97-2003，但不能访问 Office 2007。

ACE 引擎是随 Office 2007 一起发布的数据库连接组件，既可以访问 Office 2007，也可以访问 Office 97-2003。

另外：Microsoft.ACE.OLEDB.12.0 可以访问正在打开的 Excel 文件，而 Microsoft.Jet.OLEDB.4.0 是不可以的。

所以，在使用不同版本的office时，要注意使用合适的引擎。

执行过程：

拼接连接字符串（包括驱动引擎，数据源）

构造sql命令语句

执行命令，得到结果

注意：如果驱动的方法有异常的话需要使用try-catch-finally

执行没有返回值的sql语句

public static void DoOleSql(string sql, string database)

{

OleDbConnection conn = new OleDbConnection();

conn.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + database + ";Extended Properties='Excel 12.0;HDR=no;IMEX=0';"; //拼接连接字符串

try

{

conn.Open(); //打开到数据库的连接

}

catch (Exception e)

{

Console.WriteLine(e.ToString());

return;

}

OleDbCommand oleCommand = new OleDbCommand(sql, conn); //构造命令语句

try

{

//执行语句，没有返回值

oleCommand.ExecuteNonQuery(); //执行

}

catch (Exception e)

{

Console.WriteLine(e.ToString());

}

finally

{

conn.Close();

}

}

从excel中取出数据

public static DataSet ExcelToDS(string Path)

{

string strConn = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + Path + ";Extended Properties='Excel 12.0;HDR=no;IMEX=0'"; //拼接字符串

OleDbConnection conn = new OleDbConnection(strConn);

conn.Open(); //打开到数据库的连接

string strExcel = "";

OleDbDataAdapter myCommand = null;

DataSet ds = null;

strExcel = "select \* from [sheet1$]";

myCommand = new OleDbDataAdapter(strExcel, strConn); //构造sql命令语句

myCommand.Fill(ds, "table1"); //执行sql命令并返回结果

return ds;

}

遍历DataSet中数据的几种方法：

假设ds是查询出来的DataSet数据

1. 多表多行多列

foreach (DataTable table in ds.Tables)

{

foreach (DataRow row in table.Rows)

{

foreach (DataColumn col in table.Columns)

{

Console.WriteLine(row[col].ToString());

}

}

}

1. 单表多行多列

foreach (DataRow row in ds.Tables[0].Rows)

{

foreach (DataColumn col in ds.Tables[0].Columns)

{

Console.WriteLine(row[col].ToString());

}

}

1. 单表一行多列

foreach (DataColumn col in ds.Tables[0].Columns)

{

Console.WriteLine(ds.Tables[0].Rows[0].ToString());

}

1. 单表多行一列

foreach (DataRow row in ds.Tables[0].Rows)

{

Console.WriteLine(row[0].ToString());

}