ADO.NET提供了一个面向对象的数据访问架构，通过System.Data使用

ADO.NET的核心对象有五个：connection,command,datareader,dataadapter,dataset

/\*根据对象性质不同可分为数据提供者和数据使用者

\*数据提供者：connection,command,datareader,dataadapter

\*数据提供程序：SQLsever，Oracle，OLE DB，ODBC

\*数据使用者：dataset

\*/

/\*

\*根据数据库存取模式可分为联机模式和脱机模式

\*联机模式：connection,command,datareader

\*脱机模式：connection,command,dataadapter,dataset

\*/

Connection

ConnectionString 连接字符串

/\*

\*provider 配置ole db数据驱动程序，一般为microsof.jet.oledb.4.0

\*data source 数据源位置，sqlsever为服务器位置，ole db为数据库路径

\*initial catalog 数据库名称

\*integrated security||tursted\_connection win身份验证：true，false；

\*user id 用户账号

\*password 密码

\*/

Open() 打开连接

Close() 关闭连接

StateChange事件 P314

Command

Connection 配置连接对象名

CommandText SQL命令文本

CommandType 补充说明CommandText的类型：Text，StoredProcedure，TableDirect

ExecuteNonQuery() 只返回返回受影响的数目

ExecuteReader() 返回一个DataReader对象

ExecuteScalar() 返回第一行第一列，通常为count，average，max这些返回值

DataReader:

FieldCount 获取列的数目

Read() 将指针指向下一行

GetValue(i) 得到列序号为i的值；相似的方法有getint16，getchar，getstring；\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*dr.GetValue(0)==>dr[0];\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GetName(i) 得到列序号为i的列名

GetOrdinal(s) 得到列名为s的列序号

HasRow 是否有一行或多行

Close()

（ i为int，s为string）

DataAdapter

SelectCommand 配置命令对象

Fill(dataset|datatable|dataset,srctable) 填充数据库，srctable为数据库中表的名字

Update() 把修改后的dataset更新到数据库中

/\*

\*DataAdapter一般和CommandBuilder一起用，后者根据修改生成命令文本，取代了原来的\*InsertCommand，DeleteCommand，UpdateCommand

\*/

CommandBuilder

DataAdapter 配置适配器对象

DataSet

Tables 获取DataTable对象集合

Relations 获取DataRelation对象集合

\*\*\*\*\*\*\*\*\*\*\*\*可以通过ds.Tables[“tb\_info”]锁定tb\_info表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

访问SQL Sever：需要Using System.Data.SqlClient;

联机模式：

SqlConnection conn=new SqlConnection();

SqlCommand comm=new SqlCommand();

SqlDataReader dr;

conn.ConnectionString=”data source=.;initial catalog=peopeldb;integrated security=true”;

conn.Open();

comm.Connection=conn;

comm.CommandText=” select \* from tb\_info”;

dr=comm.ExecuteReader();

string s=””;

for(int i=0;i<dr.FieldCount)

s+=dr.GetName(i)+” “;

s+=”/n”;

while(dr.Read())

{

for(int i=0;i<dr.FieldCount)

s+=dr[i]+” ”;

s+=”/n”;

}

conn.Close();

Console.WriteLine(s);

脱机模式：

string s=”data source=.;initial catalog=peopeldb;integrated security=true”;

SqlConnection conn=new SqlConnection(s);

conn.Open();

string c=” select \* from tb\_info”;

SqlCommand comm=new SqlCommand(c,conn);

SqlDataAdapter da=new SqlDataAdapter(comm);

SqlCommandBuilder cb=new SqlCommandBuilder(da);

DataSet ds=new DataSet();

da.Fill(ds,”tb\_info”);

conn.Close();

conn.Open();

da.Update(ds,”tb\_info”);

conn.Close();

访问OLE DB：需要Using System.Data.Oledb;

//与访问sqlsever不一样的只有ConnectionString

string s=”provider=microsoft.jet.oledb.4.0;datasource=d:\\peopledb.mdf;”

数据库组件

DataGridView

/\*

\*设置DataSource（DataSet或BindingSource）；

\*设置DataMember（string型）

or打开编辑列，添加列，修改DataPropertyName和HeaderText

\*/

DataSet //少用

BindingSource //设置DataSource

BindingNavigator //设置BindingSource

/\*BindingNavigator与BindingSource经常是一起出现的，而且BindingNavigator只与BindingSource有关联\*/

直接用数据源窗口连接数据库（最快）