Acceso a Base de Datos

Diseño y Programación Avanzada de Aplicaciones

Curso 2002-2003

Indice

ADO.Net

- ADO.Net ofrece dos espacios de nombres de clientes
  - Uno para SQL Server
  - Bases de Datos con interfaz Ole DB

Clases Compartidas

- ***DataSet***
  - Contienen un conjunto de objetos *DataTable* y relaciones entre dichas tablas.
- ***DataTable***
  - Consiste en uno o más objetos *DataColumn* y uno o más objeto *DataRow*
- ***DataRow***
  - Un conjunto de valores similares a la fila de una tabla.

Clases Compartidas (II)

- ***DataColumn***
  - Contiene la definición de una columna, incluyendo atributos como el nombre y el tipo de datos.
- ***DataRelation***
  - Representa un enlace entre dos objetos *DataTable* pertenecientes a un mismo *DataSet*. Permiten representar claves externas y relaciones maestro/detalle
- ***Constraint***
  - Restricciones que se aplican a una o un conjunto de columnas (Ej unicidad).

Clases específicas para Base de Datos

- SqlCommand,OleDbCommand
  - Encapsulan sentencias SQL y llamadas a procedimientos almacenados
- SqlCommandBuilderOleDbCommandBuilder
  - Permiten generar sentencias SQL
- SqlConnection,OleDbConnection
  - Representan una conexión a BD
- SqlDataAdapter, OleDbDataAdapter
  - Almacenan sentencias SQL y permiten llenar un DataSet y actualizar una Base de Datos.

Clases específicas para Base de Datos (II)

- SqlDataReader, OleDbDataReader
  - Un lector de datos conectado y unidireccional
- SqlParameter, OleDbParameter
  - Parámetro de un procedimiento almacenado
- SqlTransaction, OleDbTransaction
  - Una transacción sobre una base de datos.

Conexión

- Server= (local)\\NetSDK Indica el servidor
  - Proceso NetSDK del servidor local.
- Uid y pwd. Usuario y password
  - Si se desea utilizar la seguridad de windows se debe poner IntegratedSecurity=SSPI
- Database = Nombre de la base de datos

Ejecución de comandos

- ExecuteNonQuery
  - Ejecuta un comando y no devuelve ningún resultado
- ExecuteReader
  - Ejecuta un comando y devuelve un comando que implementa IDataReader (Permite iterar a partir de los registros recibidos)
- ExecuteScalar
  - Ejecuta un comando y devuelve un valor simple

ExecuteNonQuery

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";

string comando= "Update Articulos set precio = 10";

SqlConnection conn = new SqlConnection(source);

conn.Open();

SqlCommand cmd=new SqlCommand(comando,conn);

cmd.ExecuteNonQuery();

conn.Close();
```

ExecuteReader

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";

string comando= "Select codigo, descripcion from articulos";

SqlConnection conn = new SqlConnection(source);

conn.Open();

SqlCommand cmd=new SqlCommand(comando,conn);
SqlDataReader reader= cmd.ExecuteReader();

while (reader.Read())
        MessageBox.Show(  reader.GetInt32(0) + " " + reader.GetString(1));

conn.Close();
```

ExecuteScalar

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";
string comando= "Select count(*) from articulos";
SqlConnection conn = new SqlConnection(source);
conn.Open();
SqlCommand cmd=new SqlCommand(comando,conn);
object o =  cmd.ExecuteScalar();
MessageBox.Show(  o.ToString());
```

Ejecución de procedimientos almacenados

- Ejemplo de procedimiento almacenado

```
ALTER PROCEDURE ActualizaPrecios

        (
                @incremento int
        )

AS
 update articulos set precio = precio * ( 1 + @incremento / 100)

        RETURN
```

Ejecución de procedimientos almacenados (II)

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";
string comando= "Select count(*) from articulos";
SqlConnection conn = new SqlConnection(source);
conn.Open();
SqlCommand cmd=new
SqlCommand("ActualizaPrecios",conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.Add(new SqlParameter("@incremento",SqlDbType.Int));
  cmd.UpdatedRowSource = UpdateRowSource.None;
cmd.Parameters[0].Value = 100;
 cmd.ExecuteNonQuery();
```

DataReader

- Es el mecanismo más sencillo para seleccionar un conjunto de datos de la base de datos.
- Se suelen crear con el ExecuteReader

ExecuteReader

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";

string comando= "Select codigo, descripcion from articulos";

SqlConnection conn = new SqlConnection(source);

conn.Open();

SqlCommand cmd=new SqlCommand(comando,conn);
SqlDataReader reader= cmd.ExecuteReader();

while (reader.Read())
        MessageBox.Show(  reader[0] + " " + reader["descripcion"]);
conn.Close();
```

ExecuteReader

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";

string comando= "Select codigo, descripcion from articulos";

SqlConnection conn = new SqlConnection(source);

conn.Open();

SqlCommand cmd=new SqlCommand(comando,conn);
SqlDataReader reader= cmd.ExecuteReader();

while (reader.Read())
        MessageBox.Show(  reader.GetInt32(0) + " " + reader.GetString(1));

conn.Close();
```

Conjuntos de Datos o DataSets

- Un DataSet es un contenedor offline de datos.
- Puede contener datos que provienen de una BD o de cualquier otro origen
- Es un objeto de datos sin conexión.
  - Una vez completado con datos actuará independiente
  - No necesita conectarse a la BD

El objeto DataSet

- El DataSet es una representación de datos en memoria.
- Es un objeto de datos sin conexión.
  - Una vez completado con datos actuará independiente
  - No necesita conectarse a la BD

DataSet (II)

- ***DataTable***
  - Consiste en uno o más objetos *DataColumn* y uno o más objeto *DataRow*
- Formas de Generar
  - Utilizar el motor de ejecución
  - Escribir el código para generar la tabla
  - Utilizar el generador de esquemas XML

Utilizar la tabla generada

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";

string comando= "Select codigo, descripcion from articulos";

SqlConnection conn = new SqlConnection(source);
SqlDataAdapter da = new SqlDataAdapter(comando,source);
conn.Open();

DataSet ds = new DataSet();
da.Fill(ds,"Articulos");

conn.Close();
```


Hay que incluir using System.Data.SqlClient;

DataRow

- Filas de datos
- Permite el acceso a cada fila de la tabla

```
foreach (DataRow row in ds.Tables["Articulos"].Rows)
    Console.WriteLine("{0}, {1}", row[0],row["descripcion"]);
```

DataRow (II)

- Atributos del DataRowVersion
  - *Current*. Valor actual de la columna
  - *Default* Valor por defecto de la columna
  - *Original* Valor que se cargo de la base de datos. Si se llama al método AcceptChanges del DataRow es el mismo que el valor Current
  - *Proposed* . Valor propuesto si se utiliza el método BeginEdit. Cada columna tiene un valor propuesto hasta que se haga EndEdit o CancelEdit

DataRow(III)

```
foreach (DataRow row in ds.Tables["Articulos"].Rows)
Console.WriteLine("{0}, {1}",
   row[0,DataRowVersion.Current],
                        row[0,DataRowVersion.Original]);
```

DataRow (IV)

- Atributos del DataRowState
  - Unchanged. La fila no ha cambiado
  - Added Se ha añadido una nueva fila
  - Modified La fila se ha modificado
  - Deleted La fila se ha borrado
  - Detached Una fila está en este estado una vez ha sido creada

DataRow (V)

- Métodos sobre las filas
  - NewRow Crea una nueva Fila
  - Add. Permite insertar una nueva fila
  - Delete Permite Borrar una fila
  - Actualizar se puede hacer mediante operaciones de vectores

DataRow(IV)

```
DataSet ds = new DataSet();
            sqlDataAdapter1.Fill(ds,"Articulos");

    //metodo 1
            DataRow r = ds.Tables["Articulos"].NewRow();
            r["Codigo"] = 24;
            r["Descripcion"] = "Peana";
            r["Precio"] = 4;
            r["Stock"] = 6;
            ds.Tables["Articulos"].Rows.Add(r);

    //metodo 2
            ds.Tables["Articulos"].Rows.Add
                    (new  Object[]{25,"Casas",4,5});


            sqlDataAdapter1.Update(ds,"Articulos");
```

DataRow y DataColumn (II)

- Para comprobar el estado de una fila se dispone de las siguientes propiedades:

ExecuteReader

```
string source="server=ACER-4AE69X9LGH\\NetSDK; " +
                                                    "database=prueba10;"+
                                                    "integrated security=SSPI";

string comando= "Select codigo, descripcion from articulos";

SqlConnection conn = new SqlConnection(source);

conn.Open();

SqlCommand cmd=new SqlCommand(comando,conn);
SqlDataReader reader= cmd.ExecuteReader();

while (reader.Read())
        MessageBox.Show(  reader[0] + " " + reader["descripcion"]);
conn.Close();
```

El objeto DataSet (II)

- El DataSet está formado por 5 tipos de objetos diferentes:
  - Tablas
  - Filas
  - Columnas
  - Restricciones
  - Relaciones

El objeto DataTable

- Un ***DataSet*** es una colección de tablas en forma de ***DataTableCollection***
- Cada tabla individual almacena una referencia a este objeto en su propiedad ***Tables***
- Cada tabla residente en memoria se denomina ***DataTable*** (Puede haber varias o ninguna dentro de una colección)
- Cada ***DataTable*** contiene filas de datos.
- Cada fila contiene columnas de datos

DataRow y DataColumn

- Los objetos DataRow y Data Column forman el DataTable.
- Por medio de las propiedades y métodos de ambos se pueden ver actualizar, insertar y eliminar información de una tabla.
- DataColumn representa una columna
- DataRow representa una fila

DataRow y DataColumn (II)

- Para crear una nueva fila se utiliza el méodo ***NewRow***.
- Para aceptar los cambios se utiliza ***AcceptChanges***
- Para rechazar los cambios se utiliza ***RejectChanges***.