TWIONTECH

DBCONNECTOR .NET NAMESPACE

1.0.2

2012

CODED BY YUSUFCAN YILMAZ

Twiontech.DBConnector namespace includes 2 public class to access, execute query or read data from Mysql, Mssql or SQLite databases. The project has different files for x86 and x64. You should choose suitable type for your project. The main class library supports .Net 4.0 version.

Files

File Name	Content
MySql.Data.dll	The class library includes Mysql connection namespace. The class library version is: 6.5.4
System.Data.SQLite.dll	The class library includes SQLite connection namespace. The class library version is: 1.0.81
TwionTech.DBConnector-1.0.0.dll	The main class library includes all that you need to connect different databases with one class. The class library version is: 1.0.0

You should put Mysql.Data.dll and Sysem.Data.SQLite.dll into the same folder with TwionTech.DBConnector-1.0.0.dll or you should register that dll files to the system.

Classes

DB: The class for connect a database and run query.

Public Member Functions

	DB (int type, string ConnectionString)
void	**ClearParams** ()
void	**AddParam** (string name, object param)
int	**RunQuery** (string Query)
DBReader	**GetReader** ()
void	**Dispose** ()

Properties

System.Data.ConnectionState	**State** [get]

Members:

- **DB** (int type, string Host, string DataBase, string User, string Password)
  Defining function of class. Creates and connects to the DB.
  - int type: Type of database.
    0: Mysql
    1: Mssql
    2: SQLite

- void **ClearParams**()
  Clears all parameters.

- void **AddParam** (string name, object param)
  Adds parameter to use in query. For example: _db.AddParam(“id”,5);

- int **RunQuery** (string Query)
  Runs query on he connected database. For example: _db.RunQuery(“select * from users”);

- **DBReader GetReader** ()
  Returns DBReader to read data from the connected database.

- void **Dispose** ()

- System.Data.ConnectionState **State** [get]

DBReader: The class for read data from connected database.

Public Member Functions

	DBReader (int type, object reader)
bool	**Read** ()
int	**GetColumns** (object[] array)
void	**Dispose** ()

Properties

object	**this[string column]** [get]
object	**this[int column]** [get]

Members:

- **DBReader** (int type, object reader)
  Defining function of the class.

- bool **Read** ()
  Read data from table. If Reading is success, it returns true.

- int **GetColumns** (object[] array)
  Read multiple columns from table.

- void **Dispose** ()

- object **this** [string column] [get]
  Returns the data of column. For example: int _id= (int) _reader[“id”];

- object **this** [int column] [get]
  Returns the data of column. For example: int _id= (int) _reader[0];

DBAdapter: The class for create SqlAdapter to use DataTable, DataSet. Fill and Update functions same with orginal SqlAdapter functions.

Public Member Functions

	DBAdapter (int cType, object cLink, string Query)
int	**Fill** (DataTable DT)
int	**Fill** (DataSet DS)
int	**Update** (DataTable DT)
int	**Update** (DataSet DS)

An Example With DBReader:

```
using System;
using TwionTech.DBConnector;

namespace TwionTechDBConnector_ConsoleApplicationExample
{
    class Example
    {
        static void Main(string[] args)
        {
            DB _db = new DB(0, "data source=localhost; database=testingdb; UID=root;
PWD=123456");
            _db.RunQuery("select id,name,sirname from users");
            DBReader _reader = _db.GetReader();

            Console.WriteLine("Our Users ID/Name/Sirname:");

            while (_reader.Read())
            {
                Console.WriteLine("{0}/{1}/{2}",
                    (int)_reader["id"],
                    (string)_reader["name"],
                    (string)_reader["sirname"]);
            }

            _reader.Dispose();
            _db.Dispose();
        }
    }
}
```

An Example With DBAdapter:

```
using System;
using TwionTech.DBConnector;
using System.Data;

namespace TwionTechDBConnector_ConsoleApplicationExample
{
    class Example
    {
        static void Main(string[] args)
        {
            DB _db = new DB(0, "data source=localhost; database=testingdb; UID=root;
       PWD=123456");
            DBAdapter _adapter = _db.GetAdapter();

            DataTable _dt = new DataTable();
            _adapter.Fill(_dt);
            Console.WriteLine("Our Users ID/Name/Sirname:");

            for(int i=0; i<_dt.Rows.Count;i++)
            {
                Console.WriteLine("{0}/{1}/{2}",
                    (int)_dt.Rows[i][0],
                    (string) _dt.Rows[i][1],
                    (string) _dt.Rows[i][2]);
            }

            _db.Dispose();
        }
    }
}
```