Finisar.SQLite

- An ADO.NET Data Provider for SQLite

C# SourceCode Example

This Example shows the basic things when working with SQLite.NET.
Basically this is how all ADO.NET Data Providers work ;D

Before you can use Finisar's SQLite.NET Data Provider, you have to tell your IDE about it.
This is how it is done in "Microsoft Visual C# 2005 Express Edition Beta 2" - It should not differ much in other IDEs...

1) Place the .dll files you downloaded from this website into your projects binary folder.
2) Press 'Project -> Add Reference...' in the menu bar.
3) Select the 'Browse'-Tab and select the "SQLite.dll".
4) Press 'Ok' and you are finishd ;D

C# SourceCode:

```
using Finisar.SQLite;

// [snip] - As C# is purely object-oriented the following lines must be put into a class:

// We use these three SQLite objects:
SQLiteConnection sqlite_conn;
SQLiteCommand sqlite_cmd;
SQLiteDataReader sqlite_datareader;

// create a new database connection:
sqlite_conn = new SQLiteConnection("Data
Source=database.db;Version=3;New=True;Compress=True;");

// open the connection:
sqlite_conn.Open();

// create a new SQL command:
sqlite_cmd = sqlite_conn.CreateCommand();

// Let the SQLiteCommand object know our SQL-Query:
sqlite_cmd.CommandText = "CREATE TABLE test (id integer primary key, text
varchar(100));";

// Now lets execute the SQL ;D
sqlite_cmd.ExecuteNonQuery();

// Lets insert something into our new table:
sqlite_cmd.CommandText = "INSERT INTO test (id, text) VALUES (1, 'Test Text 1');";

// And execute this again ;D
sqlite_cmd.ExecuteNonQuery();

// ...and inserting another line:
sqlite_cmd.CommandText = "INSERT INTO test (id, text) VALUES (2, 'Test Text 2');";

// And execute this again ;D
```

```
sqlite_cmd.ExecuteNonQuery();

// But how do we read something out of our table ?
// First lets build a SQL-Query again:
sqlite_cmd.CommandText = "SELECT * FROM test";

// Now the SQLiteCommand object can give us a DataReader-Object:
sqlite_datareader=sqlite_cmd.ExecuteReader();

// The SQLiteDataReader allows us to run through the result lines:
while (sqlite_datareader.Read()) // Read() returns true if there is still a result
line to read
{
// Print out the content of the text field:
System.Console.WriteLine( sqlite_datareader["text"] );
}

// We are ready, now lets cleanup and close our connection:
sqlite_conn.Close();
```