**Setting up Visual Studio to work with Oracle**

I will use Visual Studio 2013 (Professional) and set up Oracle. You can get instructions from <http://www.oracle.com/technetwork/developer-tools/visual-studio/overview/index.html>

Basically, you need to install ODAC (Oracle Data Access Components) for Oracle 11.1.0.7 or higher. I actually used **ODAC 12c Release 2 and Oracle Developer Tools for Visual Studio (12.1.0.1.2).** I found this easier to use to connect to Oracle DB server than the Oracle 11g ODAC. During installation, it will ask you to execute SQL scripts, you can ignore that message.

Also, during installation, it will ask for the database server details. Make sure you specify the correct specifications, which include: HOST = db.csep.umflint.edu; PORT = 1521; SERVICE\_NAME = csep

This will automatically set up the tnsnames.ors file as follows.

Note that ORACLE\_HOME is where ODAC is installed. It will typically be in a location such as c:\app\mmani\product\12.1.0\client\_1

The tnsnames.ora file will be set up at ORACLE\_HOME\Network\Admin\

The fill will say something like (after correct setup):

myFlintOracleDB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = db.csep.umflint.edu)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = csep)

)

)

See that myFlintOracleDB is just the name that I gave to this data source.

Now after that see that you can connect to the Oracle DB on UMFlint (using Server Explorer in Visual Studio, and choose Data Connections), for this follow the instructions in the link below, and choose the myFlintOracleDB as the data source. You must see that in your list of data sources.

See that you can see the tables as you can see using SQLDeveloper.

To build .NET applications, check

<http://www.oracle.com/technetwork/articles/dotnet/vs2010-oracle-dev-410461.html>

Let us create a new project (Windows Forms Application C# for illustration).

Add a new Data Source (let us add UMFlintOracleDB), I will actually add only 1 table.

You can add it to your form, and follow the instructions given above.

For advanced programming, you may want to research on developing forms using .NET and crystal reports (<http://www.codeproject.com/Articles/166291/Generate-a-report-using-Crystal-Reports-in-Visual> )

**A Windows Form Application (C#) with transactions**

Let us first create 2 tables and insert 1 row of data into the 2 tables as follows (in SQL Developer or using Visual Studio Query Window for the Oracle connection):

create table test1 (col1 int primary key, col2 int);

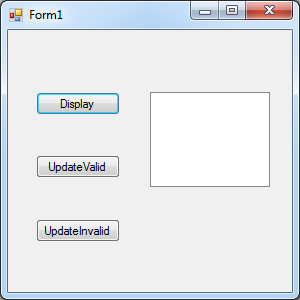
create table test2 (col3 int primary key, col4 int);

insert into test1 values (1, 11);

insert into test2 values (101, 1001);

commit;

Let me design a windows form application that looks like:



Here the IDs for the various Buttons are:

Display = button1; UpdateValid = button2; UpdateInvalid = button3

Also there is a ListBox whose ID is listBox1

The idea is: Display will display the rows in table test1 in the ListBox

UpdateValid will perform 1 transaction, that inserts 1 row into test1 and 1 row into test2

UpdateInvalid will perform 2 transactions – first transaction inserts 1 row into test1 and 1 row into test2. The insert into test2 will violate a primary key constraint, and therefore the whole transaction will fail. We must rollback the transaction when an exception happens. The second transaction will then insert 1 row into test1 and 1 row into test2, and both will succeed.

For this, first, we need to add a reference to the Oracle Data Provider (see

The code-behind file looks like the following: <http://www.oracle.com/technetwork/articles/dotnet/vs2010-oracle-dev-410461.html>

Make sure that the Reference added is Oracle.DataAccess – for this, I had to right click on References, Add Reference, Extensions, Oracle.DataAccess 4.121.1.0

If the reference is not added, the using statement in the code-behind file that says: using Oracle.DataAccess.Client will give an error.

My code-behind looks like the following. The highlighted portion was provided automatically.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Oracle.DataAccess.Client;

namespace WindowsFormsApplication3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string oradb = "Data Source=(DESCRIPTION=(ADDRESS\_LIST="

+ "(ADDRESS=(PROTOCOL=TCP)(HOST=db.csep.umflint.edu)(PORT=1521)))"

+ "(CONNECT\_DATA=(SERVER=DEDICATED)(SERVICE\_NAME=csep)));"

+ "User Id=mmani;Password=mmani;";

OracleConnection conn = new OracleConnection(oradb); // C#

conn.Open();

OracleCommand cmd = new OracleCommand();

cmd.Connection = conn;

cmd.CommandText =

"select \* from test1";

cmd.CommandType = CommandType.Text;

OracleDataReader dr = cmd.ExecuteReader();

listBox1.Items.Clear();

while (dr.Read())

{

listBox1.Items.Add("col1 = " + dr["col1"].ToString() +

" col2 = " + dr["col2"].ToString());

}

dr.Dispose();

cmd.Dispose();

conn.Dispose();

}

private void button2\_Click(object sender, EventArgs e)

{

string oradb = "Data Source=(DESCRIPTION=(ADDRESS\_LIST="

+ "(ADDRESS=(PROTOCOL=TCP)(HOST=db.csep.umflint.edu)(PORT=1521)))"

+ "(CONNECT\_DATA=(SERVER=DEDICATED)(SERVICE\_NAME=csep)));"

+ "User Id=mmani;Password=mmani;";

OracleConnection conn = new OracleConnection(oradb); // C#

conn.Open();

OracleCommand cmd = new OracleCommand();

cmd.Connection = conn;

// Start a transaction

OracleTransaction txn = conn.BeginTransaction(

IsolationLevel.ReadCommitted);

try

{

// Insert the same row twice into MyTable

cmd.CommandText = "INSERT INTO test1 VALUES (2, 12)";

cmd.ExecuteNonQuery();

cmd.CommandText = "INSERT INTO test2 VALUES (102, 1002)";

cmd.ExecuteNonQuery(); // This may throw an exception

txn.Commit();

}

catch (Exception ex)

{

// Print the exception message

Console.WriteLine("ex.Message = " + ex.Message);

// Rollback the transaction

txn.Rollback();

}

txn.Dispose();

cmd.Dispose();

conn.Dispose();

}

private void button3\_Click(object sender, EventArgs e)

{

string oradb = "Data Source=(DESCRIPTION=(ADDRESS\_LIST="

+ "(ADDRESS=(PROTOCOL=TCP)(HOST=db.csep.umflint.edu)(PORT=1521)))"

+ "(CONNECT\_DATA=(SERVER=DEDICATED)(SERVICE\_NAME=csep)));"

+ "User Id=mmani;Password=mmani;";

OracleConnection conn = new OracleConnection(oradb); // C#

conn.Open();

OracleCommand cmd = new OracleCommand();

cmd.Connection = conn;

// Start a transaction

OracleTransaction txn = conn.BeginTransaction(

IsolationLevel.ReadCommitted);

try

{

// Insert the same row twice into MyTable

cmd.CommandText = "INSERT INTO test1 VALUES (3, 13)";

cmd.ExecuteNonQuery();

cmd.CommandText = "INSERT INTO test2 VALUES (102, 1002)";

cmd.ExecuteNonQuery(); // This may throw an exception

txn.Commit();

}

catch (Exception ex)

{

// Print the exception message

Console.WriteLine("ex.Message = " + ex.Message);

// Rollback the transaction

txn.Rollback();

}

txn = conn.BeginTransaction(IsolationLevel.ReadCommitted);

try

{

// Insert the same row twice into MyTable

cmd.CommandText = "INSERT INTO test1 VALUES (3, 15)";

cmd.ExecuteNonQuery();

cmd.CommandText = "INSERT INTO test2 VALUES (103, 1005)";

cmd.ExecuteNonQuery(); // This may throw an exception

txn.Commit();

}

catch (Exception ex)

{

// Print the exception message

Console.WriteLine("ex.Message = " + ex.Message);

// Rollback the transaction

txn.Rollback();

}

txn.Dispose();

cmd.Dispose();

conn.Dispose();

}

}

}

**ASP.NET applications**

If you want to build an ASP.NET application, and use the IIS (Internet Information Services) Web server, the following links may provide useful instructions.

<http://digitizor.com/2009/02/20/how-to-install-microsoft-iis-server-on-windows-7/>

<http://technet.microsoft.com/en-us/library/cc725762.aspx>

<http://msdn.microsoft.com/en-us/library/a1zz9df4.aspx>

<http://docs.oracle.com/cd/B28359_01/appdev.111/b28844/using_aspnt.htm>

<http://msdn.microsoft.com/en-us/library/k4cbh4dh.aspx>

Some helpful tips:

You can start/stop/restart manage your IIS server by running inetmgr.

you run into difficulties, check your inetmgr and see that your ASP.NET is started.