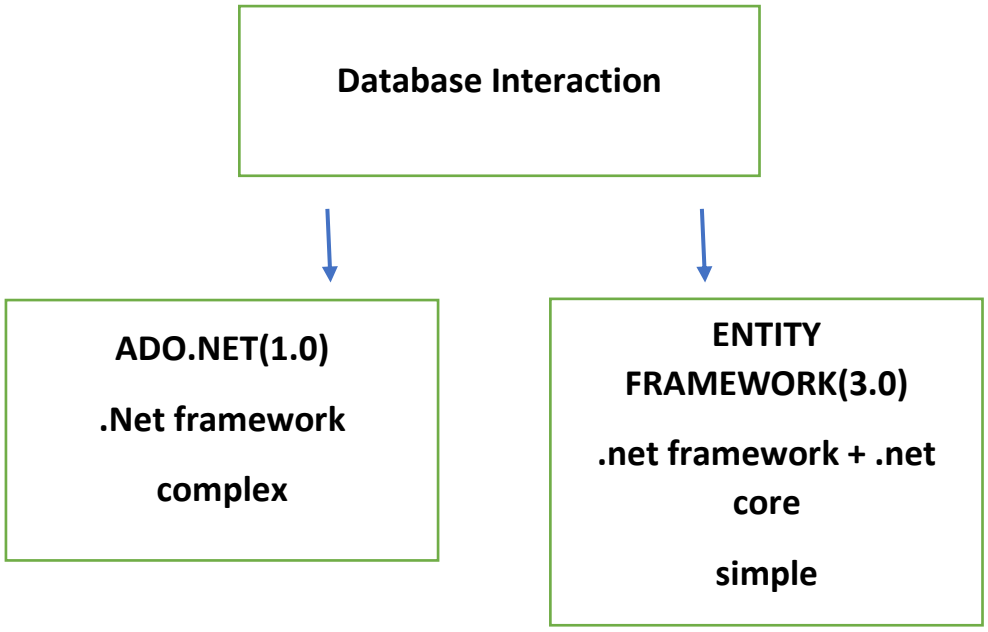
Database Programming



Ado.Net

Topics

- Introduction to Ado.Net
- Understanding connected architecture
- Understanding disconnected architecture

What is Ado.Net?

Ado.Net is a set of Framework classes which allows to interact with Data sources

Stands for **ActiveX data object**

Data Sources includes following

1. Sql Server
2. Oracle
3. MySQL
4. Access
5. Excel
6. Xml Files etc...

Windows

Console

Web

Framework classes

Namespaces in Ado.Net

- System.Data.Sqlclient -> sql server
- System.data.oracleclient -> oracle
- System.Data.oledb -> access, mysql, excel, file,sql server, oracle

(object linking embedding)

- System.Data.Sqlclient (fast + preconfigured)
  - Sql Server only 7 >
- System.Data.Oracleclient
  - Oracle only 8 >
- System.Data.Oledb(slow+ manual configuration)
  - Meant to connect , sql + oracle + Access , Excel, MySql

Models in Ado.Net

System.Data.Sqlclient -> sql server

1. Connected Architecture (online)
2. Disconnected architecture (offline)

Connected Architecture

To Perform CRUD operation, Database connection has to exists

Disconnected Architecture

All Operations are performed offline, all offline changes are updated as a bulk to database

Classes used in connected architecture

1. SqlConnection
2. SqlCommand
3. SqlDataReader

Sql Connection Class

To interact with a database, you must have a connection to it. The connection helps identify the database server, the database name, user name, password, and other parameters that are required for connecting to the data base

Example

Integrated Security : connects to Sql server using windows authenticaiton

SqlConnection con = new SqlConnection("Integrated security=true;server= ANANDPK\SQLEXPRESS;database =master");

SqlConnection con = new SqlConnection("uid=sa;pwd=india;server=remotecomputername;database =master");

Sql Command Class

The process of interacting with a database means that you must specify the actions you want to occur. This is done with a command object. You use a command object to send SQL statements to the database. A command object uses a connection object to figure out which database to communicate with.

Example

Sqlcommand cmd = new sqlcommand (“select * from customer”,con)

To send the query to sql server u need to call execute method of command class

1. Int I = cmd.ExecutenonQuery();
2. SqlDataReader dr = cmd.ExecuteReader();
3. XMLDataReader dr = cmd.ExecuteXMLReader();
4. Object ob = cmd.Executescalar();

Methods in Command class

ExecuteNonQuery : returns integer value (DDL,DML)

ExecuteReader : if query returns a records(rows) , (Select)

ExecuteXmlReader : if query returns data xml format

ExecuteScalar : if query returns single value

Reads a data from a table. (first row ,first column value)

SqlDataReader

The data reader object allows you to obtain the results of a SELECT statement from a command object

Steps in connected Data Architecture

Imports namespace

Create connection object

Open connection

Create command object

Execute Command

Extract the Results

Close the connection

Transactions

Transactions allow you to combine multiple operations into a single unit of work.

If a failure occurs at one point in the transaction, all of the updates can be rolled back to their pre-transaction state.

In ADO.NET, you can control transactions using the Connection and Transaction objects.

You can initiate a local transaction using BeginTransaction statement.

Connection.BeginTransaction.

Once you have begun a transaction, you can enlist a command in that transaction using the Transaction property of the Command object.

new SqlCommand("Your SQL Statemnt Here", Connection, transaction).ExecuteNonQuery();

You can then use the Transaction object to commit or rollback modifications made at the data source based on the success or failure of the components of the transaction.

transaction.Commit();

Disconnected Architecture

All CRUD operations are performed offline

Classes used in Disconnected

1. SqlConnection
2. SqlDataAdaptor
3. DataSet : can store many table
4. DataTable : can store single table
5. DataRow : can store 1 row
6. DataColumn : can store 1 column

SqlDataAdaptor

The SqlDataAdapter holds the SQL commands and connection object for reading and writing data. You initialize it with a SQL select statement and connection object:

Sqlcommand cmd = new sqlcommand (“select * from customer”,con)

SqlDataAdaptor cmd = new SqlDataAdaptor

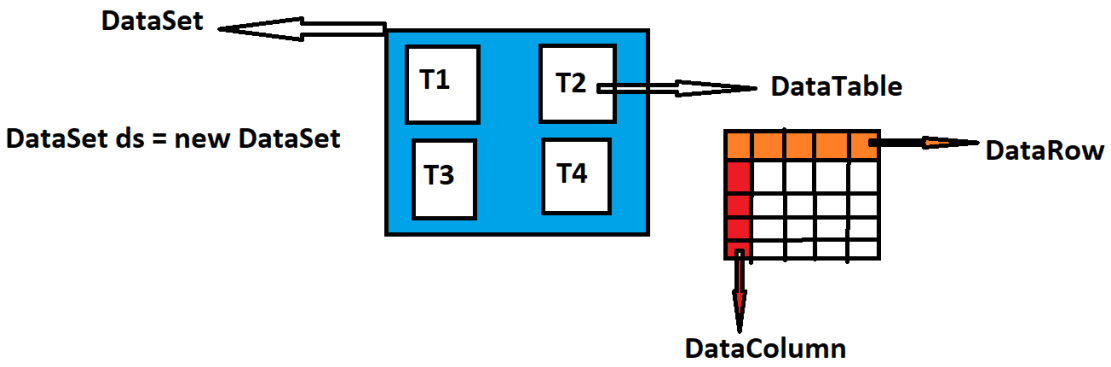
(“select * from customer”,con)

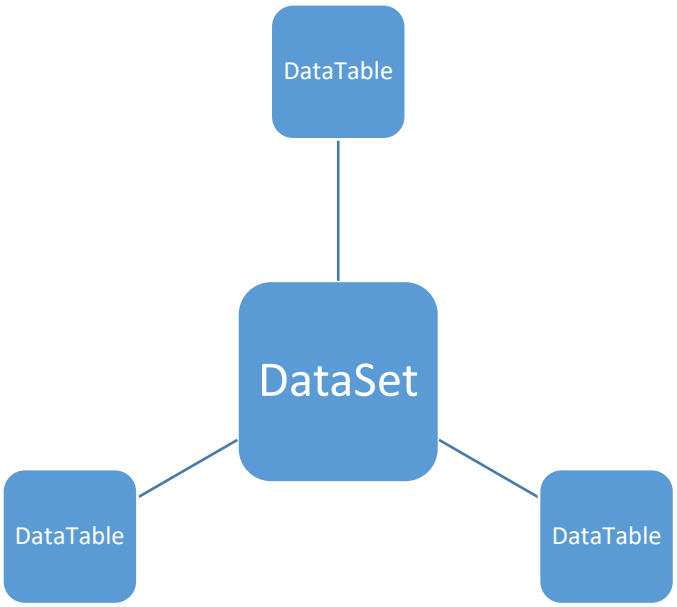
Dataset:
A Dataset is an in-memory representation of data. It can hold multiple tables (DataTable), relationships between them, and constraints. It is used to manage data retrieved from a data source and can work with disconnected data.

DataTable:
A DataTable is a single table of in-memory data. It contains rows (DataRow) and columns (DataColumn). It represents data from a database or any other data source and can be part of a Dataset.

DataRow:
A DataRow represents a single row in a DataTable. It contains data for each column in the table and allows you to access, modify, and update that data.

DataColumn:
A DataColumn represents a single column in a DataTable. It defines the structure of a column (like the data type and name) and contains information about that column





Difference Between

SqlDataReader	DataSet
1. Connection is required to read data	Not Required
2. Read-only	Read/Write
3. Forward only direction	forward/backward
4. Single Table	Multiple Tables
5. Is database specific class	Shared class

SqlCommandBuilder :

This class will automatically generate Insert, Update, Delete command (queries) based upon the changes happened in dataset.

Dt.Rows.Add(100,”dhoni”,40, “ranchi”);

Dt.rows.delete(10);

SqlCommandBuilder object converts to following output

Insert into customer values (100,”dhoni”,40, “ranchi”);

Delete from customer where cid=10

DataSet

DataSet objects are in-memory representations of data. They contain multiple Datatable objects, which contain columns and rows, just like normal database tables. You can even define relations between tables to create parent-child relationships. The DataSet is specifically designed to help manage data in memory and to support disconnected operations on data, when such a scenario make sense. The DataSet is an object that is used by all of the Data Providers, which is why it does not have a Data Provider specific prefix.