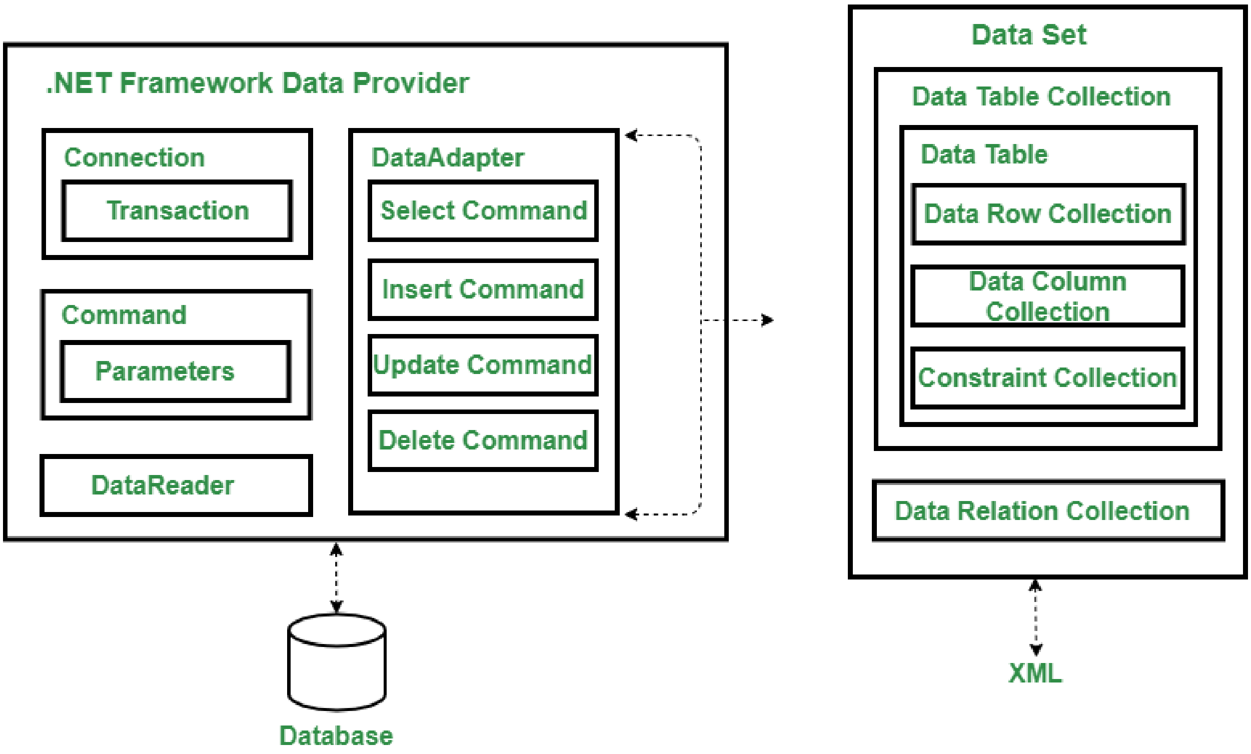
ADO.NET (Active Data Objects for .NET)

ADO.NET is an object-oriented set of libraries that allows you to interact with data sources. Commonly, the data source is a database, but it could also be a text file, an Excel spread-sheet, or an XML file. There are many different types of databases available such as Microsoft SQL Server, Microsoft Access, Oracle, Borland Interbase, IBM DB2, etc.

Connected & Disconnected Data (Architecture)

The data access with ADO.NET consists of two parts:

(1) Data Provider (2) DataSet



The Connection Object

The Connection object creates the connection to the database. Microsoft Visual Studio .NET provides two types of Connection classes: the SqlConnection object, which is designed specifically to connect to Microsoft SQL Server 7.0 or later, and the OleDbConnection object, which can provide connections to a wide range of database types like Microsoft Access and Oracle. The Connection object contains all of the information required to open a connection to the database.

Connection class exist for ODBC (OdbcConnection), OLE DB (OledbConnection), SQL Server (SqlConnection) & Oracle (OracleConnection)

Property :

- *ConnectionString* // holds the string used to open a database

- *ConnectionTimeout*
- *DataSource* // This property gets the name of the server providing the data.
- *State*
- *ServerVersion*
- *Provider*

Methods :

- *Open(), Close(), BeginTransaction()*

For Details Click here

The Command Object

The Command object is represented by two corresponding classes: SqlCommand and OleDbCommand. Command objects are used to execute commands to a database across a data connection. The Command objects can be used to execute stored procedures on the database, SQL commands, or return complete tables directly.

Command objects provide three methods that are used to execute commands on the database:

***ExecuteNonQuery**: Executes commands that have no return values such as INSERT, UPDATE or DELETE*

***ExecuteScalar**: Returns a single value from a database query*

***ExecuteReader**: Returns a result set by way of a DataReader object*

Properties :

- CommandText
- CommandType
- Connection
- Parameters

For Details Click here

The DataAdapter Object

The DataAdapter is the class at the core of ADO .NET's disconnected data access. It is essentially the middleman facilitating all communication between the database and a DataSet. The DataAdapter is used either to fill a DataTable or DataSet with data from the database with it's Fill method. After the memory-resident data has been manipulated, the DataAdapter can commit the changes to the database by calling the Update method.

***Methods** :*

- *Fill*
- *FillSchema*
- *Update*
- *FillAsync*
- *ResetFillLoadOption*

Properties :

- SelectCommand
- InsertCommand
- UpdateCommand
- DeleteCommand

DataSet

The dataset is a disconnected, in-memory representation of data. It can be considered as a local copy of the relevant portions of the database. The DataSet is persisted in memory and the data in it can be manipulated and updated independent of the database. When the use of this DataSet is finished, changes can be made back to the central database for updating.

Property :

(1) Relations : It is the collection of DataRelation objects, which defines the relationship of the DataTables within the dataset.

(2) Tables : It is the collection of DataTables contained in the dataset.

Method :

- *AcceptChanges* //Commits all the pending changes made to the dataset.
- *Clear*
- *Clone*
- *Copy*
- *GetChanges*
- *Reset*
- *RejectChanges* // Rolls back the changes made to the dataset since it was created or since the AcceptChanges method was called

DataReader:

A DataReader is a forward-only, read-only cursor used to retrieve data from a data source.It provides a fast, efficient way to read data from a database one record at a time. DataReader retrieves data on demand from the data source.

```
datareader = cmd.ExecuteReader()
While datareader.Read
   MsgBox(datareader(Name))
End While
```

Properties:

- *Item*[*index or nam*e]
- *FieldCount* //Get the number of columns in the current row
- *IsClosed* // Check if the reader is closed
- *HasRows* // Check if the reader has rows
- RecordsAffected //returns the number of rows affected by the last SQL statement

Methods:

- *Read()*
- *Close()*
- GetName(Index)

- GetValue(index of column)
- IsDBNull(index)
- *Dispose()*

Data Provider :

A DataProvider is a set of related components that work together to provide data in an efficient and performance driven manner. The .NET Framework currently comes with Four DataProviders:

- *SQL Data Provider,*
- *ODBC Data Provider,*
- *OLEDB Data Provider (object linking and embedding database)*
- *Oracle Data Provider*

Component :

- *Connection*
- *Command*
- *DataReader*
- *DataAdapter*