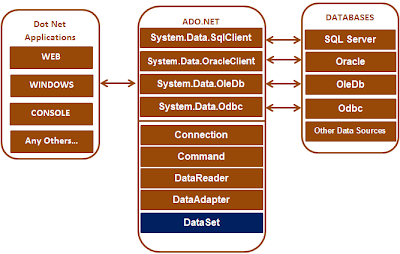
**What is ADO.NET?**  
**ADO.NET is not a different technology.** In simple terms, you can think of ADO.NET, as a set of classes (Framework), that can be used to interact with data sources like Databases and XML files. This data can, then be consumed in any .NET application. ADO stands for Microsoft ActiveX Data Objects.   
  
The following are, a few of the different types of .NET applications that use ADO.NET to connect to a database, execute commands, and retrieve data.  
**ASP.NET Web Applications**  
**Windows Applications**  
**Console Applications**  
  
**What are .NET Data Providers?**  
Databases only understand SQL. If a .NET application (Web, Windows, Console etc..) has to retrieve data, then the application needs to  
**1.** Connect to the Database  
**2.** Prepare an SQL Command  
**3.** Execute the Command  
**4.** Retrieve the results and display in the application   
  
   
  
**Sample ADO.NET code to connect to SQL Server Database and retrieve data**. Notice that we are using **SQLConnection**, **SQLCommand**and **SQLDataReader**classes . All the objects are prefixed with the word **SQL**. All these classes are present in **System.Data.SqlClient** namespace. So, we can say that the**.NET data provider for SQL Server is System.Data.SqlClient.**  
SqlConnection con = new SqlConnection("data source=.; database=Sample; integrated security=SSPI");  
SqlCommand cmd = new SqlCommand("Select \* from tblProduct", con);  
con.Open();  
SqlDataReader rdr = cmd.ExecuteReader();  
GridView1.DataSource = rdr;  
GridView1.DataBind();  
con.Close();  
  
**Sample ADO.NET code to connect to Oracle Database and retrieve data.** Notice that we are using **OracleConnection**, **OracleCommand** and **OracleDataReader**classes . All the objects are prefixed with the word **Oracle**. All these classes are present in System.Data.OracleClient namespace. So, we can say that the **.NET data provider for Oracle is System.Data.OracleClient.**  
OracleConnection con = new OracleConnection("Oracle Database Connection String");  
OracleCommand cmd = new OracleCommand("Select \* from tblProduct", con);  
con.Open();  
OracleDataReader rdr = cmd.ExecuteReader();  
GridView1.DataSource = rdr;  
GridView1.DataBind();  
con.Close();   
  
   
  
**If we want to connect to OLEDB datasources like Excel, Access** etc, we can use **OleDbConnection**, **OleDbCommand** and **OleDbDataReader** classes. So, .NET data provider for OLEDB is System.Data.OleDb.   
  
**Different .NET Data Providers**  
Data Provider for SQL Server - System.Data.SqlClient  
Data Provider for Oracle - System.Data.OracleClient  
Data Provider for OLEDB - System.Data.OleDb  
Data Provider for ODBC - System.Data.Odbc   
  
   
  
**Please note that, depending on the provider**, the following ADO.NET objects have a different prefix  
1. Connection - SQLConnection, OracleConnection, OleDbConnection, OdbcConnection etc  
2. Command - SQLCommand, OracleCommand, OleDbCommand, OdbcCommand etc  
3. DataReader - SQLDataReader, OracleDataReader, OleDbDataReader, OdbcDataReader etc  
4. DataAdapter - SQLDataAdapter, OracleDataAdapter, OleDbDataAdapter, OdbcDataAdapter etc  
  
**The DataSet object is not provider specific**. Once we connect to a Database, execute command, and retrieve data into .NET application. The data can then be stored in a DataSet and work independently of the database.