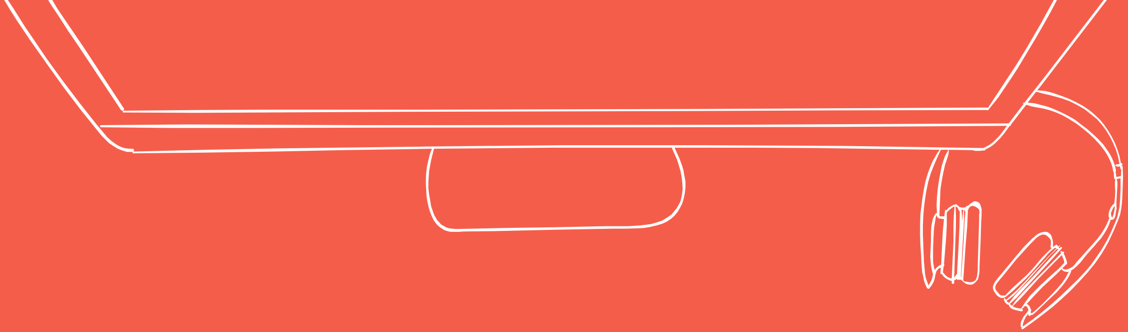
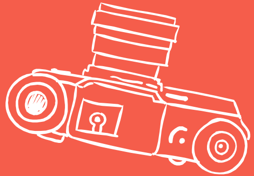
.NET Technologies





.NET Technologies - Lecture 11

Presented by: Drashti Shrimal

UNIT 3: Topics in this presentation

- ADO.NET Introduction
- Connection Architectures
- Diagramatic view
- Basic Code in Detail

ADO.NET

- ADO.NET is a set of classes (a framework) to interact with data sources such as databases and XML files. ADO is the acronym for ActiveX Data Objects. It allows us to connect to underlying data or databases. It has classes and methods to retrieve and manipulate data.
- The following are a few of the .NET applications that use ADO.NET to connect to a database, execute commands and retrieve data from the database.

a. ASP.NET Web Applications

b. Console Applications

c. Windows Applications.

Connection Architectures

- Ado.net is both connection-oriented as well as disconnection oriented. Depending upon the functionality of an application, we can make it connection-oriented or disconnection oriented.

1. Connected Architecture

- As the name suggests, connected architecture refers to the fact that the connection is established for the full time between the database and application. For e.g. we make a program in C# that is connected with the database for the full time, so that will be connected architecture.

Connection Architectures

- Connected architecture is forward only and read-only. This means the connected mode will work only in one particular direction i.e. forward and that too for read-only purpose. Application issues query then read back results and process them.
- For connected architecture, we mainly use the object of the DataReader class.
- DataReader is used to retrieve the data from the database and it also ensures that the connection is maintained for the complete interval of time.
- In connected architecture, the application is directly linked with the Database.

Connection Architectures

2. Disconnected architecture

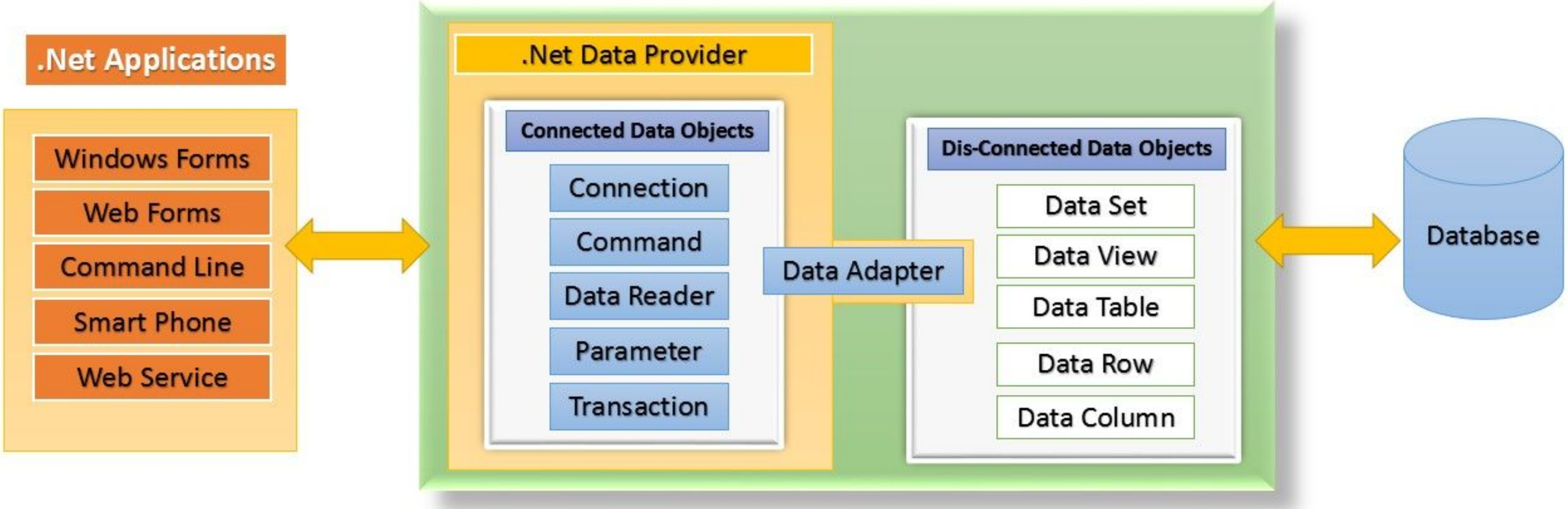
- It refers to the mode of architecture in Ado.net where the connectivity between the database and application is not maintained for the full time. Connectivity within this mode is established only to read the data from the database and finally to update the data within the database.

- This means during the processing of the application, we need data so that data is fetched from the database and kept in temporary tables. After that whenever data is required, it is fetched from the temporary tables. And finally, when the operations were completed, the connection was established to update the data within the database from the temporary tables.

Connection Architectures

- In this mode, application issues query then retrieves and store results for processing. For this purpose, we use objects of SqlDataAdapter and DataSet classes.
- In disconnected architecture, a Dataset is used for retrieving data from the database. This way there is no need to establish a connection for the full time because DataSet acts as temporary storage. All the operations can be performed on the data using the Dataset and finally modified at the database.

ADO.NET Architecture Component



.Net Applications

- Windows Forms
- Web Forms
- Command Line
- Smart Phone
- Web Service

.Net Data Provider

Connected Data Objects

- Connection
- Command
- Data Reader
- Parameter
- Transaction

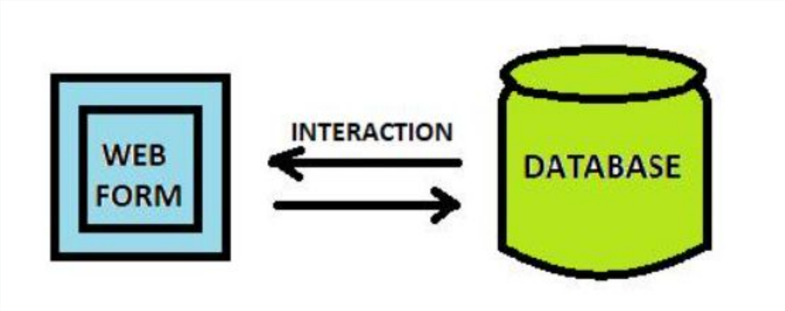
Data Adapter

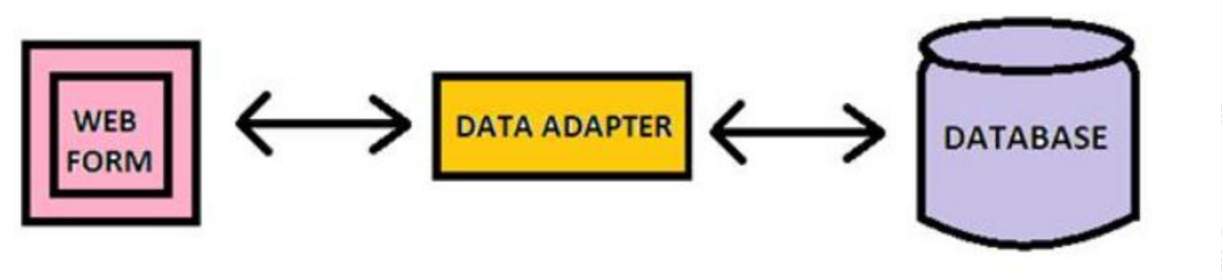
Dis-Connected Data Objects

- Data Set
- Data View
- Data Table
- Data Row
- Data Column

Database

Connection Architectures





Important Terms

- DataReader: DataReader used to read the data from the source.
- Dataset: DataSet contains the table and relation.
- DataAdapter: DataAdapter behaves as a mediator between the back end and front end. But it does not have features to contains the data. So there is a dataset that contains the data of the result set.
- Connection: Connects the source to web page.
- Command: Used to fire query.

Basic Code

```
string path = "C:\\Users\\drash\\Desktop\\Data.xls";
```

- Stores the location of excel file in variable. Use two slashes, because compiler discards 1st slash.

```
string con = @"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" + path + ";
Extended Properties=\"Excel 8.0;HDR=Yes;IMEX=1\";";
```

- Provider = name of company to which database belongs to, along with its driver and version. Data source - uses link of file to extract data. Extended properties - for extra config settings, HDR - Yes if the first row of table is column header, else No. IMEX - data description, it can be 0,1 or 2.

```
 string query = "Select * from [Sheet1$]";
```

- Query to be fired is stored in a variable.

Basic Code

```
OleDbConnection conn = new OleDbConnection(con);
```

- Represents an open connection to a data source. Creates a connection to the database

```
OleDbCommand cmd = new OleDbCommand(query, conn);
```

- Represents an SQL statement or stored procedure to execute against a data source. To fire a query, we need to create this object. Takes two parameters - the query and OLEDBConnection.

Basic Code

```
OleDbDataAdapter da = new OleDbDataAdapter(cmd);
```

- Represents a set of data commands and a database connection that are used to fill the DataSet and update the data source. Works like a bridge that transfers data from one part to other//Dataset is a type which stores databases.

```
DataSet ds = new DataSet();
```

- DataSet is tabular representation of data. Tabular representation means it represents data into row and column format. This class is counted in a disconnected architecture in . NET Framework.

```
da.Fill(ds);
```

-Adds or refreshes rows in the DataSet to match those in the data source .

Basic Code

```
g1.DataSource = ds.Tables[0];
```

- Its a property of grid view. Dataset is combination of tables. So here the datasource assigns the dataset to grid view.

```
g1.DataBind();
```

- Binding means tighten data from datasource to grid view.

```
conn.Close();
```

- Close the connection for security.

Note: For full example please refer to the program of practical no. 6