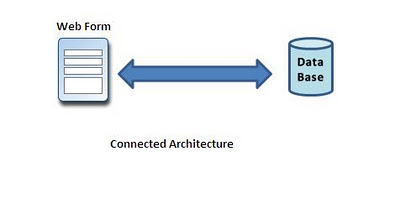
**Connected Architecture of ADO.NET**

The architecture of ADO.net, in which connection must be opened to access the data retrieved from database is called as connected architecture. Connected architecture was built on the classes **connection, command, datareader,** and **transaction.**

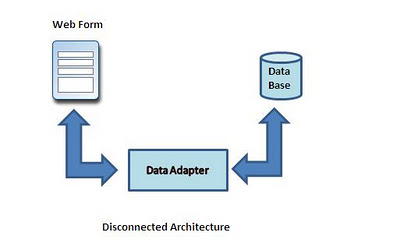
Connected architecture is when you constantly make trips to the database for any CRUD (Create, Read, Update and Delete) operation you wish to do. This creates more traffic to the database but is normally much faster as you should be doing smaller transactions.



**Disconnected Architecture in ADO.NET**

The architecture of ADO.net in which data retrieved from database can be accessed even when connection to database was closed is called as disconnected architecture. Disconnected architecture of ADO.net was built on classes **connection, dataadapter, commandbuilder, dataset** and **dataview**.

Disconnected architecture is a method of retrieving a record set from the database and storing it giving you the ability to do many CRUD (Create, Read, Update and Delete) operations on the data in memory, then it can be re-synchronized with the database when reconnecting. A method of using disconnected architecture is using a Dataset.



**DataReader** is **Connected** Architecture since it keeps the connection open until all rows are fetched one by one

**DataSet** is **DisConnected** Architecture since all the records are brought at once and there is no need to keep the connection alive

Difference between Connected and disconnected architecture

|  |  |
| --- | --- |
| **Connected** | **Disconnected** |
| It is connection oriented. | It is dis\_connection oriented. |
| Datareader | DataSet |
| Connected methods gives faster performance | Disconnected get low in speed and performance. |
| connected can hold the data of single table | disconnected can hold multiple tables of data |
| connected you need to use a read only forward only data reader | disconnected you cannot |
| Data Reader can't persist the data | Data Set can persist the data |
| It is Read only, we can't update the data. | We can update data |

**Example**

Create Database “Student”

CREATE TABLE [dbo].[Student]

(

    [ID] [int] PRIMARY KEY IDENTITY(1,1) NOT NULL,

    [Name] [varchar](255) NULL,

    [Age] [int] NULL,

    [Address] [varchar](255) NULL

)

INSERT INTO Student([Name],[Age],[Address])VALUES('NAME 1','22','PUNE')

INSERT INTO Student([Name],[Age],[Address])VALUES('NAME 2','25','MUMBAI')

INSERT INTO Student([Name],[Age],[Address])VALUES('NAME 3','23','PUNE')

INSERT INTO Student([Name],[Age],[Address])VALUES('NAME 4','21','DELHI')

INSERT INTO Student([Name],[Age],[Address])VALUES('NAME 5','22','PUNE')

**HTML**

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

    <title>Untitled Pagetitle>

</head>

<body>

    <form id="form1" runat="server">

    <div>

        <asp:GridView ID="GridView1" runat="server" BackColor="White"BorderColor="#CC9966"

            BorderStyle="None" BorderWidth="1px" CellPadding="4">

            <FooterStyle BackColor="#FFFFCC" ForeColor="#330099" />

            <RowStyle BackColor="White" ForeColor="#330099" />

            <PagerStyle BackColor="#FFFFCC" ForeColor="#330099" HorizontalAlign="Center"/>

            <SelectedRowStyle BackColor="#FFCC66" Font-Bold="True" ForeColor="#663399"/>

            <HeaderStyle BackColor="#990000" Font-Bold="True" ForeColor="#FFFFCC" />

        </asp:GridView>

        <br />

        <asp:Button ID="Connected" runat="server" OnClick="Connected\_Click"Text="Connected" />

        <asp:Button ID="Disconnected" runat="server" EnableTheming="False"OnClick="Disconnected\_Click"

            Text="Disconnected" />

    </div>

    </form>

</body>

</html>

**Code Behind**

String StrSQL = "", StrConnection = "";

        protected void Page\_Load(object sender, EventArgs e)

        {

            StrSQL = "SELECT \* FROM Student";

            StrConnection = "Data Source=ServerName;Initial Catalog=Database;User ID=Username;Password=password";

        }

        protected void Connected\_Click(object sender, EventArgs e)

        {

            using (SqlConnection objConn = new SqlConnection(StrConnection))

            {

                SqlCommand objCmd = new SqlCommand(StrSQL, objConn);

                objCmd.CommandType = CommandType.Text;

**objConn.Open();**

                SqlDataReader objDr = objCmd.ExecuteReader();

                GridView1.DataSource = objDr;

                GridView1.DataBind();

**objConn.Close();**

            }

        }

        protected void Disconnected\_Click(object sender, EventArgs e)

        {

            SqlDataAdapter objDa = new SqlDataAdapter();

            DataSet objDs = new DataSet();

            using (SqlConnection objConn = new SqlConnection(StrConnection))

            {

                SqlCommand objCmd = new SqlCommand(StrSQL, objConn);

                objCmd.CommandType = CommandType.Text;

                objDa.SelectCommand = objCmd;

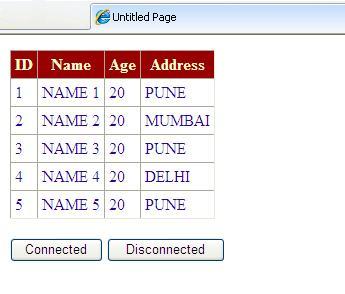
                objDa.Fill(objDs, "Student");

                GridView1.DataSource = objDs.Tables[0];

                GridView1.DataBind();

            }

        }

[](http://4.bp.blogspot.com/-wAA18nndNvU/TpgTZg8zyzI/AAAAAAAAAmM/4VXinPKB-dU/s1600/Connecteddisconneted.JPG)