

## 1. Working with MS-SQL-Server Database:

- MS-SQL-Server is the database which provides RDBMS facility.
- SQL is a standard language for accessing and manipulating databases.

### What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

### Using SQL in Your Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program (i.e. MS Access, SQL Server, MySQL)
- To use a server-side scripting language, like PHP or ASP
- To use SQL to get the data you want
- To use HTML / CSS

### Some of The Most Important SQL Commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

### SQL General Data Types

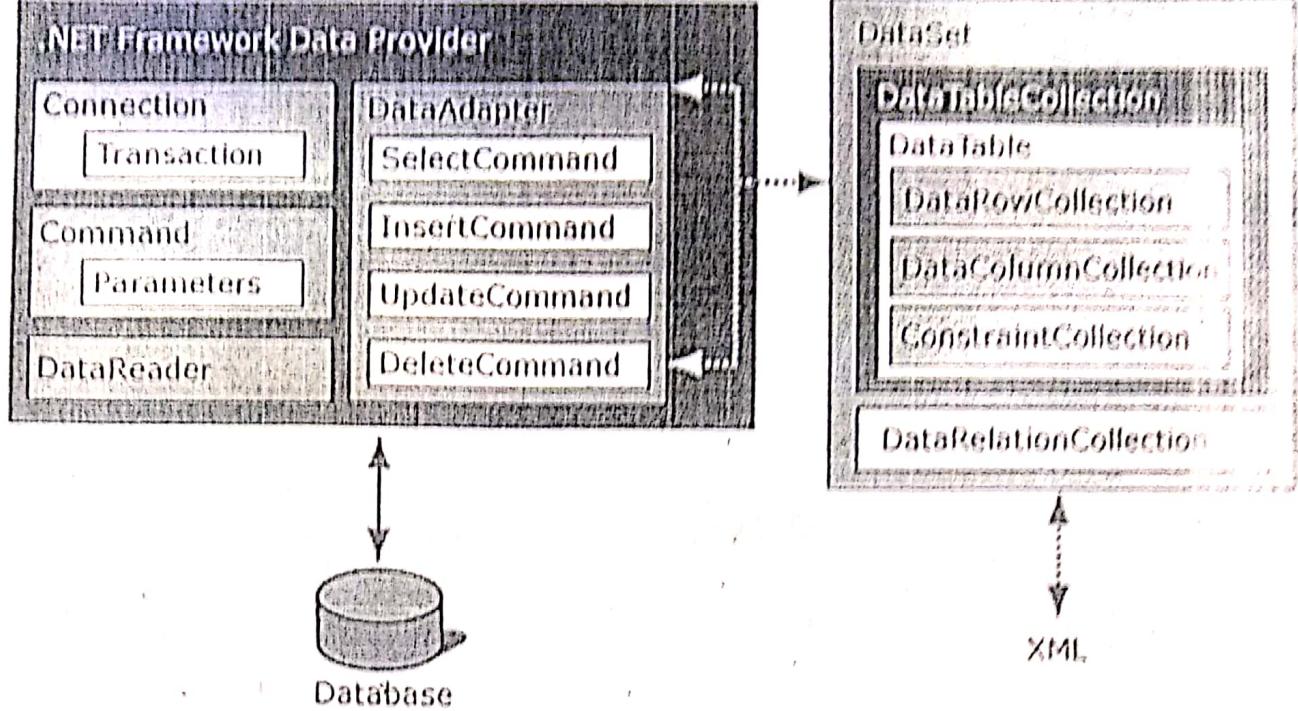
- Each column in a database table is required to have a name and a data type.
- SQL developers have to decide what types of data will be stored inside each and every table column when creating a SQL table. The data type is a label and a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data.

- The following table lists the general data types in SQL.

Data type	Description
CHARACTER(n)	Character string. Fixed-length n
VARCHAR(n) or CHARACTER VARYING(n)	Character string. Variable length. Maximum length n
BINARY(n)	Binary string. Fixed-length n
BOOLEAN	Stores TRUE or FALSE values
VARBINARY(n) or BINARY VARYING(n)	Binary string. Variable length. Maximum length n
INTEGER(p)	Integer numerical (no decimal). Precision p
SMALLINT	Integer numerical (no decimal). Precision 5
INTEGER	Integer numerical (no decimal). Precision 10
BIGINT	Integer numerical (no decimal). Precision 19
DECIMAL(p,s)	Exact numerical, precision p, scale s. Example: decimal(5,2) is a number that has 3 digits before the decimal and 2 digits after the decimal
NUMERIC(p,s)	Exact numerical, precision p, scale s. (Same as DECIMAL)
FLOAT(p)	Approximate numerical, mantissa precision p. A floating number in base 10 exponential notation. The size argument for this type consists of a single number specifying the minimum precision
REAL	Approximate numerical, mantissa precision 7
FLOAT	Approximate numerical, mantissa precision 16
DOUBLE PRECISION	Approximate numerical, mantissa precision 16
DATE	Stores year, month, and day values
TIME	Stores hour, minute, and second values
TIMESTAMP	Stores year, month, day, hour, minute, and second values
INTERVAL	Composed of a number of integer fields, representing a period of time, depending on the type of interval
ARRAY	A set-length and ordered collection of elements
MULTISET	A variable-length and unordered collection of elements
XML	Stores XML data

## 2. Working with ADO.NET

Data processing has traditionally relied primarily on a connection-based, two-tier model. As data processing increasingly uses multitier architectures, programmers are switching to a disconnected approach to provide better scalability for their applications.



## ADO.NET Components

There are two components of ADO.NET that you can use to access and manipulate data:

- .NET Framework data providers
- The Dataset

### a) .NET Framework Data Providers:

The providers help to bring data from database to our application and our application to database. There are mainly 3 data providers:

- **SQL Server:** Sql server data provider helps to connect with our application to Sql server database. This provider can connect only SQL Server database. To, connect with SQL server, we must use this data provider. This data provider uses **System.Data.SqlClient** name space.
- **OLEDB:** The full form of OLEDB is Object Linking and Embedded Database. Using this data provider, our application can connect to Ms Access and Oracle database. This data provider uses **System.Data.OleDb** name space.
- **ODBC:** The full form of ODBC is Open Database Connection. This data provider helps to connect with any database. For example: Ms Excel, Ms Access, Oracle, SQL server, etc. This data provider uses **System.Data.Odbc** name space.

**Each data provider provides:**

## **1. Connection:**

- The connection object provides the physical link between database and our application.
- For the connection with database, connection object requires security related information.
- The security related information is stored in the connection string.
- The connection object uses connection string to provide physical link between database and application.

## **2. Command:**

- The command object is used execute sql queries or stored procedure.
- To execute the query or stored procedure, this object requires open connection.
- The command object provides different methods to process the data in different fashion. For example, ExecuteNonQuery(), ExecuteScalar(), etc.

## **3. Data Reader:**

- Data reader provides the data read only and forward only.
- The data reader is a stream based which requires live connection with the data reader.
- The data reader uses the connected architecture.
- The data which is come from data reader cannot be modified.
- If the connection is closed from the server then it cannot be used for display.

## **4. Data Adapter:**

- Data Adapter uses the disconnected architecture.
- We can select, insert, update and delete records using data adapter.
- When any query, we want to execute at that time, we need open server connection. After that we can close the connection from the database.
- The data adapter does not require live connection.

- The data which is fetch by data adapter can be stored in Dataset or in Data table in our application.

A data provider contains Connection, Command, Data Adapter, and Data Reader objects. These four objects provide the functionality of Data Providers in the ADO.NET.

b) **Dataset:**

Dataset is the collection of the multiple data tables. This is a container control in our database. This control stores the data which is bring from the database or from XML. The data which is fetch from the Data Adapter object can be stored in the Dataset. The data which is stored in the XML can be retrieved in the Dataset. There can be a relationship between the multiple tables in the database. This relationship can be maintained in the dataset. So we can say that the dataset is the replica of the database. Dataset is one object can be used by `System.Data` name space.

### 3. Accessing Data with Server Explorer

- **How to create Database?**

Steps to create database:

- In solution explorer: Right click on your App\_Data folder.
- Click on add new item.
- Now dialog box will open from that select SQL Database.
- You can change the name of the database then click on the add button.
- There will be two files will be created that is mdf and ldf. the tables, stored procedures, views, etc. will be stored in the mdf file. While in ldf file sql stores the log information automatically. The log file, we cannot read it or modify it.

- **How to create Table in the database?**

Steps to create table:

- In server explorer: you can see the database file with the extension mdf.
- Right click on the table and click on add new table.
- The table will be created.

- o Click on the save button, at that time it will ask the name of the table.  
Set proper name and click on the save button.

#### 4. Binding Data:

- o Binding data is the method of all dynamic control.
- o This method attaches the data to the control and displays the data in the browser.
- o Below controls provide the functionality of Binding data.
  1. GridView
  2. DropDownList
  3. ListBox
  4. CheckBoxList
  5. RadioButtonList
  6. DataList
  7. DetailView
  8. FormView

#### **5. Database Programming using Code**

Generally we require performing following operations on database:

1. Select
2. Insert
3. Update
4. Delete

We require including system.data.sqlclient namespace to use SQL data provider.

**Code to insert, update and delete record from the database:**

Imports system.Data.SqlClient

- o Insert/Update/Delete Data Code:

```
Public Sub exeNonQuery(ByVal s As String)
    Dim cn as new sqlconnection
        cn = New SqlConnection (ConfigurationManager.
        ConnectionStrings("cn").ConnectionString)

    Try
        cmd = New SqlCommand(s, cn) //Here s means query
        cn.Open()
        cmd.ExecuteNonQuery()
    Catch ex As Exception
    End Try

End Sub
```

- o Select Data Code:

```
Public Function getDataset(ByVal s As String) As DataSet
    ds = New DataSet
    cmd = New SqlCommand(s, cn)// Here s means query
    Try
        cn.Open()
        da = New SqlDataAdapter(cmd)
        da.Fill(ds, "tbl")
        cn.Close()
    Catch ex As Exception
    End Try
    Return ds
End Function
```

## 6. Working with Data Controls:

### 1. Gridview:

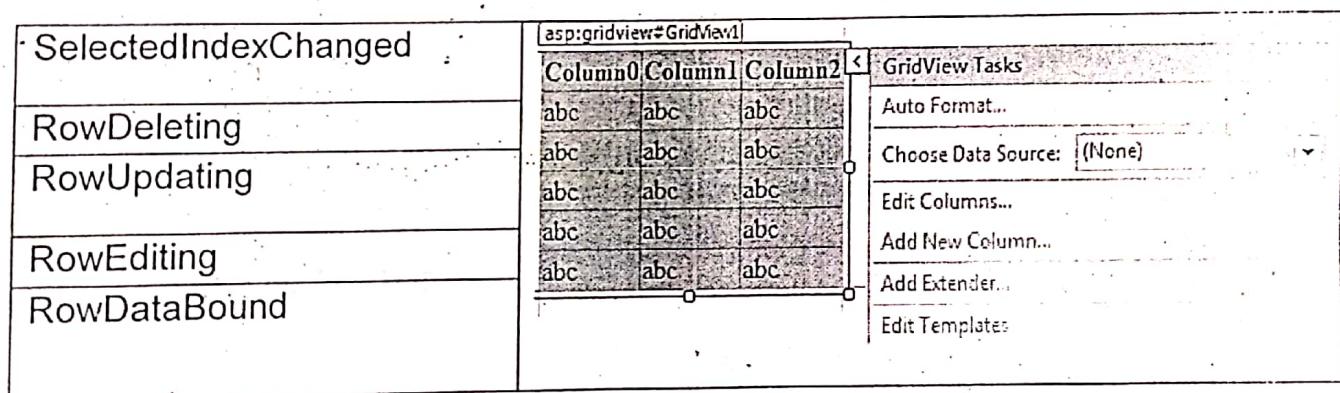
Gridview is a container control in which we can load n numbers of dynamic data. In this control we can place controls also. To display grid, we require at least one record to display. If there is no data then grid cannot display.

With this GridView control, you could display an entire collection of data, easily add sorting and paging, and perform inline editing. In addition to just displaying data, the GridView can be used to edit and delete the displayed data as well.

The GridView control provides many built-in capabilities that allow the user to sort, update, delete, select, and page through items in the control. To bind the data with the gridview control we require to set datasource property and databind method coding side.

We can format the gridview as per our requirement. Gridview is look like table control. We can set different formatting templates as well as we can set the different properties to create outstanding look. We can set the predefined columns to the gridview using edit columns properties of the GridView.

There are different events of gridview which are as below:



### 2. FormView:

Form view is a control in which we can load single record at a time. This is a dynamic control so we can load data from the database. We can insert the data, edit data and delete data using form view. Form view control provides different templates for different purposes which are listed below:

- a. Item template: this is a read only template and in which data is load from the data base for the viewing purpose.
- b. Edit item template: this is a template and in which data can be updated.
- c. Header template: this is a template in which we can set the common header for item template and edit item template.
- d. Footer template: this is a template in which we can set the common footer for item template and edit item template.

We have to design different templates as per our requirements. To load the data in the form view, we have to set the property datasource and in which we have to give dataset. We need to use data bind method to bind the data with form view and display in browser.

This is a control in which we can change the template run time. To change the template we have to use the property at coding side:

- a. Formview1.ChangeMode(FormViewMode.Edit),
- b. Formview1.ChangeMode(FormViewMode.Insert)
- c. Formview1.ChangeMode(FormViewMode.ReadOnly)

As per the template there are different events of the formview control which are given below:

- a. Mode Changing
- b. Mode Changed
- c. Item Updating
- d. Item Updated
- e. Page index changing

### 3. Details View:

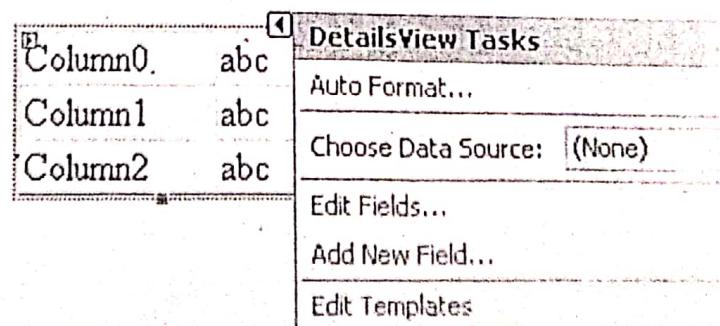
Details View is the control which is look like grid control and behaves like form view. We can load data dynamically in this control. We can load only one record at a time like FormView.

To load the data in this control we have to set DataSource property in which we have to give dataset and DataBind method which binds the data with control and displays data in the browser.

This control provides the facility to add, edit and delete the record. We can customize the look of this property. We can use the different formatting templates and properties of this control.

We can use the different events of this control which are listed below:

- a. Item inserting
- b. Item updating
- c. Item deleting
- d. Mode changing
- e. Page index changing



#### 4. DataList:

Data List is the control which behaves like gridview control and we can design as per our requirement like form view control. We can load n number of records in this control run time.

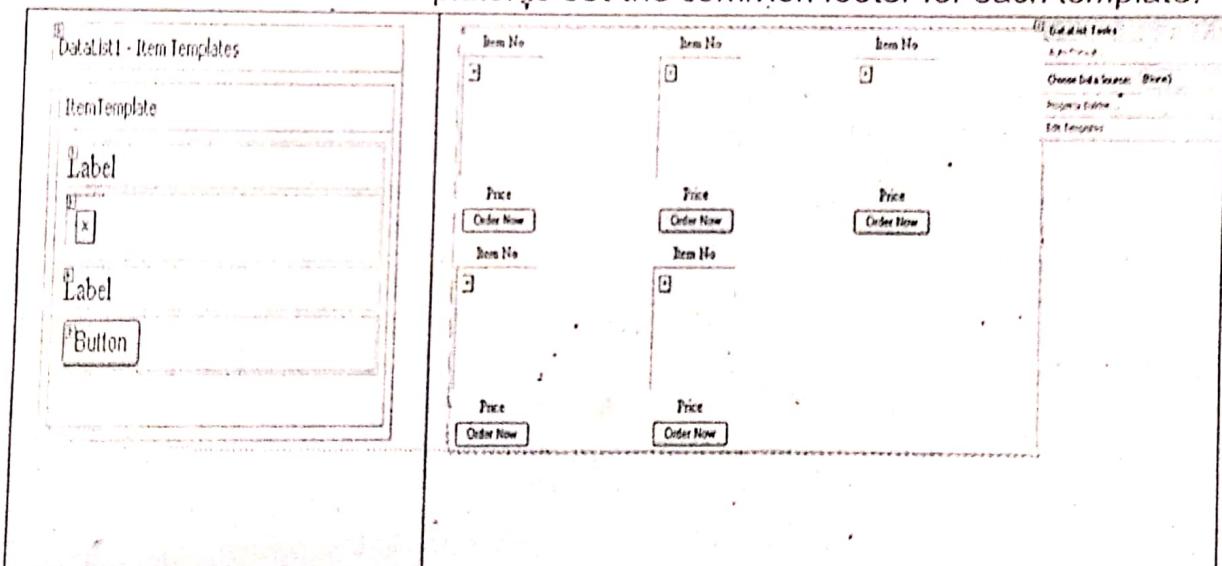
To use this control we have to design its template. This control provides the facility to display the records and to edit the records. We can also set the header and footer in this control.

You can design this control using auto format properties of this control. You can also design this control using property bar.

The important property of this control is **repeat columns** in which we can give order that data should be displayed in number of columns. the other property is **repeat direction** in which we can set that data should be displayed in vertical or horizontal direction.

Data list is providing number of templates which are:

- Item template: to display the records.
- Edit item template: to edit the records..
- Header template: to set the common header for each template.
- Footer template: to set the common footer for each template.

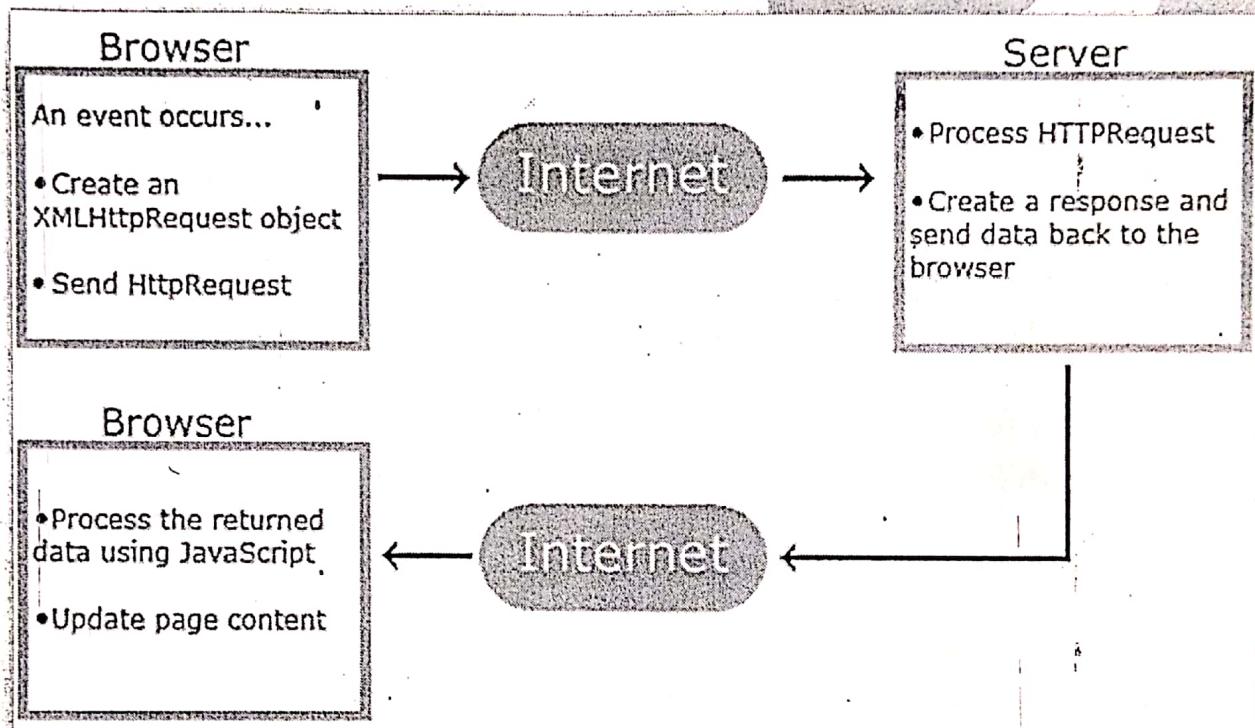


## \* What is Ajax

- AJAX = Asynchronous JavaScript and XML.
- AJAX is a technique for creating fast and dynamic web pages.
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

## \* How Ajax works

(Architecture)



When user first visits the page, the Ajax engine is initialized and loaded. From that point of time user interacts with Ajax engine to interact with the web server. The Ajax engine operates asynchronously while sending the request to the server and receiving the response from server. Ajax life cycle within the web browser can be divided into following stages:

- ✓ User Visit to the page: User visits the URL by typing URL in browser or clicking a link from some other page.
- ✓ Initialization of Ajax engine: When the page is initially loaded, the Ajax engine is also initialized. The Ajax engine can also be set to continuously refresh the page content without refreshing the whole page.
- ✓ Event Processing Loop:
  - \* Browser event may instruct the Ajax engine to send request to server and receive the response data

- \* Server response - Ajax engine receives the response from the server. Then it calls the JavaScript call back functions
- \* Browser (View) update - JavaScript request call back functions is used to update the browser. DHTML and css is used to update the browser display.

## ❖ Benefit of Ajax

- Ajax can be used for creating rich, web-based applications that look and works like a desktop application.
- Ajax is easy to learn. Ajax is based on JavaScript and existing technologies like XML, CSS, DHTML. etc. So, its very easy to learn Ajax.
- Ajax can be used to develop web applications that can update the page data continuously without refreshing the whole page.

## ❖ AJAX is Based on Internet Standards

AJAX is based on internet standards, and uses a combination of:

- ✓ XMLHttpRequest object (to exchange data asynchronously with a server)
- ✓ JavaScript/DOM (to display/interact with the information)
- ✓ CSS (to style the data)
- ✓ XML (often used as the format for transferring data)

❖ AJAX applications are browser- and platform-independent!

## ❖ Architecture of Asp.Net Ajax

Microsoft ASP.NET AJAX enables you to quickly create Web pages that include a rich user experience with responsive and familiar user interface (UI) elements. ASP.NET AJAX provides client-script libraries that incorporate cross-browser ECMAScript (JavaScript) and dynamic HTML (DHTML) technologies, and it integrates them with the ASP.NET 2.0 server-based development platform. By using ASP.NET AJAX, you can improve the user experience and the efficiency of your Web applications.

### ASP.NET AJAX Architecture

ASP.NET AJAX consists of client-script libraries and of server components that are integrated to provide a robust development framework. In addition to ASP.NET AJAX, you can use the ASP.NET AJAX Control Toolkit and the features in the ASP.NET AJAX Futures releases, which are both community supported.

## AJAX Client-Server Architecture

The ASP.NET AJAX server components basically consists of ASP.NET web server controls and components to manage UI and flow of an application, and to manage serialization, validation, control extensibility.

### ASP.NET Ajax Extension:

- 1) *ScriptManager* : Manages script resources for client components, partial-page rendering, localization, globalization, and custom user scripts.
- 2) *UpdatePanel* : Enables you to refresh selected parts of the page, instead of refreshing the whole page by using a synchronous postback.
- 3) *UpdateProgress* : Provides status information about partial-page updates in UpdatePanel controls.
- 4) *Timer* : Performs postbacks at defined intervals. You can use the Timer control to post the whole page, or use it together with the UpdatePanel control to perform partial-page updates at a defined interval.

## ASP.NET AJAX Client Architecture

The ASP.NET AJAX client-script libraries be made of JavaScript (.js) files that provide features for object-oriented development. The object-oriented features included in the ASP.NET AJAX client-script libraries enable a high level of consistency and modularity in client scripting. The following layers are included in the ASP.NET AJAX script libraries:

- 1) A browser compatibility layer. This provides compatibility across the most frequently used browsers (including Microsoft Internet Explorer, Mozilla Firefox, and Apple Safari) for your ASP.NET AJAX scripts.
- 2) ASP.NET AJAX core services, which include extensions to JavaScript, such as classes, namespaces, event handling, inheritance, data types, and object serialization.
- 3) An ASP.NET AJAX base class library, which includes components such as string builders and extended error handling.
- 4) A networking layer that handles communication with Web-based services and applications, and that manages asynchronous remote method calls.
- 5) Support for JavaScript libraries that are either embedded in an assembly or are provided as standalone JavaScript (.js) files. Embedding JavaScript libraries in an assembly can make it easier to deploy applications and can solve versioning issues.
- 6) Support for accessing server-based forms authentication and profile information in client script. This support is also available to Web applications that are not created by using ASP.NET, as long as the application has access to the Microsoft AJAX Library.

## ~~A~~ Working With AJAX Controls

### A) Accordion Control

- The Accordion is a web control that allows you to provide multiple panes and display them one at a time.
- It is like having several CollapsiblePanels where only one can be expanded at a time.
- The Accordion is implemented as a web control that contains AccordionPane web controls.
- Each AccordionPane control has a template for its Header and its Content. We keep track of the selected pane so it stays visible across postbacks.

### Important Properties

*SelectedIndex*

*AutoSize*

*FadeTransitions*

*TransitionDuration*

*FramesPerSecond*

### B) Auto Complete

- AutoComplete is an ASP.NET AJAX extender that can be attached to any TextBox control, and will associate that control with a popup panel to display words that begin with the prefix typed into the textbox.
- The dropdown with candidate words supplied by a web service is positioned on the bottom left of the text box

### Important Properties

*TargetControlID*

*ServiceMethod*

*ServicePath*

*MinimumPrefixLength*

### C) Calender

- Calendar is an ASP.NET AJAX extender that can be attached to any ASP.NET TextBox control.
- It provides client-side date-picking functionality with customizable date format and UI in a popup control. You can interact with the calendar by clicking on a day to set the date, or the "Today" link to set the current date.
- In addition, the left and right arrows can be used to move forward or back a month. By clicking on the title of the calendar you can change the view from Days in the current month, to Months in the current year. Another click will switch to Years in the current Decade. This action allows you to easily jump to dates in the past or the future from within the calendar control.

### Important Properties

*TargetControlID*

*Format*

*PopupButtonID*

*PopupPosition*

*SelectedDate*

*StartDate*

*EndDate*

### D) Filtered Textbox

- FilteredTextBox is an extender which prevents a user from entering invalid characters into a text box.
- Note that since this effect can be avoided by deactivating JavaScript, you should use this extender as a convenience for your users, but you must never expect that the data being sent to the server consists of "valid" chars only.

### Important Properties

*TargetControlID*

*FilterType* - Numbers, LowercaseLetters, UppercaseLetters, and Custom.

*FilterMode*

*ValidChars*

*InvalidChars*

*FilterInterval*

## \* Introduction to Web services

- Web Service is an application that is designed to interact directly with other applications over the internet.
- In simple sense, Web Services are means for interacting with objects over the Internet.
- The Web service consumers are able to invoke method calls on remote objects by using SOAP and HTTP over the Web.
- WebService is language independent and Web Services communicate by using standard web protocols and data formats, such as
  - HTTP
  - XML
  - SOAP

### ❖ Advantages of Web Service

- Web Service messages are formatted as XML, a standard way for communication between two incompatible system. And this message is sent via HTTP, so that they can reach to any machine on the internet without being blocked by firewall.

### ❖ Disadvantages of Web Service

- Availability.: Not available 100% at all time.
- Matching Requirement.
- Unchallengeable interface.
- Speed : slow down.
- Security and privacy issue.
- Authentication issue.
- Overheat : Conversion to and from in XML, Take extra time especially for binary image data.

### ❖ Example of Web Services

- Weather Reporting: You can use Weather Reporting web service to display weather information in your personal website.
- Stock Quote: You can display latest update of Share market with Stock Quote on your web site.
- News Headline: You can display latest news update by using News Headline Web Service in your website. And many more..

## ❖ What are Web Services?

- 1) Web services are application components
- 2) Web services communicate using open protocols
- 3) Web services are self-contained and self-describing.
- 4) Web services can be discovered using UDDI.
- 5) Web services can be used by other applications.
- 6) XML is the basis for Web services.

## ❖ How Does it Work?

- ✓ The basic Web services platform is XML + HTTP.
- ✓ XML provides a language which can be used between different platforms and programming languages and still express complex messages and functions.
- ✓ The HTTP protocol is the most used Internet protocol.
- ✓ *Web services platform elements:*
  - SOAP (Simple Object Access Protocol)
  - UDDI (Universal Description, Discovery and Integration)
  - WSDL (Web Services Description Language)

## What is SOAP?

SOAP is an XML-based protocol to let applications exchange information over HTTP.  
Or more simple: SOAP is a protocol for accessing a Web Service.

- SOAP stands for Simple Object Access Protocol
- SOAP is a communication protocol
- SOAP is a format for sending messages
- SOAP is designed to communicate via Internet
- SOAP is platform independent
- SOAP is language independent
- SOAP is based on XML
- SOAP is simple and extensible
- SOAP allows you to get around firewalls
- SOAP is a W3C standard

## What is WSDL?

WSDL is an XML-based language for locating and describing Web services.

Or

WSDL standard by which a web service can tell clients what messages it accepts and which results it will return.

- WSDL stands for Web Services Description Language
- WSDL is based on XML

- WSDL is used to describe Web services
- WSDL is used to locate Web services
- WSDL is a W3C standard

### What is UDDI?

UDDI is a directory service where companies can register and search for Web services.

- UDDI stands for Universal Description, Discovery and Integration
- UDDI is a directory for storing information about web services
- UDDI is a directory of web service interfaces described by WSDL
- UDDI communicates via SOAP
- UDDI is built into the Microsoft .NET platform.

### Why Web Services?

- 1) Interoperability has Highest Priority
- 2) Web Services take Web-applications to the Next Level
- 3) Web Services have Uses of - Connect Existing Software

### What is Discovery or .Disco Files?

→ Discovery files are used to group common services together on a web server. Discovery files .Disco and .VsDisco are XML based files that contains link in the form of URLs to resources that provides discovery information for a web service. Disco File contains URL for the WSDL, URL for the documentation and URL to which SOAP messages should be sent.

## \* CREATING AND USING WEB SERVICES

### STEP TO CREATING A WEB SERVICE.

1. Open Visual Web Developer.
2. On the File menu, click New Web Site.  
The New Web Site dialog box appears.
3. Under Visual Studio installed templates, click ASP.NET Web Service.
4. Type the name for webservices ( E.g. for TemperatureWebService.)
5. Write the web services programming under web method... here demonstrate to enter value convert for temperature.

Features

### WRITE A FOLLOWING CODE

<System.Web.Services.WebMethod()>

```
Public Function FahrenheitToCelsius(ByVal Fahrenheit As Double)
    As Double
    Return ((Fahrenheit - 32) * 5) / 9
End Function

<System.Web.Services.WebMethod()>
Public Function CelsiusToFahrenheit(ByVal Celsius As Double)
    As Double
    Return ((Celsius * 9) / 5) + 32
End Function
```

## TO TEST THE WEB SERVICES

1. In Solution Explorer, click Service.asmx, and then press **CTRL+F5**.
2. Copy the web url display in the browser.  
The Web service is invoked and a page appears in the browser that shows the methods that are exposed by the Web service.
3. Click CelsiusToFahrenheit, which invokes that method.  
A page appears that prompts you for parameter values for the CelsiusToFahrenheit method.
4. In the Celsius box, type 100, and then click **Invoke**.  
A new window appears that displays the XML that the Web service returns when the CelsiusToFahrenheit method is invoked. The value 212 appears in the XML.
5. Close the browser that contains the method results.
6. In the original browser, click Back to return to the list of methods.
7. Click FahrenheitToCelsius and test to make sure that the method is returning the results that you expect.  
If you type 212, the FahrenheitToCelsius method will return 100.
8. Close the browser.

You have finished creating the Web service; the next step is to use it.

## STEP TO USE WEB SERVICES in WEB SITE.

1. On the File menu, click New Web Site.
2. Under Visual Studio installed templates, click ASP.NET Web Site.
3. Type the name TemperatureWeb.

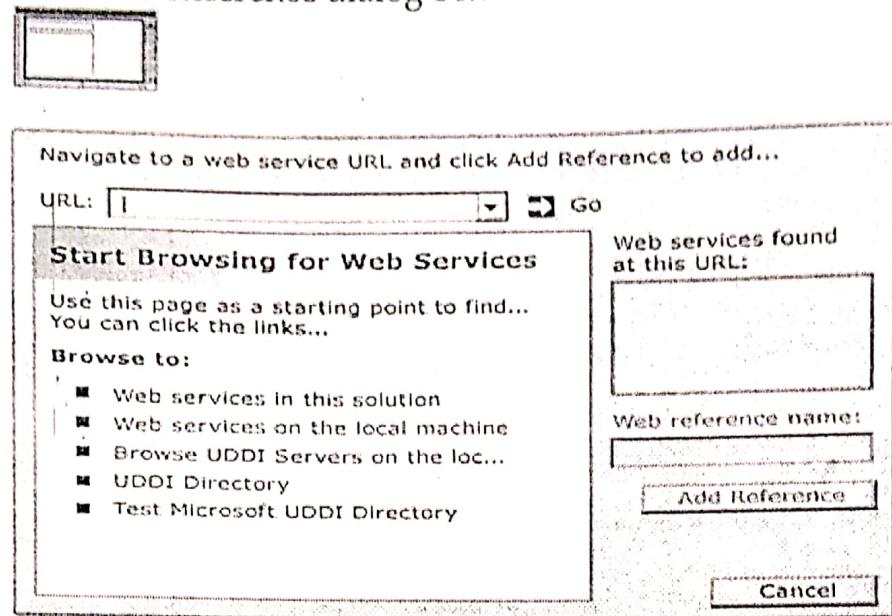
## ADDING THE WEB SERVICES AS A COMPONENT

*To create a reference to the Web service*

- 1) On the Web Site menu, click Add Web Reference.

The Add Web Reference dialog box appears, as shown in the following screenshot.

Add Web Reference dialog box

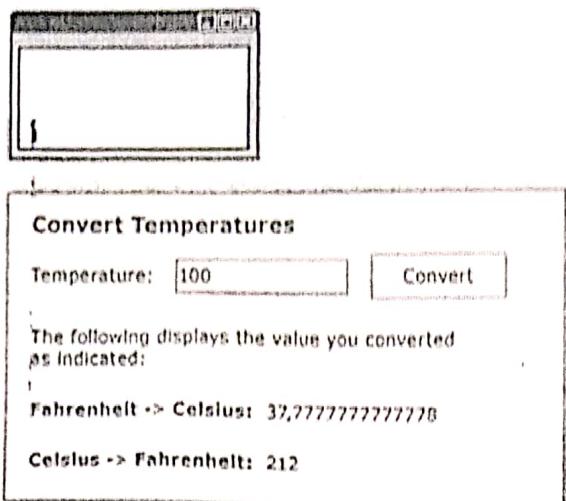


- 2) In the URL list, paste the copy url while test your web services e.g. for <http://localhost:<PortNumber>/TemperatureWebService/Convert.asmx> When Visual Web Developer finds the Web service, information about the Web service appears in the Add Web References dialog box.
- 3) Click Add Reference.

Visual Web Developer creates an App\_WebReferences folder and adds a folder to it for the new Web reference. By default, Web references are assigned a namespace corresponding to their server name (in this case, localhost). Make a note of the name for the Web reference namespace. In the folder, Visual Web Developer adds a .wsdl file that references the Web service. It also adds supporting files, such as discovery (.disco and .discomap) files, that include information about where the Web service is located.

You can now use the Web service. In this walkthrough, you will add controls to Default.aspx, and then program the controls to convert a specified temperature to both Fahrenheit and Celsius. When the page is running, it will look something like the following illustration.

Temperature conversion page



### To call the Web service methods

- 1) Open the Default.aspx page and switch to Design view.
- 2) From the Standard group in the Toolbox, drag the following controls onto the page and set their properties as indicated:

Control	Properties
Textbox	ID: TemperatureTextbox Text: (empty)
Button	ID: ConvertButton Text: (empty)
Label	ID: FahrenheitLabel Text: (empty)
Label	ID: CelsiusLabel Text: (empty)

- 3) Double-click ConvertButton to create an event handler for its Click event.
- 4) Make sure your event handler code matches the code in the following example.

```
Dim wsConvert As New localhost.Convert()  
Dim temperature As Double  
temperature = System.Convert.ToDouble(TemperatureTextbox.Text)  
FahrenheitLabel.Text = "Fahrenheit To Celsius = " & _  
    wsConvert.FahrenheitToCelsius(temperature).ToString()  
CelsiusLabel.Text = "Celsius To Fahrenheit = " & _  
    wsConvert.CelsiusToFahrenheit(temperature).ToString()
```

- 5) Press CTRL+F5 to run the page.
- 6) In the text box, type a value, such as 100, and then click Convert.

The page displays the result of converting the temperature value into both Fahrenheit and Celsius.