

Steps for opening process model

Control panel → Administrative tools → IIS → connection
→ Application Pool → Actions → Process Model

Features of Process Model

- ① Application Pool
- ② NUMA (Non Uniform Memory Access)
- ③ Ideal time limit
- ④ Response time of Ping
- ⑤ Startup time limit
- ⑥ Shutdown time limit (of IIS)

ASP : Active Server Pages

ASP.NET

- i ASP.NET was released in 2002 as a successor to classic ASP.
- ii ASP.NET Pages have the extension .aspx and are normally written in C# (C sharp)
- iii ASP.NET 4.8 is the latest official version of ASP.NET
- iv ASP.NET 5 was expected to be an important redesign of ASP.NET.
However, the development of ASP.NET 5 was stopped in favor of ASP.NET Core.

ASP includes 5 built-in objects

- i Request - to get information from the user that is passed along with an HTTP request
- ii Response - to output data to requesting client
- iii Server - To control Internet Information Server
- iv Session - to store variables associated with a given user session
- v Application - to store information that remain active for lifetime of an application such as page counter.

Advantages of ASP

- i Dynamic Web Page
- ii Browser independent
- iii Database Access
- iv Built-in objects

Difference

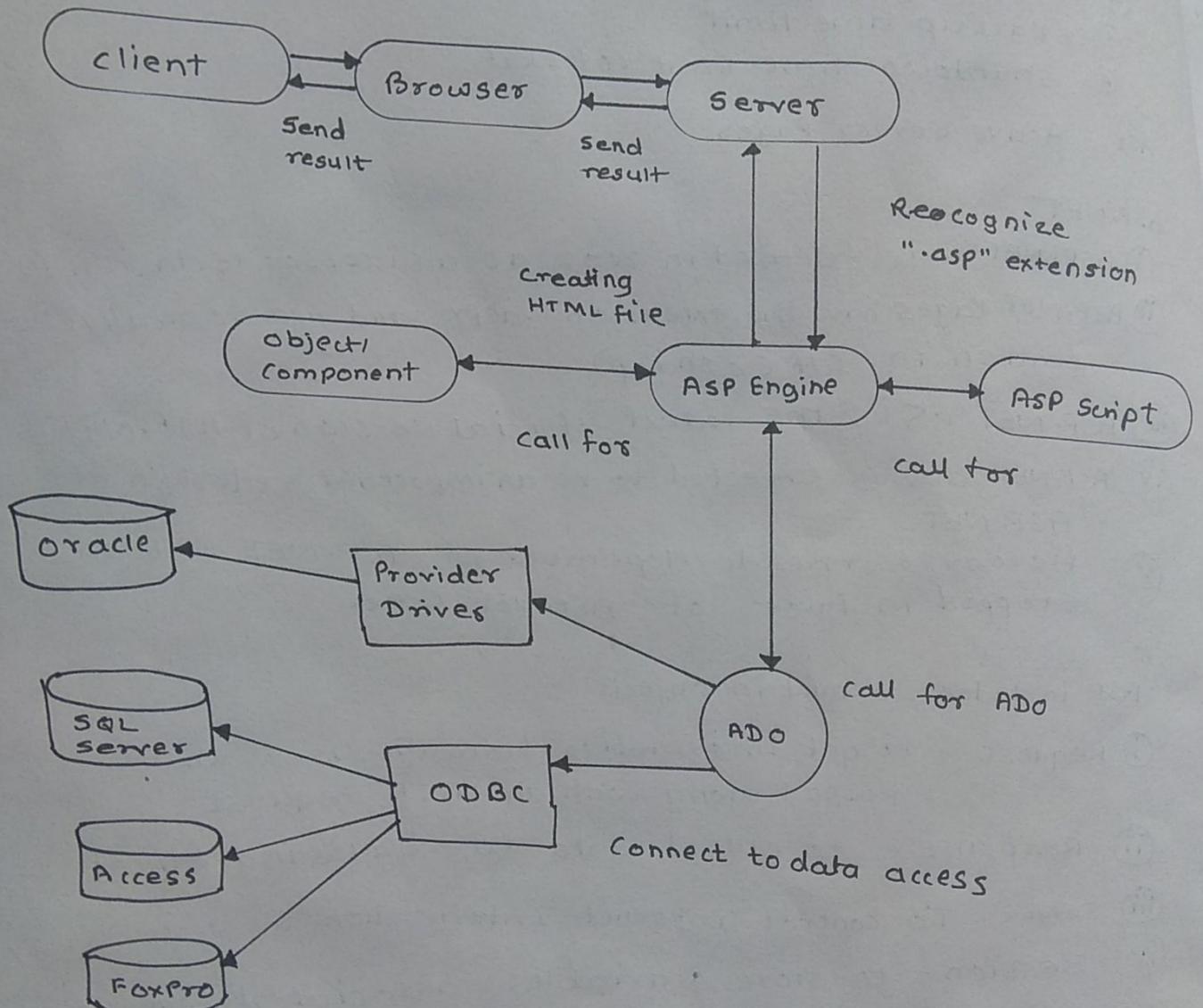


Fig. Process Model of Asp

i ASP i
ii ASP L
technol
with e
iii A
iv

Difference between ASP and ASP.NET

ASP

- i) ASP is the interpreted language.
- ii) ASP uses ADO (ActiveX Data Objects) technology to connect and work with database.
- iii) ASP is partially object oriented.
- iv) ASP Pages have the file extension .asp
- v) ASP doesn't have concept of inheritance.
- vi) ASP Pages uses scripting language
- vii) Error handling is very poor in ASP
- viii) No real state management is provided.
- ix) In ASP, files can be updated only when server is down (in off State)
- x) Mix layout of HTML and Scripting code.

ASP.NET

- i) ASP.NET is the compiled language.
- ii) ASP.NET uses ADO.NET to connect and work with database.
- iii) ASP.NET is fully object oriented
- iv) ASP.NET pages have the file ~~extension~~ extension .aspx
- v) ASP.NET inherit the class written in code behind.
- vi) ASP.NET uses full-fledged Programming language.
- vii) Error handling is very good in ASP.NET
- viii) Graphical development environment provided with state management
- ix) Files can be updated while server is in running mode.
- x) Separation of code from HTML

Features of Common Language Runtime (CLR)

- i) The common language runtime manages memory, thread execution, code execution, code safety verification, compilation and other system services
- ii) The runtime also enforces code robustness by implementing a strict type and code-verification infrastructure called common Type System (CTS)
- iii) The automatic memory management resolves the two most common app errors, memory leaks and invalid memory references.
- iv) Memory manager removes possibilities of fragmented memory and increases memory locality of reference to further increase performance.
- v) A feature called Just-in-time (JIT) compiling enables all managed code to run in native machine language of system on which its executing
- NET Framework class library
 - The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code derives functionality.

Difference between global.asax and web.config file

Global.asax

web.config

- ① It is a class file
- ② There can be only one global.asax file for an application
- ③ It can have application and session events
- ④ Need to be recompiled when changes are made
- ⑤ It's the optional file containing declaration of variables, objects and methods
- ⑥ JavaScript, VBScript, Perl will be used in Global.asax file
- ⑦ It contains code to respond to application-level events raised by ASP.NET
- ⑧ It is never called directly by the user, rather they are called automatically in response to application events
- ⑨ It is also called ASP.NET Application file

- ① It is an XML file
- ② There can be many web.config files under different subfolders
- ③ It cannot have Application and Session Events.
- ④ No need to compile when changes are made
- ⑤ It deals with single appn, manages various settings that defines your website accessibility.
- ⑥ Settings will be done in XML
- ⑦ It contains the settings and configuration file for an ASP.NET appn.
- ⑧ It gets automatically created when we create any project/website and it due to some framework versions it is not created it can be created manually
- ⑨ It is also called configuration file.

Difference between IIS and PWS

IIS

- ① IIS stands for Internet Information Services
- ② It is the most popular web server for windows 2000
- ③ It has more features
- ④ It is intended for latest versions of windows
- ⑤ It is Microsoft's full blown and robust version

PWS

- ① PWS stands for Personal web Server
- ② It is a basic web server for publishing personal web pages.
- ③ It has less features than IIS
- ④ It is supported by windows 95 or 98 or NT
- ⑤ It is scaled down version of IIS

Namespaces of ASP.NET

System: Contains classes for implementing base datatypes.

In addition, It contains classes for working with date and time.

System.Web: Contains classes and interfaces that enable browser/server communication

System.Web.UI: Contains the base classes used in building the user interface for ASP.NET Page

System.Web.UI.WebControls: Contains classes for providing web controls.

System.Drawing: Contains classes for providing access to basic graphics functions.

System.Web.Services: Contains web services classes which allows us to create web services

The Page Life Cycle phases

- 1) Initialization
- 2) Instantiation of controls on the page
- 3) Restoration and maintenance of the state
- 4) Execution of the event handler codes
- 5) Page Rendering.

Types of controls used in ASP.NET

- 1) Web Forms Standard controls
- 2) Navigation Controls
- 3) Validation Controls
- 4) Web parts Controls
- 5) HTML controls

requests that involve managed resources

Integrated mode or classic mode

HTTP Request Processing in IIS

- ① When a client browser initiates an HTTP request for a resource on the web server, HTTP.SYS intercepts the request.
- ② HTTP.SYS contacts WAS to obtain information from the configuration store.
- ③ WAS requests configuration information from configuration store, applicationHost.config
- ④ The WWW Service receives configuration information, such as application pool and site configuration.
- ⑤ The www service uses the configuration information to configure HTTP.SYS
- ⑥ WAS starts a worker process for the application pool to which the request was made
- ⑦ The worker process processes the request and returns a response to HTTP.SYS
- ⑧ The client receives a response

ASP.NET IDE

- ① Toolbox
- ② Properties window
- ③ Solution Explorer
- ④ Design Page
- ⑤ Code Behind Page
- ⑥ Server Explorer
- ⑦ The Output window

Page directives are instructions, inserted at the top of ASP.NET page to control behaviour of ASP.NET Pages

Object in ASP.NET but Response.Redirect can't use with Context object in ASP.NET

The syntax of Context object in ASP.NET

```
context.Items["Id"] = value;
```

Retrieve the context value on Next Page in ASP.NET

```
String myvalue = context.Items["Id"].ToString()
```

Process Model Settings for an Application Pool

By using <processModel> element, you can configure many of the security, performance, health and reliability features of application pools on IIS 7 and later. These include following features

- 1) Application pool identity
- 2) Web gardening and use of Non-Uniform Memory Access
- 3) maxProcesses
- 4) idleTimeout
- 5) PingEnabled, PingInterval and pingResponseTime
- 6) shutdownTimeLimit
- 7) startUptimeLimit

C# String Functions

- i) clone()
- ii) CompareTo()
- iii) Contains()
- iv) EndsWith()
- v) StartsWith()
- vi) Equals()
- vii) ToLower()
- viii) ToUpper()
- ix) Length()
- x) Remove()
- x-1) Replace()
- x-2) Trim()

Server Object

The ASP Server Object is used to access properties and methods on the server associated with IIS or PWS.

- Properties
 - 1) ScriptTimeout
- Methods
 - 1) CreateObject
 - 2) Execute
 - 3) GetLastError
 - 4) HTML Encode
 - 5) MapPath
 - 6) Transfer
 - 7) URLEncode

Application Object

The ASP Application object holds information that will be used by many pages in the application (like database connection information). The information can be accessed from any page. The information can also be changed in one place and the changes will automatically reflect on all pages.

- Collections
 - 1) Contents
 - 2) StaticObjects
- Methods
 - 1) Contents.Remove
 - 2) Contents.RemoveAll
 - 3) Lock
 - 4) Unlock
- Events
 - 1) Application_OnEnd
 - 2) Application_OnStart

Context Object

The Context Object is used to store the value and send it to other page in ASP.NET. The main difference between Context and Session is Context Object will be null when we send page to server. That means when we use Context Object we must use Server.Transfer method to redirect user to other page. If we use Response.Redirect the Context value will be null on other page. We can use Response.Redirect and Server Transfer with session.

Transactions

A transaction symbolizes code or a set of components or procedures which must be executed as a unit. All the methods must execute successfully or the complete unit fails. A transaction can be described to cover the ACID properties for mission critical applications

1. Atomicity 2. Consistency 3. Isolation 4. Durability

```

##config
    ERMSG      <!-- ##config ERMSG = "An error occurred" -->
    TIMEFMT
    SIZEFMT

##echo
    <!-- ##echo var=VarName -->

##exec
    <!-- exec ##exec CommandType = CommandDescription -->
    Command type: cgi and cmd

##flastmod
    <!-- ##flastmod PathType = FileName -->
    PathType
        file : relative
        virtual : absolute

##fsize
    <!-- #fsize PathType = FileName -->
##include
    <!-- #include PathType = FileName -->

```

ADO.NET Framework providers

① OLEDB: The OLE DB provider, expressed through the System.Data.OleDb namespace. You can use this provider to access SQL Server 6.5 and earlier, Sybase DB2/400 and Microsoft Access.

② ODBC: The ODBC provider, expressed through the System.Data.Odbc namespace. This provider is typically used when no newer provider is available.

③ SQL Server: The Microsoft SQL Server provider, expressed through the System.Data.SqlClient namespace. It contains classes that provide functionality similar to the generic OleDb provider. The difference is that these classes are tuned for SQL Server 7 and later data access.

Core objects that make up .NET framework data providers

- Connection: Establishes a connection to a specific data source. The base class for all connection objects is the DbConnection class.
- Command: Executes a command against a data source. Exposes Parameters and can execute in the scope of Transaction from a Connection. The base class for all command objects is the DbCommand class.
- DataReader: Reads a forward-only, read-only stream of data from a data source. The base class for all DataReader objects is the DbDataReader class.
- DataAdapter: Populates a Dataset and resolves updates with the data source. The base class for all DataAdapter objects is the DbDataAdapter class.

Sample code for db interaction

```
{  
    String conString = null;  
    SqlConnection cnn;  
    SqlCommand cmd;  
    String sql = null;  
  
    conString = "DataSource=ServerName; InitialCatalog=DatabaseName;  
    UserID=UserName; Password=Password";  
    sql = "Your SQL Statement";  
    cnn = new SqlConnection(conString);  
    try  
    {  
        cnn.Open();  
        cmd = new SqlCommand(sql, cnn);  
        cmd.ExecuteNonQuery();  
        cmd.Dispose();  
        cmd.Close();  
        cnn.Close();  
        MessageBox.Show("Command executed");  
    }  
    catch (Exception e)  
    {  
        Response.WriteLine("An error occurred");  
    }  
}
```

Page Counter Component

The Page Counter component creates a PageCounter object that counts and displays the no. of times a web page has been opened.

Transactions Properties

Atomicity: An atomic transaction defines the set of operations which is completed successfully.

Consistency: If any type of transaction exists at the user end, that must be conceived into valid and legal state in database.

Isolation: When multiple transactions execute at a single unit of time, noteworthy then the state of database must be fair all over the transaction period.

Durability: Finally some common external factors like system crash, power failure but failure should not affect the data once the transaction is completed on the server.

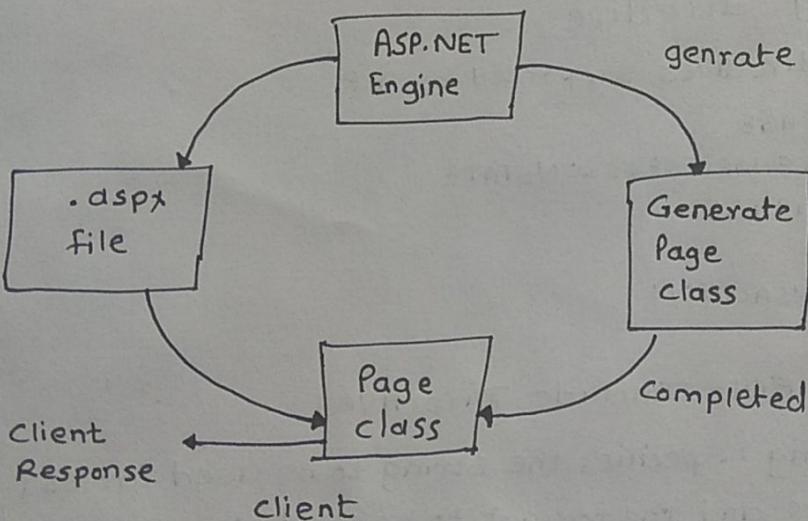


Fig. Execution model ASP.NET web page

Global.asa file

The Global.asa file is an optional file that can contain declarations of objects, variables and methods that can be accessed by every page in ASP application.

The global.asa file can contain only following:

- i) Application events
- ii) Session events
- iii) <object> declarations
- iv) TypeLibrary declarations
- v) The #include directive

NOTE: The Global.asa file must be stored in the root directory of ASP application and each application can only have one Global.asa file.

ASP directives

- ① Output directives
- ② Processing directive

Processing directives supported by ASP.

- @CODEPAGE
- @~~EN~~ABLESESSIONSTATE
- @LANGUAGE
- @LCID
- @TRANSACTION

Server Side Include Directives

- #config : Specifies the string to be used for SSI error messages and the format to be used for dates and size file sizes returned by the #lastmod and #fsize directives
- #echo : Inserts the value of a server variable into an HTML page
- #exec : Runs an application or a shell command and inserts the output into an HTML page
- #lastmod : Inserts the modification date of a file into web page, formatted by the #config directive
- #fsize : Inserts the size of a file into a Web page formatted by #config directive
- #include : Includes a file in web page. This is the only directive that can be used in ASP Pages and SSI pages.

IIS (Internet Information Server)

IIS is one of the most powerful web servers from Microsoft that is used to host your web application. IIS has its own Process Engine to handle the request. So when a request comes from client to Server, IIS takes the request and process it and send response back to the client.

Functions in ASP

- 1. Loop Functions [Do until loop, Do while loop]
- 2. Response functions [Response.Redirect, Response.write]
- 3. Date and time functions
- 4. String functions

ASP Objects

Request Object

When a browser asks for a page from server it is called a request. The request object is used to get information from a visitor. Request object uses to access any information that is passed with HTTP request. The Request object also gives access to binary data sent to the server such as file uploads.

- Collections
 - 1) Client Certificate
 - 2) Cookies
 - 3) Form
 - 4) QueryString
 - 5) Server Variables
- Properties
 - 1) TotalBytes
- Methods
 - 1) BinaryRead

Sample -

Response Object

The ASP Response object is used to send output to the user from the server. Its collections are

- collections
 - 1) Cookies
 - Properties
 - 1) Buffer
 - 2) CacheControl
 - 3) Charset
 - 4) ContentType
 - 5) Expires
 - 6) ExpiresAbsolute
 - 7) IsClientConnected
 - 8) PICS
 - 9) Status
 - methods
 - 1) AddHeader
 - 2) AppendToLog
 - 3) BinaryWrite
 - 4) Clear
 - 5) End
 - 6) Flush
 - 7) Redirect
 - 8) Write

Session Object

The Session object stores information about a user session. Variables stored in session object hold information about one single user and are available to all pages in one application. Common information stored in session variable are name, id and preferences. It allows the management of an user connection with the server.

- collections
 - 1) Contents
 - 2) StaticObjects
- Properties
 - 1) CodePage
 - 2) LCID
 - 3) SessionID
 - 4) TimeOut

Methods

- 1) Abandon ~~(Session.Abandon())~~
- 2) ContentRemoved ~~(Session.ContentRemoved())~~
- 3) ~~ContentReSession~~ ~~(Session.ContentReSession())~~

Events

- 1) Session_OnEnd
- 2) Session_OnStart

Import
① Conn
② Co
③ C

Important classes in ADO.NET required for database connectivity

- ① Connection
- ② Command
- ③ DataReader
- ④ DataAdapter
- ⑤ DataSet

DataAdapter

It fetches datasets and resolves the update with the dataSource
The base class for all DataAdapter objects is the ~~Db~~DataAdapter class

Two types of connection architecture

- 1) Connection oriented
- 2) Disconnected Architecture

ADO is the acronym for ActiveX Data Objects

Steps

- 1) First you have to create a Connection object and open a connection to the Data Source specified in the connection string
- 2) Assign the open connection to the Connection property of the Command object. Then the Command object can execute the SQL statements.
- 3) After the execution of the SQL statement, the Command object will return a result set
- 4) We can retrieve the result set using DataReader

Connection class

In ADO.NET, we use these Connection classes to connect to the database. These Connection classes also manage transaction and connection pooling

Command class

The Command class provides methods for storing and executing SQL statements and stored procedure. Following are commands Executed by Command class

- ① ExecuteReader: Returns data to the client as rows. This would typically be an SQL select or stored procedure that contains one or more select statements. This method returns a DataReader object that can be used to fill a DataTable object or used directly for printing reports and so forth.

- ② ExecuteNonQuery: Executes a command that changes the data in the database - such as an update, delete or insert statement, or a stored procedure that contains one or more of these statements. This method returns an integer, that is the no. of rows affected by the query.
- ③ ExecuteScalar: This method only returns a single value. This kind of query returns a count of rows or a calculated value.
- ④ ExecuteXMLReader: (SqlClient classes only) obtains data from an SQL Server 2000 database using XMLStream - Returns an XML Reader object.

DataReader class

The DataReader is used to retrieve data. It is used in conjunction with the Command object class to execute SQL Select statement and then access the returned rows.

DataAdapter class

The DataAdapter is used to connect datasets to databases. The DataAdapter is most useful when using data-bound controls in windowsforms, but it can be also be used to provide an easy way to manage the connection between your application and the underlying database tables, views and stored procedures.

Dataset class

The Dataset is the heart of ADO.NET. The dataset is essentially a collection of DataTable objects. In turn each object contains a collection of DataColumn and DataRow objects. The Dataset also contains a Relations collection that can be used to define relations among DataTable objects.

Common Language Runtime (CLR)

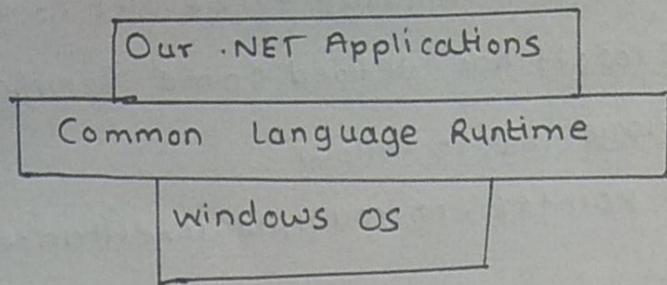


Fig. Common Language Runtime

The heart of the .Net framework. It is also called the .Net runtime. It resides above operating system and handles all .Net applications. It handles garbage collection, Code Access Security (CAS) etc.

Microsoft Intermediate Language (MSIL) code

When we compile our .Net code then it is not directly converted to native/binary code; It is first converted into intermediate code known as MSIL code which is then interpreted by CLR.

MSIL is independent of hardware and operating system. Cross language relationships are possible since MSIL is same for all .Net languages. MSIL is further converted into native code.

Just in Time Compilers

It compiles IL code into native executable code (exe or dlls). Once code is converted to IL it can be called again by JIT instead of ~~recompiling~~ recompiling that code

Framework class library

The .Net Framework provides a huge class library called FCL for common tasks. It contains thousands of classes to access Windows API and common functions like string manipulations, Data Structures, Stream, IO, thread, Security etc.

Common Language Specification (CLS)

What makes a language to be .NET compliant? Answer is CLS. Microsoft has defined some specifications that each .NET language has to follow for e.g. no pointer, no multiple inheritance etc.

Common Type System (CTS)

CTS defines some basic data types that IL can understand. Each .NET compliant language should map its data types to these standard data types. This makes it possible for two .NET compliant languages to communicate by sending/receiving parameters to and from each other. For e.g. CTS defines Int32 for C# int and VB integer data types.

The .NET Framework

It is a combination of CLR, FCL, ADO.NET and XML classes, Web/Window applications and web services.

IIS Web Server

Internet Information Server (IIS) is one of the most popular web servers from Microsoft that is used to host and provide Internet-based services to ASP.NET and ASP web applications. When a request comes from client to server IIS takes that request from user and process it and send response back to users.

Application Pools in IIS

Application pools separate applications by process boundaries to prevent an application from affecting another application on the server. In IIS 7 and later, application pools continue to use IIS 6.0 worker process isolation mode. In addition,

You can now specify a setting that determines how to process isolation mode. In addition, you can now specify a setting that determines how to process

DataAdapter

DataAdapter will act as a bridge between Dataset and DataBase. This DataAdapter object is used to read the data from database and bind that data to dataset. DataAdapter is disconnected oriented architecture.

- Let you close the connection as soon its done loading data and may even
- All the results are available in memory
- You can iterate over it as many times you need, or even look up a specific record by index.
- Has some built-in facilities for updating back to the database.

DataTable

DataTable represents a single table in the database. It has rows and columns. There is no much difference between dataset and datatable. dataset is simply collection of datatables.

A DataRow represent a row of data in datatable. You add data to the datatable using DataRow object. A DataRowCollection object represent a collection of data rows of a datatable. You use DataTable's NewRow method to return a DataRow object of datatable, add values to the data row and add a row to dataTable again by using DataRowCollection's Add method.

```
DataRow row1 = custTable.NewRow();
row1["id"] = 1001;
row1["Address"] = "Nashik";
row1["Name"] = "George Bishop";
```

web.config

- debug
- targetFramework
- customErrors
- defaultRedirect
- authenticationMode
- connectionStrings

The basic flow to accessing a data object

- i Create a connection to data object
- ii Create a command for data object to act upon
- iii Read data returned by data object.

Common methods of SqlCommand class

ExecuteReader: when the T-SQL statement gives more than a single value in return then you can use ExecuteReader command. e.g. if the query returns rows of data [SELECT statement]

ExecuteNonQuery: When you need to perform an Insert, Update or Delete function then you can use ExecuteNonQuery command.

ExecuteScalar: When the query returns a scalar value then you can use this function e.g. queries that returns total no. of rows in table.

Stack

Program:

```

using System.Collections.Generic;
protected void Button4_Click(object sender, EventArgs e)
{
    Stack<string> stk = new Stack<string>();
    stk.push("cs.net");
    stk.push("vb.net");
    stk.push("asp.net");
    stk.push("sqlserver");
    foreach (string s in stk)
    {
        Response.Write(s + "<br>");
    }
}

```

Queue

Program:

```

using System.Collections.Generic;
protected void Button5_Click(object sender, EventArgs e)
{
    Queue<string> q = new Queue<string>();
    q.Enqueue("cs.net");
    q.Enqueue("vb.net");
    q.Enqueue("asp.net");
    q.Enqueue("sqlserver");
    foreach (string s in q)
    {
        Response.Write(s + "<br>");
    }
}

```

Dictionary

Program:

```
using System.Collections.Generic;
protected void Button2_Click (object sender, EventArgs e)
{
    Dictionary<int, string> dict = new Dictionary<int, string>();
    dict.Add (1, "cs.net");
    dict.Add (2, "vb.net");
    dict.Add (3, "vb.net");
    dict.Add (4, "vb.net");

    foreach (KeyValuePair<int, string> kvp in dict)
    {
        Response.Write (kvp.Key + " " + kvp.Value);
        Response.Write ("  
");
    }
}
```

SortedList

Program:

```
using System.Collections.Generic;
protected void Button3_Click (object sender,
                           EventArgs e)
{
    SortedList<string, string> sl = new SortedList<string,
                                             string>();
    sl.Add ("ora", "Oracle");
    sl.Add ("vb", "vb.net");
    sl.Add ("cs", "cs.net");
    sl.Add ("asp", "asp.net");

    foreach (KeyValuePair<string, string> kvp in sl)
    {
        Response.Write (kvp.Key + " " + kvp.Value);
        Response.Write ("  
"); Response.Write ("  
");
    }
}
```

Output [queue]

CS.NET
VB.NET
ASP.NET
SQL Server

Generic Collections

XOR

Generic collections work on the specific type that is specified in the program whereas non generic work on the object type.

- a] Specific Type
- b) Array size is not fixed
- c] Elements can be added/removed at runtime

List

Program:

```
using System.Collections.Generic;
protected void Button1_Click(object sender, EventArgs e)
{
    List<int> lst = new List<int>();
    lst.Add(100);
    lst.Add(200);
    lst.Add(300);
    lst.Add(400);
    foreach (int i in lst)
    {
        Response.Write(i + "<br>");
    }
}
```

Stack

Program

Protected void Button4_Click(object sender, EventArgs e)

```
{  
    Stack Stk = new Stack();  
    Stk.push("cs.net");  
    Stk.push("vb.net");  
    Stk.push("asp.net");  
    Stk.push("sqlserver");  
    foreach (object o in Stk)  
    {  
        Response.Write(o + "<br>");  
    }  
}
```

Output

sqlserver
asp.net
vb.net
cs.net

Queue

Program

XOX

using System.Collections;

```
protected void Buttons_Click (object o, EventArgs e)  
{  
    Queue q = new Queue();  
    q.Enqueue ("cs.net");  
    q.Enqueue ("vb.net");  
    q.Enqueue ("asp.net");  
    q.Enqueue ("sqlserver");  
    foreach (object o in q)  
    {  
        Response.Write(o + "<br>");  
    }  
}
```

3

Sortedlist

1. Is a class that has the combination of arraylist and hashtable
2. Represents data as a key and value pair
3. Arranges all the items in sorted order

Program:

```
using System.Collections;
```

```
Protected void Button3_Click(object sender, EventArgs e)
```

```
{  
    SortedList sl = new SortedList();  
    sl.Add("ora", "oracle");  
    sl.Add("vb", "vb.net");  
    sl.Add("cs", "cs.net");  
    sl.Add("asp", "asp.net");
```

```
foreach (DictionaryEntry d in sl)
```

```
{  
    Response.WriteLine(d.Key + " " + d.Value);  
    Response.Write("<br>");  
}
```

```
}
```

Output:

asp	asp.net
cs	cs.net
ora	oracle
vb	vb.net

Hash table

Hash table is similar to arraylist but represents the items as a combination of key and value.

Program:

```
using System.Collections;
Protected void Button1_Click(object sender, EventArgs e)
{
    Hashtable ht = new Hashtable();
    ht.Add("ora", "oracle");
    ht.Add("vb", "vb.net");
    ht.Add("cs", "cs.net");
    ht.Add("asp", "asp.net");
    foreach (DictionaryEntry d in ht)
    {
        Response.Write(d.Key + " " + d.Value);
        Response.Write("<br>");
    }
}
```

DictionaryEntry : is a class whose object represents data in a combination of key and value pairs.

Non-generic Collections

- a] Each element can represent a value of different type.
- b] Array size is not fixed.
- c] Elements can be added / removed at runtime.

— XOX —

ArrayList

1. ArrayList is a class that is similar to array but it can be used to store values of various types
2. An ArrayList doesn't have a specific size
3. Any number of elements can be stored

Program

Using System.Collections;

```
protected void Button1_Click (Object sender, EventArgs e)
{
    ArrayList al = new ArrayList();
    String str = "Hello world";
    int x = 7;
    DateTime d = DateTime.Parse("8-oct-1985");
    al.Add(str);
    al.Add(x);
    al.Add(d);

    foreach (Object o in al)
    {
        Response.Write(o);
        Response.Write("<br>");
    }
}
```

NOTE: ArrayList allocates memory for 4 items, whenever an object is created, when a fifth is added, memory for another 4 items are added. It reduces the memory allocated for the object.

Capacity: is a property that returns the no. of items for which memory is allocated.

<%@ Page EnableSessionState="False" %>

Machine

To Create folder: `CreateDirectory (DirectoryPath);`

To Delete folder: `Directory.Delete(directoryPath);`

• Collections

Non-generic

ArrayList

HashTable

SortedList

Stack

Queue

Generic

List

Dictionary

SortedList

Stack

Queue

collections are similar to Arrays, it provides a more flexible way of working with a group of objects.
In a collection, you don't need to define the size of the collection beforehand. You can add elements or even remove elements from the collection at any point of time.

ArrayList: The ArrayList collection is similar to Arrays datatype in C#. The biggest difference is the dynamic nature of array list collection.

Stack: The Stack is a special case collection which represents a last in First out (LIFO) concept.

Queue: The Queue is a special case collection which represents the first in first out concept.

HashTable: A ~~hashtable~~ hashtable is a special collection that is used to store ~~key~~ key-value items.

SortedList: The SortedList is a collection which stores key-value pairs in the ascending order of key by default.

BitArray: A bit array is an array of data structure which stores bit.

Machine.config vs Web.config

Machine.config

- i) The machine.config file is the master configuration file on your system with a lot of default settings.
- ii) The settings of Machine.config file are applied to the whole asp.net applications on your server.
- iii) Each .NET framework version has only one machine.config file.
- iv) The machine.config file is at the highest level in the configuration hierarchy.
- v) The machine.config would be to share values between many apps on the server such as SMTP Server settings.
- vi) If you make any changes in machine.config you will have to restart the application.
- vii) The machine.config file will automatically installed when you install Visual Studio .Net
- viii) Machine.config is configuration file for all the application in the IIS

Web.config

- i) web.config is the file for the local settings to be applied for a website which store configuration data in XML format.
- ii) Settings made in web.config file are applied to the particular web application only.
- iii) each web app has its own web.config file. Directories inside a web app can also have web.config files too.
- iv) web.config file override the settings from the machine.config file.
- v) web.config file contains application specific items such as database connection strings.
- vi) If you make any changes to web.config web application will immediately load the changes.
- vii) web.config will automatically created when you create Asp.net web application project
- viii) web.config is a configuration file for a particular app or folder

Browse
pre
or

Sending mail

```
Protected void Page_Load ( object sender , EventArgs e )  
{  
    try  
    {  
        MailMessage mailMessage = new MailMessage ();  
        mailMessage . To . Add (" your . own @gmail . com ");  
        mailMessage . From = new MailAddress (" another@ gmail . com ");  
        mailMessage . Subject = " ASP . NET email Test ";  
        mailMessage . Body = " HelloWorld ";  
        SmtpClient smtpClient = new SmtpClient (" smtp .  
        gmail . com ");  
        smtpClient . Send ( mailMessage );  
        Response . Write (" Email - sent ");  
    }  
    catch ( Exception e )  
    {  
        Response . Write (" Could not send e-mail " + e );  
    }  
}
```

Browser Capabilities

```
protected void Page_Load(object sender, EventArgs e)
```

{

```
    System.Web.HttpBrowserCapabilities browser =
```

```
        Request.Browser;
```

```
    string s = "Browser Capabilities\n"
```

```
+ "Type = " + browser.Type + "\n"
```

```
+ "Name = " + browser.Browser + "\n"
```

```
+ "Version = " + browser.Version + "\n"
```

```
+ "Major Version = " + browser.MajorVersion + "\n"
```

```
+ "Minor Version = " + browser.MinorVersion + "\n"
```

```
+ "Platform = " + browser.Platform + "\n"
```

```
+ "Supports Cookies = " + browser.Cookies + "\n"
```

```
+ "Supports Java Applets = " + browser.JavaApplets + "\n";
```

}

ADO.NET in ASP.NET

`SqlDataReader` object provides a connection oriented data access to the SQL Server data sources from C# applications.

`ExecuteReader()` in the `SqlCommand` object sends the SQL statements to `SqlConnection` object and populate a `SqlDataReader` object based on the SQL Statement or Stored Procedure.

`SqlDataReader` `sqlReader = SqlCommand.ExecuteReader();`

when the `ExecuteReader` method in the `SqlCommand` object execute, it will instantiate a `SqlClient.SqlDataReader` object. When we started to read from a `DataReader` it should always be open and positioned prior to the first record. The `Read()` method in the `DataReader` is used to read the rows from `DataReader` and it always moves forward to a new valid row, if any row exist.

Program:

```
try
{
    SqlCnn.Open();
    SqlCommand = new SqlCommand(Sql, SqlCnn);
    SqlDataReader SqlReader = SqlCommand.ExecuteReader();
    while (SqlReader.Read())
    {
        MessageBox.Show(SqlReader.GetValue(0) + " - " +
        SqlReader.GetValue(1) + " - " + SqlReader.GetValue(2));
    }
    SqlReader.Close();
    SqlCommand.Dispose();
}
catch (Exception e)
{
    MessageBox.Show("You can not open connection");
}
```

DataReader and GridView

```
SqlDataReader sdr = cmd.ExecuteReader();
gvUserinfo.DataSource = sdr;
gvUserinfo.DataBind();
conn.Close()
```

-----XOK-----

Dataset

Dataset is a disconnected oriented architecture that means there is no need of active connections during work with datasets and it is collection of DataTables and relations between tables. It is used to hold multiple tables with data. You can select data from tables, create views based on tables and use child rows over relations. Also dataset provides you with rich features like saving data as XML and Loading XML data.

Program :

```
protected void BindGridView()
{
    SqlConnection conn = new SqlConnection ("DataSource=
    C:\; Integrated Security=true; Initial Catalog=Test");
    conn.Open();

    SqlCommand cmd = new SqlCommand ("Select
    Username, FirstName, LastName, Location FROM Users", conn);

    SqlDataAdapter sda = new SqlDataAdapter (cmd);
    DataSet ds = new DataSet ();
    sda.Fill(ds);

    gvUserinfo.DataSource = ds;
    gvUserinfo.DataBind();
}
```

How to retrieve and display data from database

- ① Create a SqlConnection object using connection string
 - ② Handle exceptions
 - ③ Open the connection
 - ④ Create a SQL command
 - ⑤ Execute the command (use executeReader)
 - ⑥ Get the result (use SqlDataReader). This is a forward only / readonly data object.
 - ⑦ Close the connection
 - ⑧ Process the result
 - ⑨ Display the result
-

Xox.

Data Column : Each Data Column object has a Datatype property that determines the kind of data the column holds.

Three Types of binding

- Declarative Binding
- Static Binding
- Programmaticaly Binding

Declarative Binding

Declarative data source controls, which were introduced in ASP.NET 2.0 let you access and modify the Scheduler's data with minimal amount of code. Binding RadScheduler to such a data source can be completed without writing any code : You need only configure the datasource and set a few properties to link the scheduler to the data source and indicate which fields the scheduler is to use.

Programmatically data binding of gridview

Program:

```
Protected void Page_Load (object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        SqlDataSource SqlDataSource1 = new SqlDataSource();
        SqlDataSource1.ID = "SqlDataSource1";
        this.page.Controls.Add (SqlDataSource1);
        SqlDataSource1.ConnectionString =
            System.Configuration.ConfigurationManager.ConnectionStrings["connection"];
        SqlDataSource1.SelectCommand = "SELECT top 10 contactname,
            City, Country, PostalCode from Customers";
        GridView1.DataSource = SqlDataSource1;
        GridView1.DataBind();
    }
}
```

Static data binding of dropdown list control

Program:

```
private void Page_Load (Object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        SqlConnection myConn = new SqlConnection("server=
            localhost;Database=pubs;Integrated Security =SSPI");
        SqlCommand myCmd = new SqlCommand("SELECT au_id,
            au_lname FROM Authors", myConn);
        SqlDataReader myReader = myCmd.ExecuteReader();
        AuthorList.DataSource = myReader
        AuthorList.DataTextField = "au_lname";
        AuthorList.DataValueField = "au_id";
        AuthorList.DataBind();
        myConn.Close();
        myReader.Close();
    }
}
```

Declarative Binding

Control can bind records, lists, columns of data into their structure through data source control by creating Connection String for creation of Connection String.

How to create a database through various controls derived from base databound control class is called declarative data binding.

Programmatic Binding

By using ADO connection and command object we can simply insert, update and delete records in database without any control we cannot see database at runtime for this to avoid we can use declaration of programmatic binding together by which simultaneously data can be updated, deleted and viewed by various control.

Difference between Dataset and DataAdapter

Dataset

- i All DML operations are performed only when a database is disconnected
- ii Using Dataset you can accept any no. of values from DB
- iii Dataset is connected to XML
- iv You can see the changes when you refresh DataBase using DataAdapter

DataAdapter

- i It is a bridge between Database and Dataset and it works in a disconnected architecture mode
- ii It's a multipurpose object with its class i.e. we can insert, update, delete in disconnected architecture type.
- iii If It fetches all the data from a database and pass that to dataset.
- iv It shows all the data when database is refreshed in both connected or disconnected model architecture.

Overview of data Binding

Data Binding provides a way for developers to create a link between the controls on a form and the data in their application (their data model). Classically, data binding was used within applications to take advantage of data stored in databases. Windows forms data binding allows you to access data from databases as well as data in other structures such as arrays and collections.

```
private void Form1_Load(object sender, EventArgs e)
{
    SqlDataAdapter = new SqlDataAdapter("Select * from student_detail",
                                         connstring);
    dset = new DataSet();
    SqlDataAdapter.Fill(dset);
    dataGridView1.DataSource = dset.Tables[0].DefaultView;
}
```

Postback

The IsPostBack property can be used to determine if the page is submitted to itself. When a form is submitted back to the same page that contains it, it's called a postback. ASP.NET provides a property called IsPostBack that is TRUE when the page is being loaded as a result of postback, and is FALSE otherwise.

Namespace required for sending email in ASP.NET
System.Net.Mail
System.Text
System.Configuration

Data List

- Datalist is an unformatted Data Control like repeater control in ASP.NET
- Datalist Controls are used to display a list of Items in repeating structure
- The Datalist Control is useful for displaying data in repeating structure
- Eval() and Bind() can be used to bind ~~the~~ data in Datalist.
- Details View
- DetailsView is formatted data control like GridView Control in ASP.NET
- The DetailsView Control displays only a single data record at a time, even if its data source exposes multiple records
- CRUD operations are possible in DetailsView
- asp:BoundField is used to bind data in DetailsView

DataGrid

The DataGrid Control displays the fields of data sources columns in a table. Each row in the control represents a record in the data source. The control supports selection, editing, deleting, paging and sorting.

Form View

Displays the values of a single record from a data source using user defined templates. The form view control allows you to edit, delete and insert records.

Repeater

A databound list control that allows custom layout by repeating a specified template for each item displayed in the list.

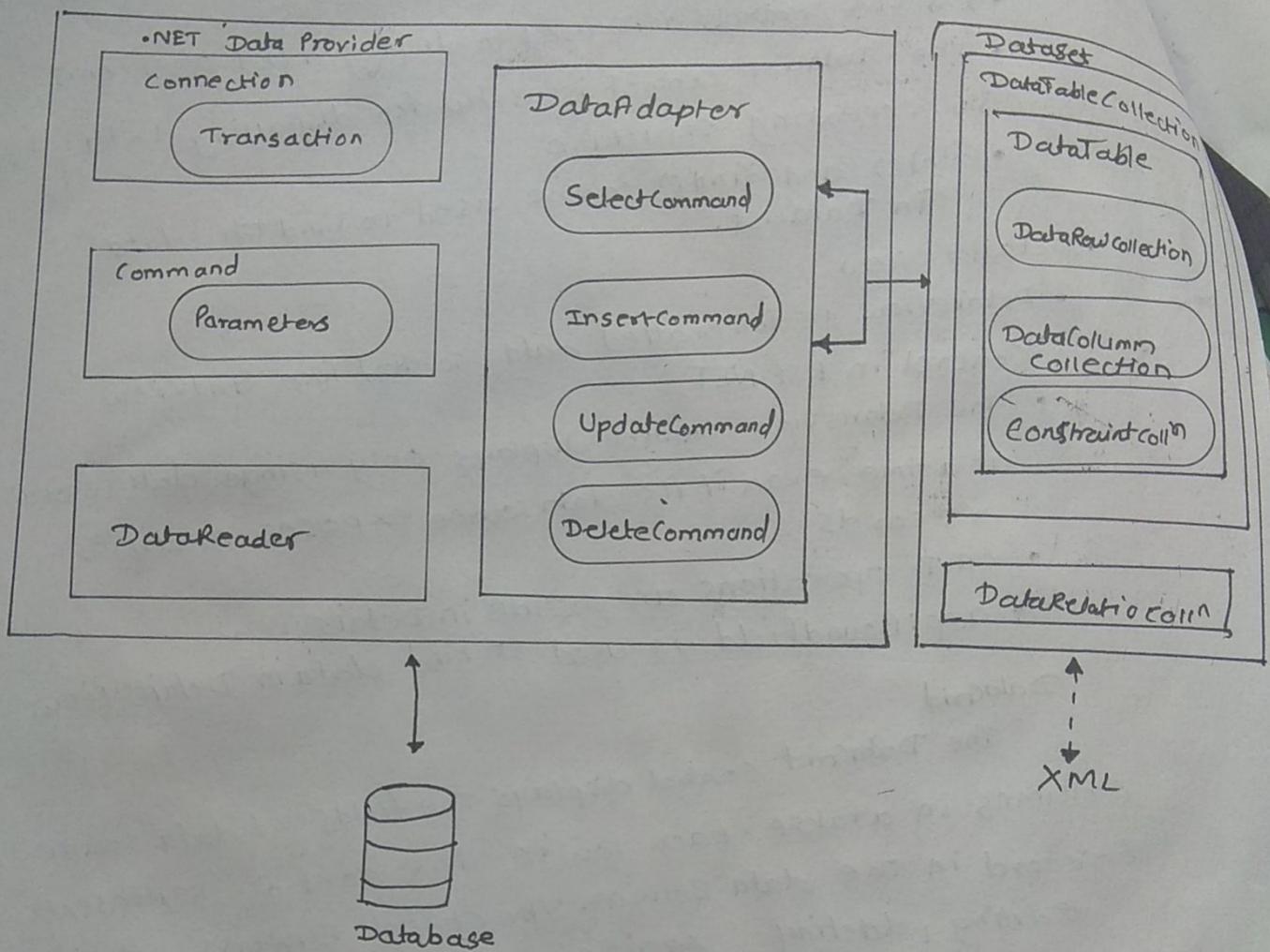


Fig. ADO.NET Architecture

Server control templates available for Data Binding

- Repeater
- DataList and Data Grid view controls
- Detail view control
- Form view control

Compare DataReader and DataAdapter

ASP16

DataReader

- ① It works in connected mode
- ② It can have only one record at a time.
- ③ It is forward only and read only
- ④ It is faster
- ⑤ It handles Database table
- ⑥ It has no storage capacity

DataAdapter

- ① It works in disconnected mode
- ② It can have more than one record
- ③ It can navigate front and back and is editable
- ④ It is slower
- ⑤ It handles XML files, text files and database files
- ⑥ It has temporary storage capacity.