Dot Net Interview Success Kit

Preview of Interview Success Kit Containing 9,000+ Interview Questions & Answers
http://interview-kit.vyomworld.com

If you like this Kit:

This eBook contains 100 Interview Questions & Answers. If you like this eBook, you will definitely love the full version of Interview Success Kit – which contains 9,000 Interview Questions & Answers.

Read more about the Complete Interview Success kit at

http://interview-kit.vyomworld.com/

Interview Success Kit is a complete Interview Cheat-sheet,
written by 58 Subject Experts.

DOT NET INTERVIEW QUESTIONS

By www.Coolinterview.com

Dot Net Study Materials

1. How do you turn off cookies for one page in your site?

Use the Cookie. Discard Property which Gets or sets the discard flag set by the server. When true, this property instructs the client application
not to save the Cookie on the users hard disk when a session ends.
Or
it can be turned off by mentioning cookie state= false in web.config file

2. How u can create XML file?

To write Dataset Contents out to disk as an XML file use:
MyDataset.WriteXML(server.MapPath("MyXMLFile.xml"))

3. What is CLR?

CLR is .NET equivalent of Java Virtual Machine (JVM). It is the runtime that converts a MSIL code into the host machine language code, which is then executed appropriately. The CLR is the execution engine for .NET Framework applications. It provides a number of services, including:
Code management (loading and execution)
Application memory isolation
Verification of type safety
Conversion of IL to native code.
Access to metadata (enhanced type information)
Managing memory for managed objects
Enforcement of code access security
Exception handling, including cross-language exceptions
Interoperation between managed code, COM objects, and pre-existing DLL's (unmanaged code and data)
Automation of object layout
Support for developer
Services (profiling, debugging, and so on).

4. Which two properties are on every validation control?

Control to validate & Error Message.

5. What Is The Difference Between ViewState and SessionState?

View State persist the values of controls of particular page in the client (browser) when post back operation done. When user requests another page previous page data no longer available.
SessionState persist the data of particular user in the server. This data available till user close the browser or session time completes.

6. What is CTS and CLS?

=>cls stands for common language specification where as
cts stands for common type system.

=>CLS : cls is a standard for .net . cls is small set of specification to make all languages as a .net compliant languages. cls make a use of cts and clr. if my languages (c#,vb.net,j#,vc++) wants to be compliant language it has to follow cls standard.
=>CTS : Common Type System is also a standard like cls. If two languages (c# or vb.net or j# or vc++) wants to communicate with eachother, they have to convert into some common type (i.e in clr common language runtime). In c# we use int which is converted to Int32 of CLR to communicate with vb.net which uses Integer or vice versa.

7. Which property on a Combo Box do you set with a column name, prior to setting the Data Source, to display data in the combo box?
ComboBox.DataValueField = "ColumnName"

When we use Databind method for the Combo box we set Display Member and Display Value property to column name.

8. In .NET Compact Framework, can I free memory explicitly without waiting for garbage collector to free the memory?

Yes you can clear the memory using gc.collect method but it is recommended that u should not call this coz we don't know the exact time when the gc will be called automatic.

9. What is "Common Type System" (CTS)?
CTS define all of the basic types that can be used in the .NET Framework and the operations performed on those type. All this time we have been talking about language interoperability, and .NET Class Framework. None of this is possible without all the language sharing the same data types. What this means is that an int should mean the same in VB, VC++, C# and all other .NET compliant languages. This is achieved through introduction of Common Type System (CTS).

10. What is the difference between the C#.NET and VB.NET?
VB.NET
- It didn't have the XML Documentation.
- It didn't have the Operator Overloading.
- It didn't have the Pointer Type variables.
C#.NET
- It has XML Documentation, Operator Overloading and supports Pointer Variables using unsafe keyword

11. How do you create a permanent cookie?
By setting the expiry date of the cookie to a later time (like 10 years later.)

12. Why is catch(Exception) almost always a bad idea?
Well, if at that point you know that an error has occurred, then why not write the proper code to handle that error instead of passing a new Exception object to the catch block? Throwing your own exceptions signifies some design flaws in the project.

13. What tags do you need to add within the asp datagrid tags to bind columns manually?
<asp:BoundColumn> is the tag you usually use to add a bound column. You may also use
<asp:TemplateColumn>

<ItemTemplate><%#...%></ItemTemplate>
</asp:TemplateColumn>
to add a bound column to a datagrid.

14. What tags do you need to add within the asp:datagrid tags to bind columns manually?
Inside tag in asp:Datagrid we set Text attribute to<%= #container.Eval (databinder.dataitem,"Col_Name") %>

15. What tag do you use to add a hyperlink column to the DataGrid?
Anchor tag is used to add a hyperlink column to the data grid

16. How does VB.NET/C# achieve polymorphism?
VB.Net / C# provide polymorphism through the following mechanisms:
1. Inheritance - base class provides overridable methods which are re-implemented in derived classes.
2. Interface - different class can implement same interface, resulting in different implementations of interface methods.
3. Virtual methods - achieve runtime polymorphism.

17. Can a .NET web application consume Java web service?
Yes Offcourse.Actually Webservices are independent to language. it depends on WSDL and SOAP. so any one can develope the Webservices anddisclose the wsdl and users can consume the webservices.wsdl and soap both are xml based.. and all languages having xml parsing capability and access to http protocol will be able to work with Webservices.

18. What are value types and reference types?
Value type - bool, byte, chat, decimal, double, enum , float, int, long, sbyte, short, strut, uint, ulong, ushort. Value types are stored in the Stack
Reference type - class, delegate, interface, object, string. Reference types are stored in the Heap

19. Which control cannot be placed in MDI?
The controls that do not have events.

20. Can we run DOT.NET in UNIX platform?
One of the disadvantages of using Visual Studio.NET and the .NET framework to develop applications has been the lack of cross-platform support. Since the introduction of the .NET framework and common language run time a few years ago, there have been a few projects designed to bring the .NET framework to other platforms, including Linux and UNIX. The DotGNU project is touted as the "Free software alternative to .NET" and encompasses a number of projects, including DotGNU Portable .NET, which is designed to be used to compile and run C# and C applications on a multitude of platforms, including GNU/Linux, FreeBSD, Mac OS X, and Windows.One of the main features of the product is it's compatibility with EMCA standards for C# and the Common Language Infrastructure (CLI), as well as Microsoft's own CLI implementation in the .NET framework. The project chose to go with a "Virtual Machine" implementation, where bytecode is transformed into a simple instruction set which is then passed to a "Converted Virtual Machine", which then are executed through an interpreter. This approach is different to other open source .NET implementations, but provides more flexibility when porting the project to other platforms.At the core of the project is the runtime engine (ilrun) and compiler (cscc) with an implementation of System.Windows.Forms that make

developing for the platform easier, as it doesn't required translation through another toolkit or toolset

21. What are the Types of Assemblies?

One of the disadvantages of using Visual Studio.NET and the .NET framework to develop applications has been the lack of cross-platform support. Since the introduction of the .NET framework and common language run time a few years ago, there have been a few projects designed to bring the .NET framework to other platforms, including Linux and UNIX. The DotGNU project is touted as the "Free software alternative to .NET" and encompasses a number of projects, including DotGNU Portable .NET, which is designed to be used to compile and run C# and C applications on a multitude of platforms, including GNU/Linux, FreeBSD, Mac OS X, and Windows. One of the main features of the product is it's compatibility with EMCA standards for C# and the Common Language Infrastructure (CLI), as well as Microsoft's own CLI implementation in the .NET framework. The project chose to go with a "Virtual Machine" implementation, where bytecode is transformed into a simple instruction set which is then passed to a "Converted Virtual Machine", which then are executed through an interpreter. This approach is different to other open source .NET implementations, but provides more flexibility when porting the project to other platforms. At the core of the project is the runtime engine (ilrun) and compiler (cscc) with an implementation of System.Windows.Forms that make developing for the platform easier, as it doesn't required translation through another toolkit or toolset

22. What is .NET / .NET Framework?

It is a Framework in which Windows applications may be developed and run. The Microsoft .NET Framework is a platform for building, deploying, and running Web Services and applications. It provides a highly productive, standards-based, multi-language environment for integrating existing investments with next-generation applications and services as well as the agility to solve the challenges of deployment and operation of Internet-scale applications. The .NET Framework consists of three main parts: the common language runtime, a hierarchical set of unified class libraries, and a componentized version of Active Server Pages called ASP.NET. The .NET Framework provides a new programming model and rich set of classes designed to simplify application development for Windows, the Web, and mobile devices. It provides full support for XML Web services, contains robust security features, and delivers new levels of programming power. The .NET Framework is used by all Microsoft languages including Visual C#, Visual J#, and Visual C++.

23. How many classes can a single .NET DLL contain?

Unlimited

24. What is the difference between thread and process?

Thread - is used to execute more than one program at a time.

process - executes single program

A thread is a path of execution that run on CPU, a process is a collection of threads that share the same virtual memory. A process has at least one thread of execution, and a thread always run in a process context.

25. How do you turn off cookies for one page in your site?

By setting the Cookie. Discard property false.

26. How to get the column count of a report?

SY-LINSZ system variable gives the column count (line size) and SY-LINCT for line count.

27. **What is Full Trust?**

Your code is allowed to do anything in the framework, meaning that all (.Net) permissions are granted. The GAC has Full Trust because it's on the local HD, and that has Full Trust by default, you can change that using caspol .

28. What is cyclomatic complexity and why is it important?

Cyclomatic complexity is a computer science metric (measurement) developed by Thomas McCabe used to generally measure the complexity of a program. It directly measures the number of linearly independent paths through a programs source code.

The concept, although not the method, is somewhat similar to that of general text complexity measured by the Flesch-Kincaid Readability Test.

Cyclomatic complexity is computed using a graph that describes the control flow of the program. The nodes of the graph correspond to the commands of a program. A directed edge connects two nodes, if the second command might be executed immediately after the first command. By definition,

$CC = E - N + P$

where

CC = cyclomatic complexity

E = the number of edges of the graph

N = the number of nodes of the graph

P = the number of connected components

29. What is the Scope of public/private/friend/protected/protected friend?

Scope of public/private/friend/protected/protected friend.

Visual Basic/Visual C#

Public/public All members in all classes and projects.

Private/private Members of the current class only.

Friend/internal All members in the current project.

Protected/protected All members in the current class and in classes derived from this member's class. Can be used only in member definitions, not for class or module definitions.

Protected Friend/protected internal All members in the current project and all members in classes derived from this member's class. Can be used only in member definitions, not for class or module definitions.

30. What is Machine.config?

Machine configuration file: The machine. config file contains settings that apply to the entire computer. This file is located in the %runtime install path%Config directory. There is only one machine. config file on a computer. The Machine.Config file found in the "CONFIG" subfolder of your .NET Framework install directory (c:WINNTMicrosoft.NETFramework{Version Number}CONFIG on

Windows 2000 installations). The machine.config, which can be found in the directory $WINDIR$Microsoft.NETFrameworkv1.0.3705CONFIG, is an XML-formatted configuration file that specifies configuration options for the machine. This file contains, among many other XML elements, a browserCaps element. Inside this element are a number of other elements that specify parse rules for the various User-Agents, and what properties each of these parsings supports.

For example, to determine what platform is used, a filter element is used that specifies how to set the platform property based on what platform name is found in the

User-Agent string. Specifically, the machine.config file contains:

```
<filter>

<case match="Windows 95|Win95">

platform=Win95

</case>

<case match="Windows 98|Win98">

platform=Win98
```

```
</case>

<case match="Windows NT|WinNT">

platform=WinNT

</case>

...

</filter>
```

That is, if in the User-Agent string the string "Windows 95" or "Win95" is found, the platform property is set to Win95. There are a number of filter elements in the browserCaps element in the machine.config file that define the various properties for various User-Agent strings.

Hence, when using the Request. Browser property to determine a user's browser features, the user's agent string is matched up to particular properties in the machine.config file. The ability for being able to detect a user's browser's capabilities, then, is based upon the honesty in the browser's sent User-Agent string. For example, Opera can be easily configured to send a User-Agent string that makes it appear as if it's IE 5.5. In this case from the Web server's perspective (and, hence, from your ASP.NET Web page's

perspective), the user is visiting using IE 5.5, even though, in actuality, he is using Opera.

31. What is the difference between "using System.Data;" and directly adding the reference from "Add References Dialog Box"?

When you compiles a program using command line, u add the references using /r switch. When you compile a program using Visual Studio, it adds those references to our assembly, which are added using "Add Reference" dialog box. While "using" statement facilitates us to use classes without using their fully qualified names.

For example: if u have added a reference to "System.Data.SqlClient" using "Add Reference" dialog box then u can use SqlConnection class like this:

```
System.Data.SqlClient.SqlConnection
```

But if u add a "using System.Data.SqlClient" statement at the start of ur code then u can directly use SqlConnection class.

On the other hand if u add a reference using "using System.Data.SqlClient" statement, but don't add it using "Add Reference" dialog box, Visual Studio will give error message while we compile the program.

32. How to clear a datagrid on a button click?

You need to Clear the DataSource of the dadaGrid.

So try this:

```
dataSet1.Clear();

dataGrid1.DataSource = dataSet1.TableNameHere.DefaultView;
```

or

```
C#: dataGrid1.DataSource = null;

VB: dataGrid1.DataSource = nothing
```

33. With respect to security, which one is the better choice?.Net or J2EE? Explain?

As per majority programmers .NET is the best one which have single vendor compare to, the ease of use, more use friendly, you can send any problem ,it'll be solved.

34. How can I read .doc document in ASP.Net?

You can read from a text file like this.

```
private void Button12_Click(object sender, System.EventArgs e)
{
string path="C:Inetpub ew1.txt";
using(StreamReader reader=new StreamReader(path))
{
string line;
while ((line=reader.ReadLine())!=null)
{
Label2.Text+="<br>"+line;
}
```

}

}

from .doc file try yourself

35. Should validation (did the user enter a real date) occur server-side or client-side? Why?

Validation will be done in both sides i.e., at the server side and client side. Server side validation will be done even though client side validation performed on the code because of proper security reasons.

36. Briefly explain how the server control validation controls work?

A validation control works by evaluating the value of an input server control on the page to see whether it meets certain criteria. The input control to evaluate is specified through the validation controls Control To Validate attribute. You can programmatically cause a validation control to perform its evaluation by calling its Validate method, which sets the validation controlls IsValid property to true if validation succeeds and false if it fails. You can cause all of the validation controls on a page to validate their input controls by calling the Page.Validate method.

The Page Is Valid property is set to true if validation succeeds for all validation controls within the page; otherwise, it's false. The Page.Validate method is automatically invoked when a button control whose CausesValidation property is set to true is clicked. Causes Validation is set to false by default for a reset or clear button. The button click event is suppressed if IsValid is false for any validation control on the page.

There are following types of validation controls provided by ASP.Net:

1. Required Field Validator

2. Compare Validator

3. Range Validator

4. Regular Expression Validator

5. Custom Validator

These controls are server side controls. However in up-level browsers that is IE 4.0 and above these can be enabled on client side also.

All these controls have a property called Control To Validate. Required Field Validator ensures data entry into the Control To Validate.

Compare Validator uses Control To Validate, Control To Compare and operator properties to compare a control's value with another control's value. It also has a property 'Value To Compare' for comparing Control To Validate against a constant value.

Range Validator uses Control To Validate, Maximum Value and Minimum Value to compare a controls value against a range.

Regular Expression Validator uses Control To Validate and Validation Expression properties.

Custom Validator performs its task by using Client Validation Function and OnServer Validate delegate.

37. How does the Xml Serializer work? What ACL permissions does a process using it require?

Xml Serializer requires write permission to the system's TEMP directory.

38. Difference between Panel and Group Box classes?

Panel & Group box both can used as container for other controls like radio buttons & check box.

the difference in panel & group box are

Panel

1) In case of panel captions cannot be displayed

2) Can have scroll bars.

Group box

1) Captions can be displayed.

2) Cannot have a scroll bar

39. How would ASP and ASP.NET apps run at the same time on the same server?

Both ASP and ASP.net can be run at same server, becuase IIS has the capability to respond/serve both ASP and ASP.NET request.

40. What is the Main difference between ASP and ASP.NET ?

ASP contains scripts which are not compiled where as in ASP.net the code is compiled.

41. What exactly is being serialized when you perform serialization?

Serialization is the process of converting an object into stream of bytes. We perform it at the time of trans port an object in remoting.

42. What is a Manifest?

An assembly manifest contains all the metadata needed to specify the assembly's version requirements and security identity, and all metadata needed to define the scope of the assembly and resolve references to resources and classes. The assembly manifest can be stored in either a PE (Portable Executable) file (an .exe or .dll)

with Microsoft intermediate language (MSIL) code or in a standalone PE (Portable Executable) file that contains only assembly manifest information. The following table shows the information contained in the assembly manifest. The first four items — the assembly name, version number, culture, and strong name information — make up the assembly's identity. Assembly name: A text string specifying the assembly's name. Version number: A major and minor version number, and a revision and build number. The common language runtime uses these numbers to enforce version policy.

Culture: Information on the culture or language the assembly supports. This information should be used only to designate an assembly as a satellite assembly containing culture- or language-specific information. (An assembly with culture information is automatically assumed to be a satellite assembly.) Strong name information: The public key from the publisher if the assembly has been given a strong name. List of all files in the assembly: A hash of each file contained in the assembly and a file name. Note that all files that make up the assembly must be in the same directory

as the file containing the assembly manifest. Type reference information: Information used by the runtime to map a type reference to the file that contains its declaration and implementation. This is used for types that are exported from the assembly.

Information on referenced assemblies: A list of other assemblies that are statically referenced by the assembly. Each reference includes the dependent assembly's name, assembly metadata (version, culture, operating system, and so on), and public key, if the assembly is strong named.

43. Can you give an example of what might be best suited to place in the application_Start and Session_Start subroutines?

Application
Start - We can place code to initialize variables once during application start. (e.g., db connection string)

Session Start - We can place code to initialize variables specific to the session (e.g., USER ID and other specific info related to this session)

44. Briefly explain how server form post-back works ?

Post Back: The process in which a Web page sends data back to the same page on the server.

View State: View State is the mechanism ASP.NET uses to keep track of server control state values that don't otherwise post back as part of the HTTP form.

View State Maintains the UI State of a Page View State is base64-encoded. It is not encrypted but it can be encrypted by setting Enable View Stat MAC="true" & setting the machine Key validation type to 3DES. If you want to NOT maintain the View State, include the directive < %@ Page Enable View State="false" % > at the top of an .aspx page or add the attribute Enable View State="false" to any control.

45. What is the purpose of DOTNET?

Dot Net Is a collection of products like C#, ASp.Net, Vb.Net, XML. The purpose of .Net is that we can easily upgrade and degrade our programs which is written in Microsoft visual studio program. It provides user friendly environment.

46. What is the transport protocol you use to call a Web service SOAP?

Http is preferred for Soap while tcp for binary i.e. HTTP is used in web services and tcp works well in remoting.

47. Can you explain what inheritance is and an example of when you might use it?

The process of deriving a new class from an existing class is called Inheritance. The old class is called the base class and the new class is called derived class. The derived class inherits some or everything of the base class. In Visual Basic we use the Inherits keyword to inherit one class from other.

Ex:

```
Public Class Base

---

---

End Class

Public Class Derived

Inherits Base

---

End Class
```

48. What is strong-typing versus weak-typing? Which is preferred? Why?

Strong type is checking the types of variables as soon as possible, usually at compile time. While weak typing is delaying checking the types of the system as late as possible, usually to run-time. Which is preferred depends on what you want. For scripts & quick stuff you'll usually want weak typing, because you want to write as much less (is this a correct way to use Ensign?) code as possible. In big programs, strong typing can reduce errors at compile time.

49. Whats wrong with a line like this? DateTime.Parse(myString).

The result returned by this function is not assigned to anything, should be something like

var x = DateTime.Parse(myString)

50. What is CLR?

CLR(Common Language Runtime) is the main resource of .Net Framework. it is collection of services like garbage collector, exception handler, jit compilers etc. with the CLR cross language integration is possible.

51. Explain Dataset Accept Changes and Data Adapter Update methods?

Data Adapter Update method Calls the respective INSERT, UPDATE, or DELETE statements for each inserted, updated, or deleted row in the Data Set.

Data Set Accept Changes method Commits all the changes made to this row since the last time Accept Changes was called.

52. Differences between Datagrid, Datalist and Repeater?

1. Datagrid has paging while Datalist doesnt.

2. Datalist has a property called repeat. Direction = vertical/horizontal. (This is of great help in designing layouts). This is not there in Datagrid.

3. A repeater is used when more intimate control over html generation is required.

4. When only checkboxes/radiobuttons are repeatedly served then a checkboxlist or radiobuttonlist are used as they involve fewer overheads than a Datagrid.

The Repeater repeats a chunk of HTML you write, it has the least functionality of the three. DataList is the next step up from a Repeater; accept you have very little control over the HTML that the control renders. DataList is the first of the three controls that allow you Repeat-Columns horizontally or vertically. Finally, the DataGrid is

the motherload. However, instead of working on a row-by-row basis, you're working on a column-by-column basis. DataGrid caters to sorting and has basic paging for your disposal. Again you have little contro, over the HTML. NOTE: DataList and DataGrid both render as HTML tables by default. Out of the 3 controls, I use the Repeater the most due to its flexibility w/ HTML. Creating a Pagination scheme isn't that hard, so I rarely if ever use a DataGrid.

Occasionally I like using a DataList because it allows me to easily list out my records in rows of three for instance.

53. How to use ActiveX Control in .Net ?

ActiveX control is a special type of COM component that supports a User Interface. Using ActiveX Control in your .Net Project is even easier than using COM component. They are bundled usually in .ocx files. Again a proxy assembly is made by .Net utility AxImp.exe (which we will see shortly) which your application (or client) uses as if it is a .Net control or assembly.

• Making Proxy Assembly For ActiveX Control: First, a proxy assembly is made using AxImp.exe (acronym for ActiveX Import) by writing following command on

Command Prompt:

C:>AxImp C:MyProjectsMyControl.ocx

This command will make two dlls, e.g., in case of above command

MyControl.dll

AxMyControl.dll

The first file MyControl.dll is a .Net assembly proxy, which allows you to reference the ActiveX as if it were non-graphical object.

The second file AxMyControl.dll is the Windows Control, which allows u to use the graphical aspects of activex control and use it in the Windows Form Project.

• Adding Reference of ActiveX Proxy Assembly in your Project Settings: To add a reference of ActiveX Proxy Assembly in our Project, do this:

o Select Project?d Reference… (Select Add Reference from Project Menu).

o This will show you a dialog box, select .Net tab from the top of window.

o Click Browse… button on the top right of window.

o Select the dll file for your ActiveX Proxy Assembly (which is MyControl.dll) and click OK

o Your selected component is now shown in the 'Selected Component' List Box. Click OK again

Some More On Using COM or ActiveX in .Net

• .Net only provides wrapper class or proxy assembly (Runtime Callable Wrapper or RCW) for COM or activeX control. In the background, it is actually delegating the tasks to the original COM, so it does not convert your COM/activeX but just imports them.

• A good thing about .Net is that when it imports a component, it also imports the components that are publically referenced by that component. So, if your component, say MyDataAcsess.dll references ADODB.dll then .Net will automatically import that COM component too!

• The Visual Studio.NET does surprise you in a great deal when u see that it is applying its intellisense (showing methods, classes, interfaces, properties when placing dot) even on your imported COM components!!!! Isn't it a magic or what?

• When accessing thru RCW, .Net client has no knowledge that it is using COM component, it is presented just as another C# assembly.

• U can also import COM component thru command prompt (for reference see Professional C# by Wrox)

• U can also use your .Net components in COM, i.e., export your .net components (for reference see Professional C# by Wrox).

54. How different are interface and abstract class in .Net?

Abstract classes can not be instantiated it can have or cannot have abstract method basically known as mustinherit as the methods are static in nature

where interfaces are the declaration and r defined where they are called used for dynamic methods

55. Briefly explain what user controls are and what server controls are and the differences between the two.

An ASP.NET control (sometimes called a server control) is a server-side component that is shipped with .NET Framework. A server control is a compiled DLL file and cannot be edited. It can, however, be manipulated through its public properties at design-time or runtime. It is possible to build a custom server control (sometimes called a custom control or composite control). (We will build these in part 2 of this article.)

In contrast, a user control will consist of previously built server controls (called constituent controls when used within a user control). It has an interface that can be completely edited and changed. It can be manipulated at design-time and runtime via properties that you are responsible for creating. While there will be a multitude of controls for every possible function built by third-party vendors for ASP.NET, they will exist in the form of compiled server controls, as mentioned above. Custom server controls may be the .NET answer to ActiveX Web controls.

56. Which template must you provide, in order to display data in a Repeater control?

ItemTemplate.To display data in the ItemTemplate, declare one or more Web server controls and set their data-binding expressions to evaluate to a field in the Repeater control's (that is, the container control's) DataSource.

57. Which method do you invoke on the DataAdapter control to load your generated dataset with data?

dataAdapter.Fill(ds). The beauty of this method is it automatically implicitly opens the connection to database and closes it once done.We dont need to worry about opening and closing the connection to the database.

58. In what order do the events of an ASPX page execute. As a developer is it important to understand these events?

if you try to access it in Page_Load that is way earlier than Control's own Render method. See ther Page/control lifecycle goes like this for the Page and controls (being already in the controls collection, dynamic ones play catchup)

1. Instantiate

2. Initialize

3. TrackViewState

4. LoadViewState (postback)

5. Load postback data (postback, IPostBackDatahandler.LoadPostdata)
6. Load

7. Load postback data for dynamical controls added on Page_Load (postback)
8. Raise Changed Events (postback, IPostBack Datahandler. RaisePost Data Changed)
9. Raise postback event (postback, IPostBackEventHandler.RaisePostBackEvent)
10.PreRender
11. SaveViewState

12. Render

13. Unload

14. Dispose null

59. What is assembly?

Assemblies are the building blocks of .NET Framework applications; they form the fundamental unit of deployment, version control, reuse, activation scoping, and security permissions. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. An assembly provides the common language runtime with the information it needs to be aware of type implementations. To the runtime, a type does not exist outside the context of an assembly.Assemblies are a fundamental part of programming with the .NET Framework. An assembly performs the following functions: ? It contains code that the common language runtime executes. Microsoft intermediate language (MSIL) code in a portable executable (PE) file will not be executed if it does not have an associated assembly manifest. Note that each assembly can have only one entry point (that is, DllMain, WinMain, or Main). ? It forms a security boundary. An assembly is the unit at which permissions are requested and granted. ? It forms a type boundary. Every type's identity includes the name of the assembly in which it resides. A type called MyType loaded in the scope of one assembly is not the same as a type called MyType loaded in the scope of another assembly. ? It forms a reference scope boundary. The assembly's manifest contains assembly metadata that is used for resolving types and satisfying resource requests. It specifies the types and resources that are exposed outside the assembly. The manifest also enumerates other assemblies on which it depends. ? It forms a version boundary. The assembly is the smallest versionable unit in the common language runtime; all types and resources in the same assembly are versioned as a unit. The assembly's manifest describes the version dependencies you specify for any dependent assemblies. ? It forms a deployment unit. When an application starts, only the assemblies that the application initially calls must be present. Other assemblies, such as localization resources or assemblies containing utility classes, can be retrieved on demand. This allows applications to be kept simple and thin when first downloaded. ? It is the unit at which side-by-side execution is supported. Assemblies can be static or dynamic. Static assemblies can include .NET Framework types (interfaces and classes), as well as resources for the assembly (bitmaps, JPEG files, resource files, and so on). Static assemblies are stored on disk in PE files. You can also use the .NET Framework to create dynamic assemblies, which are run directly from memory and are not saved to disk before execution. You can save dynamic assemblies to disk after they have executed.There are several ways to create assemblies. You can use development tools, such as Visual Studio .NET, that you have used in the past to create .dll or .exe files. You can use tools provided in the .NET Framework SDK to create assemblies with modules created in other development environments. You can also use common language runtime APIs, such as Reflection.Emit, to create dynamic assemblies.

60. What are the differences between ASP and ASP .Net ?

1. ASP: Code is Interpreted

ASP.NET: Code is Compiled

2. ASP: Business Logic and Presentation Logic are in a single file

ASP.NET: Business Logic and Presentation Logic are in separate files (.cs or

.vb) and (.aspx) respectively.

3. ASP: No Web Server Controls

ASP.NET: Web Server Controls supported by strong .NET Framework

4. ASP: No RAD* in Classic ASP

ASP.NET: Supports RAD

____

* RAD ? Rapid Application Development

Or

1.Asp is interpreted

Asp.net is compiled which is faster than asp.

2 Asp.net maintains its own CLR and is managed as it runs by CLR

Where as asp is unmanaged

3 We can mainatin sessions in state server and sql server which is Outproc,

where in asp sessions will be last if we restart webserver or make changes.

4 In asp.net we can configure each application using web.config file which is availble in application itself and we have machine.config wherer we can configure all applications.

In asp we cannot configure single aplication

5 Asp.net we have autopostback event which is not in asp

6 In asp.net we have global.asax where can hadle some global things which is not in asp.

7 We have well built GUI to work in asp.net

8 We have ado.net and as well as disconnected architecture in asp.net

9 We have Xcopy deployment in asp.net

10. We can work with any language as code behind technique in asp.net that supports .net frame work

61. What are Assemblies?

Assemblies are similar to dll files. Both has the reusable pieces of code in the form of classes/ functions. Dll needs to be registered but assemblies have its own metadata.

Or

Assembly is a single deployable unit that contains information about the implementation of classes, structures and interfaces. it also stores the information about itself called metadata and includes name and verison of the assembly, security information, information about the dependencies and the list of files that constitute the assembly.

Assembly also contains namespaces. In the .Net Framework, applications are deployed in the form of assemblies.

Or

An assembly is a single deployable unit that contains all the information about the implementation of :

- classes

- structures and

- interfaces

An assembly stores all the information about itself. This information is called METADATA and include the name and the verison number of the assembly, security information, information about the dependencies and a lost of files that constitute the assembly.

All the application developed using the .NET framework are made up of assemblies.

Namespaces are also stored in assemblies

62. What is serialization, how it works in .NET?

Serialization is when you persist the state of an object to a storage medium so an exact copy can be re-created at a later stage.
Serialization is used to save session state in ASP.NET.
Serialization is to copy objects to the Clipboard in Windows Forms
Serialization is used by remoting to pass objects by value from one application domain to another.

63. Explain how Viewstate is being formed and how it is stored on client.

The type of ViewState is System.Web.UI.StateBag, which is a dictionary that stores name/value pairs. ViewState is persisted to a string variable by the ASP.NET page framework and sent to the client and back as a hidden variable. Upon postback, the page framework parses the input string from the hidden variable and populates the ViewState property of each control. If a control uses ViewState for property data instead of a private field, that property automatically will be persisted across round trips to the client. (If a property is not persisted in ViewState, it is good practice to return its default value on postback.)

64. What is the root class in .Net ?

Object

65. Which property of the textbox cannot be changed at runtime?

Locked Porperty.

66. With respect to security, which one is the better choice?Net or J2EE? Explain.

As per majority programmers .NET is the best one whch have single vendor compare to ,the eace of use, more use friendly can send any problem ,it'll be solved.

67. What is Delegation?

A delegate acts like a strongly type function pointer. Delegates can invoke the methods that they reference without making explicit calls to those methods. Delegate is an entity that is entrusted with the task of representation, assign or passing on information. In code sense, it means a Delegate is entrusted with a Method to report information back to it when a certain task (which the Method expects) is accomplished outside the Method's class.

68. What is Reflection?

It extends the benefits of metadata by allowing developers to inspect and use it at runtime. For example, dynamically determine all the classes contained in a given assembly and invoke their methods.

Reflection provides objects that encapsulate assemblies, modules, and types. You can use reflection to dynamically create an instance of a type, bind the type to an existing object, or get the type from an existing object. You can then invoke the type's methods or access its fields and properties.

Namespace: System.Reflection

69. How is a property designated as read-only?

In VB.NET: Public Read-Only Property Property Name As Return Type Get ?Your Property Implementation goes in here End Get End Property.

In C# if you specify a get accessor but do not specify a set accessor for a property it becomes a read only propert

70. In what order do the events of an ASPX page execute. As a developer is it important to understand these events?

1. Object Initialization

2. Load View state Data

3. Load Post Data Processes Post back Data

4. Object Load

5. Raise Post Back Change Events

6. Process Client-Side Post Back Event

7. Prerender the Objects

8. View State Saved

9. Render To HTML

10. Disposal

71. Describe session handling in a web farm, how does it work and what are the limits ?

In ASP.NET there are three ways to manage session objects. one support the in-proc mechanism and other two's support the out-proc mechanism.

1) In-Proc (By Default)

2) SQL-Server (Out-proc) - Using SQL server or any other database for storing sessions regarding current logged in user.

3) State-Server (Out-Proc) - Using State Server, as one dedicated server for managing sessions. State Server will run as service on web server having dotnet installed.

72. Can you give an example of when it would be appropriate to use a web service as opposed to a non-serviced .NET component?

A web service has the following characteristics:

1. It communicates using open protocols like HTTP

2. Processes XML messages framed using SOAP

3. May be described using WSDL

4. Can be discovered using UDDI

Any application which is supposed to reach a wide customer base should be written as a web service, as opposed to customized applications for specific customers.

For example, services which help in stock trading by providing analysis of market trends could best be implemented as a web service. The reasons are

1. Such a web service would be easily discovered by potential customers when they need it.

2. It will require minimal setup on the client machines

3. The updates to the system will be automatically available to all its consumers, without any need to deploy the updates on their machines.

73. Why The JavaScript Validation Not Run on the Asp.Net?

The Asp.Net Button Is post backed on the server & not yet Submit & when It goes to the server its states is lost So if we r using JavaScript in our application so we always use the Input Button in the asp Button.

74. How does output caching work in ASP.NET?

Output caching is a powerful technique that increases request/response throughput by caching the content generated from dynamic pages. Output caching is enabled by default, but output from any given response is not cached unless explicit action is taken to make the response cacheable.
To make a response eligible for output caching, it must have a valid expiration/validation policy and public cache visibility. This can be done using either the low-level Output Cache API or the high-level @ Output Cache directive. When output caching is enabled, an output cache entry is created on the first GET request to the page. Subsequent GET or HEAD requests are served from the output cache entry until the cached request expires.
The output cache also supports variations of cached GET or POST name/value pairs.
The output cache respects the expiration and validation policies for pages. If a page is in the output cache and has been marked with an expiration policy that indicates that the page expires 60 minutes from the time it is cached, the page is removed from the output cache after 60

minutes. If another request is received after that time, the page code is executed and the page can be cached again. This type of expiration policy is called absolute expiration - a page is valid until a certain time.

75. What are the authentication methods in .NET?

1.WINDOWS AUTHENTICATION

2.FORMS AUTHENTICATION

3.PASSPORT AUTHENTICATION

4.NONE/CUSTOM AUTHENTICATION

The authentication option for the ASP.NET application is specified by using the tag in the Web.config file, as shown below:

other authentication options

1. WINDOWS AUTHENTICATION Schemes

I. Integrated Windows authentication

II. Basic and basic with SSL authentication

III. Digest authentication

IV. Client Certificate authentication

2. FORMS AUTHENTICATION

You, as a Web application developer, are supposed to develop the Web page and authenticate the user by checking the provided user ID and password against some user database

3.PASSPORT AUTHENTICATION

A centralized service provided by Microsoft, offers a single logon point for clients. Unauthenticated users are redirected to the Passport site

4 NONE/CUSTOM AUTHENTICATION:

If we don?t want ASP.NET to perform any authentication, we can set the authentication mode to ?none?.

The reason behind this decision could be: We don?t want to authenticate our users, and our Web site is open for all to use. We want to provide our own custom authentication.

76. Parts

event name Required. Name of the event to trigger.

Argument list Optional. Comma-delimited list of variables, arrays, or expressions. The argument list argument must be enclosed by parentheses. If there are no arguments, the parentheses must be omitted.

77. What is the difference between proc. sent BY VAL and By Ref?

BY VAL: changes will not be reflected back to the variable.

By REF: changes will be reflected back to that variable.(same as & symbol in c, c++)

78. How to Creating a Key Pair ?

You can create a key pair using the Strong Name tool (Sn.exe). Key pair files usually have an .snk extension.

To create a key pair At the command prompt, type the following command:

sn –k <file name>

In this command, file name is the name of the output file containing the key pair.

The following example creates a key pair called sgKey.snk.

sn -k sgKey.snk

79. Can implement same method name in both base class And derived class with different action?

method overloading is possible. suppose baseclass class class1 is having method called getname(string name) this method can be overloaded in derived class class2:class1

```
{

void getmethod(int i,string name)

{

}

}
```

80. How would you implement inheritance using VB.NET/C#?

When we set out to implement a class using inheritance, we must first start with an existing class from which we will derive our new subclass. This existing class, or base class, may be part of the .NET system class library framework, it may be part of some other application or .NET assembly, or we may create it as part of our existing application. Once we have a base class, we can then implement one or more subclasses based on that base class. Each of our subclasses will automatically have all of the methods, properties, and events of that base class including the implementation behind each method, property, and event. Our subclass can add new methods, properties, and events of its own - extending the original interface with new functionality. Additionally, a subclass can replace the methods and properties of the base class with its own new implementation - effectively overriding the original behavior and replacing it with new behaviors. Essentially inheritance is a way of merging functionality from an existing class into our new subclass. Inheritance also defines rules for how these methods, properties, and events can be merged.

81. . NET is Compile Time OR RunTime Environment?

.Net's framework has CLS,CTS and CLR.CTS checks declartion of types at the time when u write code and CLS defines some rules and restrictions.and CLR comile everything at runtime with following benefits:
Vastly simplified development
Seamless integration of code written in various languages
Evidence-based security with code identity
Assembly-based deployment that eliminates DLL Hell
Side-by-side versioning of reusable components
Code reuse through implementation inheritance
Automatic object lifetime management
Self describing objects

83. What is an interface and what is an abstract class? Please, expand by examples of using both. Explain why.

In a interface class, all methods are abstract without implementation where as in an abstract class some methods we can define concrete. In interface, no accessibility modifiers are allowed. An abstract class may have accessibility modifiers. Interface and abstract class are basically a set of rules which u have to follow in case you are using them (inheriting them).

84. What is an Interface?

An interface is not a class. It is an entity that is defined by the word Interface. An interface has no implementation; it only has the signature or in other words, just the definition of the methods without the body. As one of the similarities to Abstract class, it is a contract that is used to define hierarchies for all subclasses or it defines specific set of methods and their arguments. The main difference between them is that a class can implement more than one interface but can only inherit from one abstract class. Since C# doesn?t support multiple inheritance, interfaces are used to implement multiple inheritance.

86. What is the difference between VB6 and VB.Net?

1,Object-based Language
2,Doesnot support Threading
3,Not powerful Exception handling
mechanism
4,Doesnot having support for the
console based applications
5,Cannot use more than one version
of com objects in vb application
called DLL error
6,Doesnot support for the Disconnected
data source.
VB.Net

____
1,Object-oriented Language
2,supports Threading
3,powerful Exception handling
mechanism
4,having support for the
console based applications
5,More than one version of
dll is supported
6,supports the Disconnected
data source by using Dataset class

87. What is ASP.NET Authentication Providers and IIS Security?

ASP.NET implements authentication using authentication providers, which are code modules that verify credentials and implement other security functionality such as cookie generation. ASP.NET supports the following three authentication providers:

Forms Authentication: Using this provider causes unauthenticated requests to be redirected to a specified HTML form using client side redirection. The user can then supply logon credentials, and post the form back to the server. If the application authenticates the request (using application-specific logic), ASP.NET issues a cookie that

contains the credentials or a key for reacquiring the client identity. Subsequent requests are issued with the cookie in the request headers, which means that subsequent authentications are unnecessary.

Passport Authentication: This is a centralized authentication service provided by Microsoft that offers a single logon facility and membership services for participating sites. ASP.NET, in conjunction with the Microsoft? Passport software development kit (SDK), provides similar functionality as Forms Authentication to Passport users.

Windows Authentication: This provider utilizes the authentication capabilities of IIS. After IIS completes its authentication, ASP.NET uses the authenticated identity's token to authorize access.

To enable a specified authentication provider for an ASP.NET application, you must create an entry in the application's configuration file as follows:

```
// web.config file

<authentication mode = "[Windows/Cookie/Passport/None]">

</authentication>
```

88. How many types of assemblies are there , wat are they?

Private, Public/Shared, Satellite. A private assembly is normally used by a single application, and is stored in the application's directory. A shared assembly is normally stored in the global assembly cache, which is a repository of assemblies maintained by the .NET runtime. Satellite assemblies are often used to deploy language-specific resources for an application. These language-specific assemblies work in side-by-side execution because the application has a separate product ID for each language and installs satellite assemblies in a language-specific subdirectory for each language.

89. What are Attributes?

Attributes are declarative tags in code that insert additional metadata into an assembly. There exist two types of attributes in the .NET Framework: Predefined attributes such as Assembly Version, which already exist and are accessed through the Runtime Classes; and custom attributes, which you write yourself by extending the System. Attribute class.

90. What is a Metadata?

Metadata is information about a PE. In COM, metadata is communicated through non-standardized type libraries. In .NET, this data is contained in the header portion of a COFF-compliant PE and follows certain guidelines; it contains information such as the assembly's name, version, language (spoken, not computer—a.k.a.,

"culture"), what external types are referenced, what internal types are exposed, methods, properties, classes, and much more. The CLR uses metadata for a number of specific purposes. Security is managed through a public key in the PE's header. Information about classes, modules, and so forth allows the CLR

to know in advance what structures are necessary. The class loader component of the CLR uses metadata to locate specific classes within assemblies, either locally or across networks. Just-in-time (JIT) compilers use the metadata to turn IL into executable code. Other programs take advantage of metadata as well. A common example is placing a Microsoft Word document on a Windows 2000 desktop. If the document file has completed comments, author, title, or other Properties metadata, the text is displayed as a tool tip when a user hovers the mouse over the document on the desktop. You can use the Ildasm.exe utility to view the metadata in a PE. Literally, this

tool is an IL disassembler.

91. What is managed code and managed data?

Managed code is code that is written to target the services of the Common Language Runtime. In order to target these services, the code must provide a minimum level of information (metadata) to the runtime. All C#, Visual Basic .NET, and JScript .NET code is managed by default. Visual Studio .NET C++ code is not managed by

default, but the compiler can produce managed code by specifying a command-line switch (/CLR). Closely related to managed code is managed data--data that is allocated and de-allocated by the Common Language Runtime's garbage collector. C#, Visual Basic, and JScript .NET data is managed by default. C# data can, however, be marked as unmanaged through the use of special keywords. Visual Studio .NET C++ data is unmanaged by default (even when using the /CLR switch), but when using Managed Extensions for C++, a class can be marked as managed using the __gc keyword. As

the name suggests, this means that the memory for instances of the class is managed by the garbage collector. In addition, the class becomes a full participating member of the .NET Framework community, with the benefits and restrictions that it brings. An example of a benefit is proper interoperability with classes written in other languages (for example, a managed C++ class can inherit from a Visual Basic class). An example of a restriction is that a managed class can only inherit from one base class.

92. What is "Common Language Runtime" (CLR)?

CLR is .NET equivalent of Java Virtual Machine (JVM). It is the runtime that converts a MSIL code into the host machine language code, which is then executed appropriately. The CLR is the execution engine for .NET Framework applications. It provides a number of services, including: • Code management (loading and execution)

- Application memory isolation

- Verification of type safety

- Conversion of IL to native code.

- Access to metadata (enhanced type information)

- Managing memory for managed objects

- Enforcement of code access security

- Exception handling, including cross-language exceptions

- Interoperation between managed code, COM objects, and pre-existing DLL's (unmanaged code and data)

- Automation of object layout

- Support for developer services (profiling, debugging, and so on).

93. Can you give an example of when it would be appropriate to use a web service as opposed to a non-serviced .NET component?

Webservice:
Appropriate, when exposing the functionalitiss across different Platforms (windows/Linux/Solaris..etc.,) or.NET to J2EE.

.NET Component:

Appropriate, when exposing the functionalities within propritory O/S(.Net-.Net, or Java-Java platform)

94. What does WSDL stand for?

WSDL stands for Web Services Description Language. It is an XML representation of the web service interface.

There are two types of the operations specified in the WSDL file, as represented by the <soap:binding> attribute of the file.

1. Document oriented operations -- are the ones which contain XML documents as input and output

2. Result oriented operations -- are the ones which contain input parameters as the input message and result as the output message

95. How can you provide an alternating color scheme in a Repeater control?

AlternatingItemTemplate Like the ItemTemplate element, but rendered for every otherrow (alternating items) in the Repeater control. You can specify a different appearancefor the AlternatingItemTemplate element by setting its style properties.

96. How would one do a deep copy in .NET?

System.Array.CopyTo() - Deep copies an Array

97. What are the features of ADO.Net ?

ADO.NET features:

1. Disconnected Data Architecture

2. Data cached in Datasets

3. Data transfer

in XML format

4. Interaction with the database is done through data commands

98. What is an interface and what is an abstract class? Please, expand by examples of using both. Explain why?

Abstract classes are closely related to interfaces. They are classes that cannot be instantiated, and are frequently either partially implemented, or not at all implemented. One key difference between abstract classes and interfaces is that a class may implement an unlimited number of interfaces, but may inherit from only one abstract (or any other kind of) class. A class that is derived from an abstract class may still implement interfaces. Abstract classes are useful when creating components because they allow you specify an invariant level of functionality in some methods, but leave the implementation of other methods until a specific implementation of that class is needed. They also version well, because if additional functionality is needed in derived classes, it can be added to the base class without breaking code.

99. What is "Common Language Specification" (CLS)?

What is "Common Language Specification" (CLS)?

100. Where on the Internet would you look for Web services?

UDDI.org-Universal Description, Discovery Integration

If you like this Kit:

This eBook contains 100 Interview Questions & Answers. If you like this eBook, you will definitely love the full version of Interview Success Kit – which contains 9,000 Interview Questions & Answers.

Read more about the Complete Interview Success kit at

http://interview-kit.vyomworld.com/

Interview Success Kit is a complete Interview Cheat-sheet, written by 58 Subject Experts.