

## Difference between ASP.NET 3.5 and ASP.NET 4.0

S.No	ASP.NET 3.5	ASP.NET 4.0
1	<p><b>Whether Client data can be directly accessed ?</b> In ASP.NET 3.5, the data from the client side cannot be directly accessed. The client side data can only be accessed using script manager's Page methods, interface named ICallbackEventHandler or by using the component XMLHttpRequest.</p>	<p><b>Whether Client data can be directly accessed ?</b> In ASP.NET 4, the data from the client side can be directly accessed using client data view and client data context objects.</p> <p>Following methods are available in ASP.NET 4.0 to access Client data directly,</p> <ol style="list-style-type: none"> <li>1) Client data controls</li> <li>2) Client templates</li> <li>3) Client data context</li> </ol>
2	<p><b>Whether we can set MetaTags (Keywords,Description) in the Page Directive?</b> In ASP.NET 3.5, two meta tags can be used, one with name as keywords and other with name as description to record the keywords and description for SEO purpose.</p> <p>Please look at <b>MetaTags for ASP.NET 3.5</b></p>	<p><b>Whether we can set MetaTags (Keywords,Description) in the Page Directive?</b> The meta keywords and meta description is really useful for SEO optimization of the web page. In ASP.NET 4, the keywords and description can be included as part of page directive itself.</p> <p>Please look at <b>MetaTags for ASP.NET 4.0</b></p>
3	<p><b>Whether ViewState can be applied at the Control level ?</b> ASP.NET 3.5 EnableViewState property cannot be used to apply ViewState at the Control level. It is mainly used at the page level. Its default value is True and its acceptable values are True and False.</p>	<p><b>Whether ViewState can be applied at the Control level ?</b> In ASP.NET 4, ViewState mechanism is improved to set ViewState at the Control level besides at the page level set by EnableViewState property in ASP.NET 3.5. Its default value is Inherit and acceptable values are Enabled, Disabled and Inherit.</p>
4	<p><b>How ASP.NET 3.5 identifies ClientID ?</b> In ASP.NET 3.5, ClientId property has to be used</p>	<p><b>How ASP.NET 4.0 identifies ClientID ?</b> In ASP.NET 4, a property called ClientIDMode is newly introduced to identify and record the ClientId easily.</p>

	<p>to find the dynamically generated client id.</p>	<p>ClientIDMode has following values.</p> <p><b>AutoID</b> – Same as ASP.NET 3.5</p> <p>Static – There won't be any separate clientid generated at run time</p> <p><b>Predictable</b>-These are used particularly in datacontrols. Format is like clientIDrowsuffix with the clientid vlaue</p> <p><b>Inherit</b>- This value specifies that a control's ID generation is the same as its parent.</p> <p>The default value of <a href="#">ClientIDMode</a> for a page is <a href="#">Predictable</a>. The default value of <a href="#">ClientIDMode</a> for a control is <a href="#">Inherit</a>. Because the default for controls is <a href="#">Inherit</a>, the default generation mode is <a href="#">Predictable</a>.</p>
5	<p><b>Whether permanent redirection is possible or not ?</b></p> <p>There is no RedirectPermanent() method available in ASP.NET 3.5. Redirect method is less useful than the RedirectPermanent method. It will cause search engine results to be less current, and this can also impact performance because visitors will not be accessing the best URL. Redirect may be most useful for login pages or more complex situations.</p>	<p><b>Whether permanent redirection is possible or not ?</b></p> <p>ASP.Net 4.0 introduced a new URL redirection method RedirectPermanent() which avoids round trips.</p> <p>We can implement this as shown below:</p> <p><b>RedirectPermanent("/newpath/newpage.aspx");</b></p> <p>RedirectPermanent returns a 301 HTTP response—it redirects permanently to another location. Search engines such as Google and Bing will change their indexes to point to the new page directly. To call RedirectPermanent you will need to get the Response object from the HttpContext.</p> <p><b>Then</b>, you can call RedirectPermanent: if you pass false as the second parameter, you can perform further actions and avoid an exception.</p>
6	<p><b>Output Caching Enhancement:</b></p> <p>OutPut Cache in ASP.Net 3.5 has a limitation - generated content always has to be stored in memory, and on servers that are experiencing heavy traffic, the memory consumed by output caching can compete with</p>	<p><b>Output Caching Enhancement:</b></p> <p>ASP.NET 4 adds an extensibility point to output caching that enables you to configure one or more custom output-cache providers. Output-cache providers can use any storage mechanism to persist HTML content. This makes it possible to create custom output-cache providers for diverse persistence mechanisms, which can include local or remote disks, cloud storage, and distributed cache engines.</p>

	memory demands from other portions of a Web application.	In order to know how to implement Custom Output Caching, please refer to the following URL,  <a href="http://www.technologymaterial.com/QAndADetails.aspx?qid=3&amp;type=aspnet4">http://www.technologymaterial.com/QAndADetails.aspx?qid=3&amp;type=aspnet4</a>
7	<p><b>QueryExtender Control for filtering the data returned by EntityDataSource and LinqDataSource controls:</b></p> <p>There is no QueryExtender control available in ASP.NET 3.5 to filter the data returned by DataSource Controls( EntityDataSource and LinqDataSource).It has to explicitly use 'Where clause' in the data source.</p>	<p><b>QueryExtender Control for filtering the data returned by EntityDataSource and LinqDataSource controls:</b></p> <p>QueryExtender Control is an add-on to the DataSource Controls: EntityDataSource and LinqDataSource. QueryExtender is used to filter the data returned by these controls. As the QueryExtender control relies on LINQ, the filter is applied on the database server before the data is sent to the page, which results in very efficient operations.</p> <p>i.e., QueryExtender Control is intended to be used to create filters for data that is retrieved from a data source, without using an explicit Where clause in the data source. The control can be used to filter data in the markup of a Web page by using declarative syntax.</p> <p>Please look at <b>QueryExtender Control in ASP.NET 4.0</b></p>

### MetaTags for ASP.NET 3.5:

```
<meta name="keywords" content="These, are, my, keywords" />
<meta name="description" content="This is the description of my page" />
```

### MetaTags for ASP.NET 4.0:

```
< %@ Page Language="C#" CodeFile="Default.aspx.cs"
Inherits="_Default"
Keywords="Keyword1,Key2,Key3,etc"
Description="description" %>
```

### QueryExtender Control in ASP.NET 4.0:

```
<asp:LinqDataSource ID="dataSource" runat="server"> TableName="Products">

</asp:LinqDataSource> <asp:QueryExtender TargetControlID="dataSource"
runat="server"> <asp:SearchExpression DataFields="ProductName,
Supplier.CompanyName" SearchType="StartsWith"> <asp:ControlParameter
ControlID="TextBoxSearch" /> </asp:SearchExpression> </asp:QueryExtender>
```

**References:**

<http://www.rapidprogramming.com/questions-answers/What-is-the-difference-between-ASP-NET-4-and-ASP-NET-3-5--720>

<http://rahulgbhatia.wordpress.com/2011/11/04/main-differences-between-asp-net-3-5-and-asp-net-4-0/>

<http://onlydifferencefaq.blogspot.in/2012/08/difference-between-enableviewstate-and.html>

<http://msdn.microsoft.com/en-us/library/system.web.ui.control.clientid.aspx>

<http://www.technologymaterial.com/QAndA2.aspx?type=aspnet4>

<http://www.dotnetperls.com/redirect>

<http://www.codeproject.com/Articles/79742/How-to-Using-the-ASP-NET-4-QueryExtender-Control>

<http://www.technologymaterial.com/QAndADetails.aspx?qid=8&type=aspnet4>

**And, further updates on difference between questions and answers, please visit my blog @ <http://onlydifferencefaq.blogspot.in/>**