

Re: asp.net and XHTML

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-05/msg05069.html>

- *From:* "Alex D." <alexware69@xxxxxxxxxxxxx>
 - *Date:* Thu, 26 May 2005 11:18:32 -0400
-

this is the difference:

Some of the XHTML rules that differ from HTML include the following:

- All elements either include an explicit closing tag or are self-closing (with />).
- Tag and attribute names are rendered in lowercase, and attribute values are included in double quotation marks. For example, if you use a System.Web.UI.WebControls.GridView control on your page, when the page is rendered, the System.Web.UI.WebControls.GridView control emits HTML that is XHTML compliant. All generated elements use explicit opening and closing tags (or self-closing tags), and attribute values are included in double quotation marks.
- Formatting information is rendered using only cascading style sheet styler. To support this standard, ASP.NET controls do not render tags or attributes such as bgcolor.
- Controls do not render custom attributes (expando attributes) that are not defined in the XHTML standards.
- In ASP.NET, if controls generate IDs, as occurs in the Repeater, GridView, and other controls, the format of the IDs is XHTML 1.1 compliant.
- ASP.NET dynamically adds an action attribute to the form tag. Because XHTML 1.1 prohibits it, form tags do not include a name attribute.
- Because XHTML requires all elements to be enclosed in a container element, ASP.NET controls such as input elements are rendered in div elements. This includes the HTML markup rendered for controls such as the TextBox, CheckBox, and RadioButton controls. It also includes hidden fields, such as the element used to store view-state data.
- ASP.NET codes characters such as & (for example, as &). This includes URLs that are generated to reference JavaScript script and the contents of encoded values such as view state.
- Any script tags rendered into the page include an appropriate type

Re: asp.net and XHTML

attribute. (This pertains to scripts that are created by the page or controls or by the RegisterHiddenField, RegisterStartupScript, and RegisterClientScriptBlock methods; script blocks that you create are not automatically amended with a type attribute.) In ASP.NET any script tags rendered by the page use the appropriate type attribute (type="type/javascript") and do not include a language attribute. ASP.NET renders script tags when the page includes certain controls (such as the System.Web.UI.WebControls.HyperLink , System.Web.UI.WebControls.LinkButton , and System.Web.UI.WebControls.Calendar controls) that require client script to perform a postback. Script is also rendered for controls that rely on client-side functionality, such as the System.Web.UI.WebControls.TreeView and validator controls.

j.. IF ASP.NET renders script blocks, the contents of the script blocks is rendered in CDATA elements (inside and]> tags) to prevent special characters such as & and < from being interpreted as XML.</p></div><div data-bbox="115 390 587 423" data-label="Text"><p>"Patrice" <nobody@xxxxxxxxxxxx> wrote in message
news:u9FC%23MgYFHA.3488@xxxxxxxxxxxxxxxxxxxxxxxxxxxx</p></div><div data-bbox="115 424 329 439" data-label="Text"><p>> Yes this is what I meant...</p></div><div data-bbox="116 443 134 456" data-label="Text"><p>></p></div><div data-bbox="115 457 716 490" data-label="Text"><p>> I don't see what difference you ake between "XHTML" and "HTML" encoded
> scripts...</p></div><div data-bbox="116 493 134 506" data-label="Text"><p>></p></div><div data-bbox="115 507 609 523" data-label="Text"><p>> For example the problme seems to be related to & being &amp;;</p></div><div data-bbox="116 526 134 539" data-label="Text"><p>></p></div><div data-bbox="115 540 292 557" data-label="Text"><p>> What if you try just :</p></div><div data-bbox="116 560 134 573" data-label="Text"><p>></p></div><div data-bbox="115 573 567 590" data-label="Text"><p>> TextBox1.Attributes.Add("onkeydown","alert(8 && 4);")</p></div><div data-bbox="116 593 134 606" data-label="Text"><p>></p></div><div data-bbox="115 606 322 622" data-label="Text"><p>> Even if it is rendered as :</p></div><div data-bbox="115 623 629 640" data-label="Text"><p>> onkeydown="alert(8 & & 4);" does it work as expected ?</p></div><div data-bbox="116 643 134 656" data-label="Text"><p>></p></div><div data-bbox="115 656 189 672" data-label="Text"><p>> Patrice</p></div><div data-bbox="116 675 134 688" data-label="Text"><p>></p></div><div data-bbox="116 691 134 704" data-label="Text"><p>></p></div><div data-bbox="115 705 159 721" data-label="Text"><p>> --</p></div><div data-bbox="116 724 134 737" data-label="Text"><p>></p></div><div data-bbox="115 738 637 755" data-label="Text"><p>> "Alex D." <alexware69@xxxxxxxxxxxx> a écrit dans le message de</p></div><div data-bbox="115 756 572 772" data-label="Text"><p>> news:OBsL35fYFHA.3380@xxxxxxxxxxxxxxxxxxxxxxxxxxxx</p></div><div data-bbox="115 772 702 789" data-label="Text"><p>>> HTML encoded javascript woks fine here too.... what doesn't work fine is</p></div><div data-bbox="115 789 375 806" data-label="Text"><p>>> XHTML encoded javascript....</p></div><div data-bbox="115 806 686 823" data-label="Text"><p>>> If I use Me.ClientScript.RegisterClientScriptBlock then I guess I need to</p></div><div data-bbox="115 822 567 839" data-label="Text"><p>>> create a whole function() and then call the function from</p></div><div data-bbox="115 839 251 855" data-label="Text"><p>> attributes.add...</p></div><div data-bbox="115 855 241 872" data-label="Text"><p>>> is that right?</p></div><div data-bbox="116 875 144 888" data-label="Text"><p>>></p></div><div data-bbox="116 891 144 904" data-label="Text"><p>>></p></div><div data-bbox="115 904 547 922" data-label="Text"><p>>> "Patrice" <nobody@xxxxxxxxxxxx> wrote in message</p></div><div data-bbox="115 937 320 956" data-label="Page-Footer"><p>Re: asp.net and XHTML</p></div><div data-bbox="925 937 950 955" data-label="Page-Footer"><p>2</p></div>

Re: asp.net and XHTML

>> news:%237FOIyfYFHA.2076@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>> > What I meant was to use Me.ClientScript.RegisterClientScriptBlock to
>> > render
>> > the script and then use attributes.add just to call it... AFAIK scripts
>> > registered his way are not encoded (the trick is that this function
> knows
>> > this a script while attributes.add doesn't know what you are doing).
>> >
>> > What browser are you using ? Here HTML encoded JavaScript code seems to
>> > work
>> > fine...
>> >
>> > Hope I'm clearer...
>> >
>> > Patrice
>> >
>> > --
>> >
>> > "Alex D." <alexware69@xxxxxxxxxxxx> a écrit dans le message de
>> > news:OOMB4rfYFHA.3712@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>> >> my friend, I get the same problem even if I insert the code here:
>> >>
>> >> <FORM id=Form1 method=post runat="server" onload=
>> >> "if((event.which && event.which== 13)||((event.keyCode && event.keyCode
> ==
>> >> 13)){ document.Form1.cmdLogin.click();return false; }else return
> true;");>
>> >>
>> >> but if I insert the code BEFORE the FORM tag , let say, inside the
> BODY,
>> >> then it render & instead of & which is OK. SO it is an XHTML
> related
>> >> problem
>> >>
>> >> If you are interested you can read this, I tried the suggested
>> >> solution
>
>> >> to
>> >> disable XHTML but it didn't work.
>> >>
>> >> ASP.NET and XHTML Compliance
>> >> ASP.NET allows you to create Web pages that are compliant with XHTML
> 1.1
>> >> standards. XHTML is a W3C standard that defines HTML as an XML
> document.
>> >> Creating Web pages that are conformant with XHTML standards guarantees
>> > that
>> >> the elements in the pages are well formed. Because many browsers are
>> > moving
>> >> toward supporting XHTML, creating pages that conform to XHTML
>> >> standards

Re: asp.net and XHTML

>>> helps ensure that your pages will be compatible with browsers in the
>>> future.
>>> XHTML is also extensible, allowing the definition of new elements.
>>> Finally,
>>> an XHTML page is much easier to read programmatically for situations
>>> in
>>> which the Web page is processed by a computer rather than read by
> users,
>>> and
>>> the document can be manipulated using transformations. For more
>>> information
>>> about the XHTML standard, see the W3C site at www.w3.org/TR/xhtml1.
>>>
>>> Compliance Features
>>> XHTML defines elements and attributes more strictly than HTML. By
>>> default,
>>> all markup produced by ASP.NET and Web server controls included with
>>> ASP.NET
>>> now conforms to this standard.
>>>
>>> Some of the XHTML rules that differ from HTML include the following:
>>>
>>> a.. All elements either include an explicit closing tag or are
>>> self-closing (with />).
>>>
>>> b.. Tag and attribute names are rendered in lowercase, and attribute
>>> values are included in double quotation marks. For example, if you use
> a
>>> System.Web.UI.WebControls.GridView control on your page, when the page
> is
>>> rendered, the System.Web.UI.WebControls.GridView control emits HTML
> that
>>> is
>>> XHTML compliant. All generated elements use explicit opening and
> closing
>>> tags (or self-closing tags), and attribute values are included in
> double
>>> quotation marks.
>>>
>>> c.. Formatting information is rendered using only cascading style
> sheet
>>> styler. To support this standard, ASP.NET controls do not render
>>>
>>> tags
>>> or attributes such as bgcolor.
>>>
>>> d.. Controls do not render custom attributes (expando attributes)
> that
>>> are
>>> not defined in the XHTML standards.
>>>

Re: asp.net and XHTML

>>> e.. In ASP.NET, if controls generate IDs, as occurs in the Repeater, >>> GridView, and other controls, the format of the IDs is XHTML 1.1 >>> compliant.

>>> f.. ASP.NET dynamically adds an action attribute to the form tag. >>> Because >>> XHTML 1.1 prohibits it, form tags do not include a name attribute.

>>> g.. Because XHTML requires all elements to be enclosed in a >>> container >>> element, ASP.NET controls such as input elements are rendered in div >>> elements. This includes the HTML markup rendered for controls such as > the >>> TextBox, CheckBox, and RadioButton controls. It also includes hidden >>> fields, >>> such as the element used to store view-state data.

>>> h.. ASP.NET codes characters such as & (for example, as &). This >>> includes URLs that are generated to reference JavaScript script and >>> the >>> contents of encoded values such as view state.

>>> i.. Any script tags rendered into the page include an appropriate > type >>> attribute. (This pertains to scripts that are created by the page or >>> controls or by the RegisterHiddenField, RegisterStartupScript, and >>> RegisterClientScriptBlock methods; script blocks that you create are > not >>> automatically amended with a type attribute.) In ASP.NET any script > tags >>> rendered by the page use the appropriate type attribute >>> (type="type/javascript") and do not include a language attribute.

> ASP.NET >>> renders script tags when the page includes certain controls (such as > the >>> System.Web.UI.WebControls.HyperLink , >>> System.Web.UI.WebControls.LinkButton >>> , >>> and System.Web.UI.WebControls.Calendar controls) that require client >>> script >>> to perform a postback. Script is also rendered for controls that rely > on >>> client-side functionality, such as the > System.Web.UI.WebControls.TreeView >>> and validator controls.

>>> j.. IF ASP.NET renders script blocks, the contents of the script > blocks >>> is >>> rendered in CDATA elements (inside <![CDATA[and]]> tags) to prevent >>> special characters such as & and < from being interpreted as XML.

Re: asp.net and XHTML

>>>>
>>>> Controlling XHTML Rendering of ASP.NET Page and Controls
>>>> Under some circumstances, you might not want ASP.NET controls to
>>>> render
>>>> XHTML 1.1 markup. For example, XHTML 1.1 standards prohibit the use of
> a
>>>> name attribute in an HTML form tag, so ASP.NET does not render that
>>>> attribute. However, in your application, you might have existing pages
>>>> with
>>>> client script that relies on the form tag's name attribute.
>>>>
>>>> You can configure ASP.NET so that it does not render XHTML 1.1 markup
> to
>>>> allow pages created in earlier versions of ASP.NET to work as
>>>> designed.
>>>> For
>>>> more information, see How to: Configure ASP.NET for Non-XHTML
> Rendering.
>>>>
>>>>
>>>>
>>>> "Patrice" <nobody@xxxxxxxxxxxx> wrote in message
>>>> news:e56AGhfYFHA.2124@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>>>>> As noted by Joerg, this is HTML not XHTML...
>>>>>
>>>>> If it really causes a problem you could register the client script
>>>>> using
>>>>> the
>>>>> "ClientScript" member of the page class and just make the call using
>>>>> Attributes.Add. Here you render the script as an HTML attribute
> causing
>>>>> the
>>>>> string to be HTML encoded (likely ASP.NET doesn't check anything,
> this
>>>>> is
>>>>> an
>>>>> HTML attribute then it encodes it)...
>>>>>
>>>>> Let us know about your findings (especially if using HTML encoded
>>>>> scripts
>>>>> causes a problem on some browsers)...
>>>>>
>>>>> Patrice
>>>>>
>>>>> --
>>>>>
>>>>> "Alex D." <alexware69@xxxxxxxxxxxx> a écrit dans le message de
>>>>> news:OJnkyJfYFHA.3712@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
>>>>>> like this:
>>>>>>
>>>>>> this.txtUserName.Attributes.Add("onkeydown","if((event.which &&

>>>
>>>
>>
>>
>
>

• **References:**

- ◆ **[asp.net and XHTML](#)**
 ◇ *From: Alex D.*
 - ◆ **[Re: asp.net and XHTML](#)**
 ◇ *From: Patrice*
 - ◆ **[Re: asp.net and XHTML](#)**
 ◇ *From: Alex D.*
 - ◆ **[Re: asp.net and XHTML](#)**
 ◇ *From: Patrice*
 - ◆ **[Re: asp.net and XHTML](#)**
 ◇ *From: Alex D.*
 - ◆ **[Re: asp.net and XHTML](#)**
 ◇ *From: Patrice*
 - ◆ **[Re: asp.net and XHTML](#)**
 ◇ *From: Alex D.*
 - ◆ **[Re: asp.net and XHTML](#)**
 ◇ *From: Patrice*
- Prev by Date: **[Re: Browser problem or dotnet bug?](#)**
 - Next by Date: **[Reading UTF-8 Data from XML file](#)**
 - Previous by thread: **[Re: asp.net and XHTML](#)**
 - Next by thread: **[Re: asp.net and XHTML](#)**
 - Index(es):
 - ◆ **[Date](#)**
 - ◆ **[Thread](#)**