

HtmlTableCellNonBlanking

Automatically Inserting Non-Breaking Spaces Into Empty Table Cells

Many browsers suffer from the *feature* that they omit the table borders from empty cells which. You have different possibilities to solve this issue.

CSS property `empty-cells`:

```
table {  
    empty-cells: show;  
}
```

The default value is `collapse`, so empty cells are not shown. More info you can find in the [CSS 2 spec](#).

Non-breaking space via XSLT:

The following XSLT snippet, a `xsl:template` match, activates on encountering a table cell and replaces empty cells and cells that contain only whitespace with a non-breaking space.

To use this code, simply copy the snippet into one of your XSLT stylesheets.

```
<xsl:template match="td">  
    <td>  
        <xsl:choose>  
            <!-- If cell has child nodes or contains text then output them -->  
            <xsl:when test="* or normalize-space()">  
                <xsl:apply-templates/>  
            </xsl:when>  
            <!-- Otherwise the cell is empty and we output a non-breaking space -->  
            <xsl:otherwise>  
                <xsl:text>&#160;</xsl:text>  
            </xsl:otherwise>  
        </xsl:choose>  
    </td>  
</xsl:template>
```

Notes

- The expression in the `test` attribute is evaluated to a boolean value.
- `*` is evaluated to true if a child element exists.
- `normalize-space()` is evaluated to true if the function returns a string. All descendant text nodes of the context `td` element are concatenated, leading and trailing whitespace characters are removed. Multiple spaces, tabs and new line characters are replaced by a single space. More info in the [XPath spec](#).
- ` ` is the character reference for the non-breaking space.

– based on a solution provided by *AlanHodgkinson*.

Further comments

If ` ` is not the solution you prefer, but you want to use the ` ` known from HTML, you can add an entity to the XSLT stylesheet in front of the `xsl:stylesheet` root element:

```
<!DOCTYPE xsl:stylesheet [  
    <!ENTITY nbsp "&#160;">  

```

More info in the [XML spec](#).

– based on a solution provided by *Anonymous author*.

The originally posted solution by Alan Hodgkinson uses `disable-output-escaping`, but this is not recommended:

```
<xsl:text disable-output-escaping="yes">&amp;amp;nbsp;</xsl:text>
```

Why? Please read on this issue at [XSLTDisableOutputEscaping](#).

Or use a comment

I believe that a comment is just as good as a space as far as Netscape is concerned. It's a bit bigger, but it has the advantage that it's not really "content", so it shouldn't affect formatting as much as a space, which a browser will be rendering in a particular font, hence it will have some "size".

e.g. use this template in conjunction with an identity transformation:

```
<xsl:template match="td[not(normalize-space())]">
  <xsl:copy>
    <xsl:copy-of select="@*" />
    <xsl:comment>NS hack</xsl:comment>
    <xsl:apply-templates/>
  </xsl:copy>
</xsl:template>
```

– [Con](#).