Using Vertical Layout in Internet Explorer 5.5

Mark Grinols
Microsoft Corporation

October 2000

Summary: One of the principal new features of Microsoft Internet Explorer 5.5 is vertical layout. This article explores the basic concepts of vertical layout and helps you avoid some common mistakes. (4 printed pages)

> **Note** The samples in this article require Internet Explorer 5.5 or later and a Japanese OS or Language Pack to display the Japanese characters.

Contents

What is Vertical Layout?

When you look at almost any Web page—this one, for example—the flow of text in a paragraph begins on the left and moves toward the right. When text wraps, the new line begins below the previous one at the left side of the page. It's so natural that you don't even notice it.

For much of the world, this isn't natural at all. If you've ever picked up a Japanese or Chinese book or magazine, you know that it is common to layout printed text vertically. The text begins at the top-right corner of the page and flows downward, wrapping the next line to the top of the page and to the left of the previous line. Newspaper headlines, for example, run down the right side of the page. Street signs, love letters, and most other forms of written text are laid out this way as well. The exceptions are Web pages. HTML can't handle this type of layout. However, this hasn't stopped Web authors from trying everything from bitmaps to very narrow table cells, but there hasn't been a satisfactory way to achieve vertical layout until now.

The Writing-mode Attribute

In Microsoft® Internet Explorer 5.5, vertical layout support is enabled with a CSS attribute called writing-mode (currently proposed for CSS 3). The following examples compare horizontal to vertical layout with both English and Japanese content. Look through the examples and then we'll go through the details.

The **writing-mode** attribute has two values to examine:

- tb-rl
- lr-tb

Tb-rl stands for *top-to-bottom, right-to-left*, which is a description of how the content will flow in the element to which this style is applied. Lr-tb stands for *left-to-right, top-to-bottom* and is the default flow for all text. This is the typical horizontal flow to which readers of English are accustomed. As with any other CSS attribute, these values can be specified inline, with an external CSS file, or through the object model (style.writingMode).

Note that although East Asian languages inspired vertical layout support, text of any language can be given a vertical text flow. Within this flow, the orientation of individual characters depends on the language or script to which they belong. In keeping with the tradition of vertical typography, characters from East Asian scripts are

displayed with an upright orientation. All other characters, including Latin, are rotated 90Ã‚Â° clockwise and appear on their sides when in vertical layout.

Applicable Elements

A list of elements to which **writing-mode** applies can be found in the MSDN Library documentation. However, you won't find the BODY element on the list. This means that the **BODY** element always has horizontal layout, though elements that it contains can have vertical layout. Sometimes it's difficult to anticipate how this combination of layout rules will look, as the following example demonstrates.

```
<HTML>
<BODY>
<DIV style="writing-mode:tb-rl; width:50cm">
This is a sentence in a DIV element with vertical layout.
</DIV>
</BODY>
<HTML>
```

View the sample.

When you view this markup in Internet Explorer 5.5, you don't see the text at first, but a horizontal scroll bar does appear. You can see the text if you scroll the page all the way to the right. Why does this happen?

The **BODY** element always has horizontal layout and the browser always aligns the viewport to start at the origin of the top-most element in the markup. In the preceding example, this origin is the upper-left corner of the page. In order for the page to load with your text showing, you must cause the body to scroll to the right. This is achieved by making the direction property on the body RTL (right-to-left), and then setting the direction back to LTR (left-to-right) in a global DIV element. This puts the origin point in the upper right corner.

```
<HTML>
<BODY style="direction:rtl">
<DIV style="direction:ltr; writing-mode:tb-rl; width:50cm">
This is a vertical sentence.
</DIV>
</BODY>
</HTML>
```

View the sample.

Interaction with Other CSS Attributes

Many CSS attributes have possible values related to direction or position, such as top, bottom, left, and right. How should these values be interpreted in a vertical layout context? Generally, CSS properties that affect direction or position are interpreted absolutely. That is physically, meaning that the implicit or explicit directionality of the style doesn't change with vertical layout. For example, consider the border-left attribute. For example, in a vertical layout, **border-left** means that a border is drawn on the left edge of the element. Because it is absolutely positioned, **border-left** does not become border-top in vertical layout.

Some CSS attributes, however, are interpreted relatively. That is logically, meaning it makes no sense to interpret them absolutely as above. An example is the line-height attribute. This attribute controls a line's height if the line is horizontal, or the line's width if the line is in vertical layout. In other words, the **line-height** attribute controls the size of the line in the dimension perpendicular to the baseline. For this reason, it doesn't make sense to interpret the property absolutely. The World Wide Web Consortium (W3C) provides a list of CSS properties that are interpreted logically.

Containership and Inheritance

It's important to remember that the **writing-mode** attribute determines how the content of the element is laid out, not the position of the element itself. When an element has the tb-rl value for **writing-mode**, the entire element's content has vertical layout (ignoring, for the moment, limitations on inheritance or the case of a child element that sets horizontal layout). This means that text and child elements flow top-to-bottom, right-to-left.

View the sample.

Inheritance of the **writing-mode** attribute is not uniform. Some elements do not inherit it at all. Other elements pass the **writing-mode** attribute on to their children but don't use it themselves. This is documented on the writing-mode reference page. Move the mouse pointer over any element in the "Applies To" list to see if inheritance limitations apply to that element. For an example of limited inheritance, see the MSDN Library reference for the BUTTON element.

Element Height

In horizontal layout, if no particular sizing has been specified, a block element usually has the same width as its parent element, minus margins, borders, and padding. The element's height will be adjusted based on the volume of its content (the more wrapped lines of text, the taller the element becomes). A vertical element with a parent that has vertical layout works similarly. A block element has the same height as its parent element, minus margins, borders, and padding.

Element height gets tricky when you consider the behavior of a vertical element with a horizontal parent (and remember that because the **BODY** element is always horizontal, this is a common situation). The height of the horizontal parent is variable—at times very small and at times very large, based on the parent's content. It's not a good solution to determine the child element's height from the parent's height. Instead, a calculation is performed to determine a best-fit height. This calculation is based on a number of variables, including the size of the characters (based on the first one) and the longest word in the text run. The minimum height of the vertical child element is usually about 10 characters tall.

View the English sample.
View the Japanese sample.

Of course, you can avoid most of this complexity and get better control of your layout by specifying heights, widths, and positions in your markup.

Platform Limitations

Operating system limitations prevent the correct display of East Asian characters in an upright orientation in some configurations. This limitation has been eliminated in Microsoft Windows® 2000, but Microsoft Windows NT® 4.0 and Windows 9*x* systems require a version of the operating system that has been localized for an East Asian language to ensure that these characters display correctly.

Summary

The implementation of vertical layout in Internet Explorer 5.5 is an important milestone on the road to making the World Wide Web more useful to people worldwide. Vertical layout makes possible a whole new kind of expression for Web designers of East Asian content, whether that means better emulation of traditional publishing formats or inventing a completely new look. As with anything else, the best way to learn about vertical layout is to experiment. Once you get the hang of it, get out there and put it to work.