Zoom Version 2:
Windows® Internet Explorer® 8
Beta 1 for Developers

Making the Web Work for You

March 2008

For more information, press only:
Rapid Response Team
Waggener Edstrom Worldwide
(503) 443-7070
rrt@waggeneredstrom.com

OVERVIEW

Page zoom lets you enlarge or reduce the view of a Web page to improve readability. The feature is particularly useful on really large and really small displays, and allows for scaling of content while maintaining the intended layout of the page. The first iteration of the zoom feature shipped in Windows Internet Explorer 7; this second iteration focuses on improving the existing experience by providing a zoom experience that is higher quality, more predictable, and more persistent. Primary features in this release include the elimination of horizontal scroll bars for most mainstream scenarios and the introduction of persistent zoom states.

COMPATIBILITY: Changes in Behavior from Internet Explorer 7

1. The page zoom feature in Internet Explorer 7 "magnifies" the original Web page. Page elements are scaled after layout and then displayed on the screen. The improved page zoom feature in this release performs layout after scaling. This means that the page is no longer optically zoomed: instead of scaling linearly, each page element will scale dependent on how its dimensions were defined. For example, consider a box with width defined as 50% of body, which computes to 100px at 100%. When zoomed to 200% in Internet Explorer 7, the width of this box would grow to 200px because a re-layout was not performed. With adaptive page zoom in Internet Explorer 8 Beta 1 for Developers, the width would still be 100px at 200% because the width of the containing block (body) has not changed. This is explained further in the section "Feature Details."

2. All DHTML properties are now assumed to be logical. In the previous version some properties (such as mouse coordinates) were treated as physical while others were logical (offset). This essentially required Web developers to be aware of or calculate the zoom state.
   In this release all properties are logical.

FEATURE DETAILS

Zoom version 2 works by adjusting the internal state of the dots-per-inch calculations within Internet Explorer 8 Beta 1 for Developers. To understand this better, consider how different display settings affect a page's layout.

Screen Resolution: Assume an initial screen resolution of 1200 × 1600. Changing this to 800 × 600 means changing the spacing between pixels; in other words, pixels are effectively bigger. An element of 100px × 100px is now twice as big on the screen.

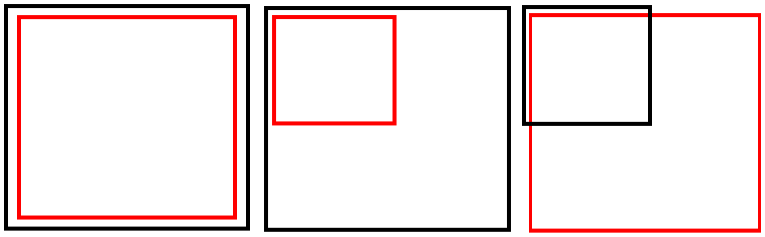
System DPI: Assume an initial value of 96 DPI. Changing this to 192 DPI, but keeping all other factors the same, implies that there are more pixels in an inch. The spacing between pixels has not changed; however, more pixels are required to form a logical inch.

Zoom Factor: Changing zoom factor to 2 means that everything should be twice as big.

HTML Layout Measurements: This value is always 96 DPI; this section of the CSS 2.1 spec provides details on why.

The Internet Explorer page zoom feature cannot change any of the above settings except for the zoom factor. However, it is possible to layout or render a page at a DPI setting different from the configured system DPI, thus achieving zoom results.

Examples:

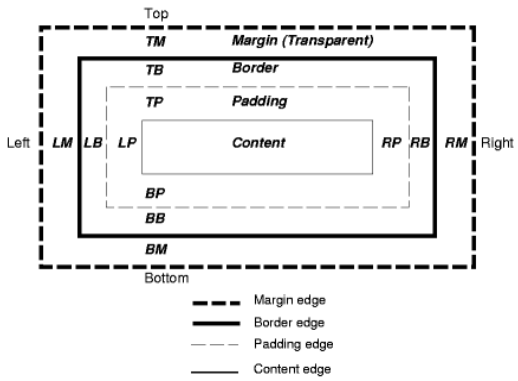


Zoom factor = SystemDPI/LayoutDPI

Effect of Zoom on the CSS Box Model

The CSS Box Model defines the "space" occupied by an element. An element is defined by its width, height, padding, border widths, and margins.

The following diagram illustrates this:



When zooming in on a box model, the behavior will depend on the widths and heights of the content, and how the padding, border and margins have been defined.

- If the width/height of the content area, or the padding/border/margin, are defined as a percentage, then the percentage value is not scaled. Consider this Example:

```
#someDiv {
      width: 50%;
}
```

When zoom changes to (say) 150%, the width will not change to 75%. It will remain 50%.
Since percentage values depend on the containing block's properties, the width and height of the content area will grow or shrink only if the containing block changes size.
A body-level element with its width/height defined in percentage units will not change size, unless viewport size changes.

- If the width/height is set to auto, then the box will grow as necessary.
  NOTE: if the content-area dimensions are set to auto, but the padding/border/margin is specified, then the padding/border/margin area will grow and the content area will shrink as the zooming factor increases. At some point the content area will be reduced to nothing.

- If the width/height/padding/border/margin have been defined in ex or em, they rely on font size property of that element. As fonts do not scale linearly, dimensions defined using em/ex will not scale linearly.
  There are situations where font scaling may not match zoom exactly. For example, if font size is 12 pt and zoom is 110%, the new font size will be 13.2, which is an invalid font size. (Rounding will occur to arrive at a valid size). Therefore, em/ex scaling are affected as well.

NOTE: Since widths can become fractional, the border width has a floor value of 1px.

Effect of Zoom on Common Elements

The following chart lists common elements found in a page's source and the effect zoom has on size and positioning.

Element Name	Result
Floats	Floats will be positioned with respect to their container as per the normal rules of CSS. If the container width changes with zoom, the position of the float will change. The position of the float with respect to the container should not change. If the container is anchored to the BODY element, then its position will not change. If there are two more floats with widths that expand with zoom, and they are positioned next to each other at body level then: 1. A horizontal scroll bar will appear when the first defined float width exceeds viewport width. 2. Floats lower in the source order will move below as the combined width of all floats exceeds the viewport width. The same will happen when floats are defined in a container having width that does not change with zoom.
Relatively Positioned Elements	The position of a relatively positioned element will change depending on how it is defined and what other element it is anchored to. The positional units will increase/decrease with zoom if • Defined in px, ex, em, in, cm, mm, pt, or pc. • Defined in % but the dimensions of the container change.

	This means that if an element is positioned relative to the body using % units, the position will not change. If the combined value of the left position plus the fixed width of a relatively positioned element exceeds the viewport width, a horizontal scroll bar will appear.
Absolutely Positioned Elements	The positional behavior of an absolutely positioned element is identical to that of a relatively positioned element.
Fixed Positioned Elements	A fixed-position element anchored to the viewport will be zoomed even though the viewport is not zoomed. Therefore, if an image is anchored to the bottom right corner, it will increase in size and remain anchored to the bottom right corner of the viewport.
Fonts	Font size is scaled to match the zoom factor; glyphs are not optically magnified. For example, the specified or calculated font size might be 12 pts at 100%.At 200% the font-size will be 24 pts. The problem this introduces is that fractional font sizes are possible. For example, if the initial font size is 12 pts, a zoom level of 110% would require a font-size of 13.2 pts, which is not available. The size is rounded to the closest available font. Important notes: • Width/height dimensions dependent on font size will not scale linearly. • Some Web sites may appear not to scale if the zoom-level change is too small for the font size to jump from one level to another. For example, if the text size on a site is 8 pt, changing the zoom level from 100% to 105% will not have any noticeable effect on text as the font will be scaled from 8 pt to 8.4 pt. (An invalid size, 8.4 pt will round down to 8 pt). • A percentage width box (relative to the body so width doesn't change)that contains justified text might display some strange behavior. • Bitmap fonts are designed for a specific DPI setting and in limited font sizes. They look best at their native pixel size and do not scale well.
Tables	Table scaling is similar to what happens with the basic box model. The main difference: once the total width of a table exceeds the width of a viewport, a horizontal scrollbar will appear. Table cells cannot be flowed differently, e.g. the way floated elements can be moved down.
Overflow	It is important to note that elements with overflow set to hidden could experience problems when the page is zoomed. As mentioned earlier, it is possible that box dimensions do not scale with zoom (e.g., set to percentage of body, or constrained by float next to it). However content will most likely continue to scale, and if overflow is set to hidden then it will be clipped.
Images	Images are scaled using a complex algorithm intended to maintain image

	clarity and fidelity.
IFrames	Zoom affects inline frames the same way it affects the basic box model. The contents of an IFrame will be scaled with the same zoom factor.
Intrinsic Controls	Intrinsic controls will be zoomed in the same way as content.
Fly-out Menus	The behavior of fly-out menus will depend on design and will thus vary from site to site.

Zoom APIs

OLECMDID_ZOOM – This corresponds to the **View** menu's **Text Size** command. Only the text is scaled by using this command. Sites that have fixed text size are not changed in anyway.

This command is used for primarily three purposes: querying the current zoom value, displaying the zoom dialog box, and setting a zoom value.

OLECMDID_GETZOOMRANGE – This command returns two values (**LOWORD**, **HIWORD**) that represent the minimum and maximum of the zoom value range.

This command is used when the user activates zoom UI.

OLECMDID_OPTICAL_ZOOM – Internet Explorer 7 and later. Sets the zoom factor of the browser. Takes a VT_I4 parameter in the range of 10 to 1000 (percent).

OLECMDID_OPTICAL_GETZOOMRANGE – Internet Explorer 7 and later. Retrieves the minimum and maximum browser zoom factor limits. Returns a VT_I4 parameter, where the **LOWORD** is the minimum zoom factor, and the **HIWORD** is the maximum.

Examples

The following are some examples of how Internet Explorer 8 Beta 1 for Developers adaptive page zoom affects layout as compared to Internet Explorer 7 optical page zoom.

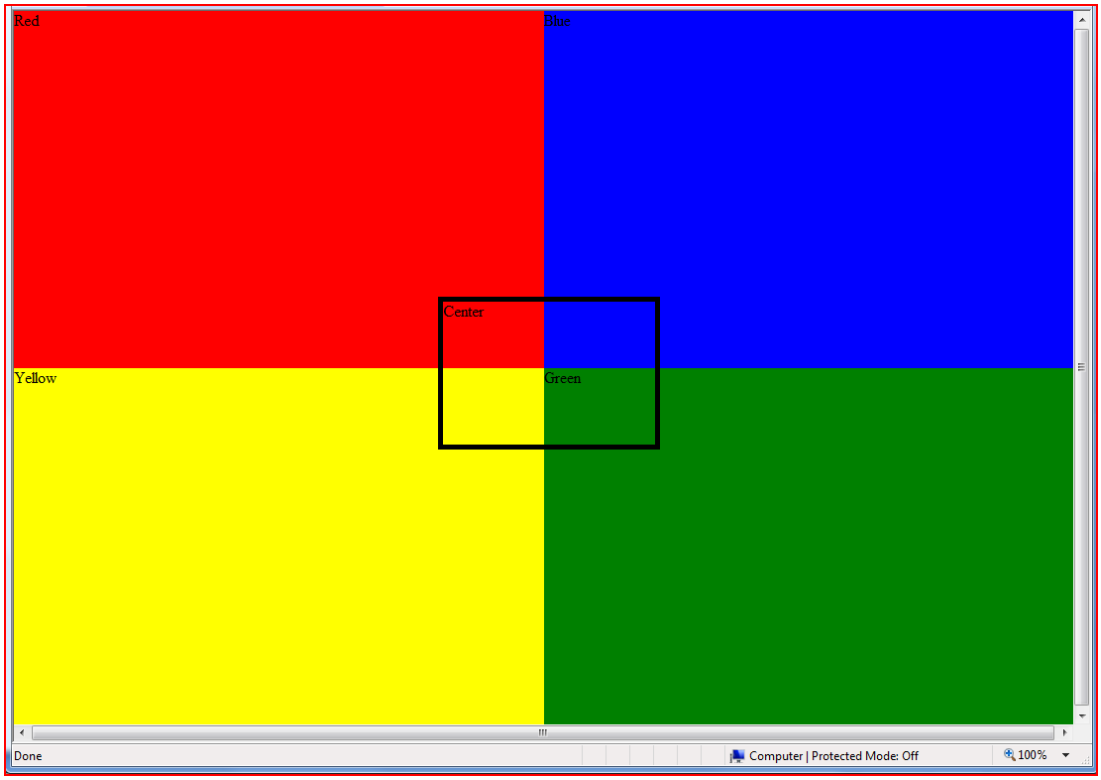
- Four boxes have been absolutely positioned, anchored to the corners of the viewport. Each of them is sized using relative dimensions (%). The box in the middle is anchored to the center of the body.

  This example shows the difference between adaptive page zoom and optical page zoom with respect to relative sizing and positioning. It shows the importance of carefully selecting the anchor for positioned elements when it comes to zoom. In addition, it highlights the difference between using absolute or device dependent dimensions (such as px) and using relative dimensions for positioning. If the center DIV had been positioned using px, it would have moved from the center when zoomed.

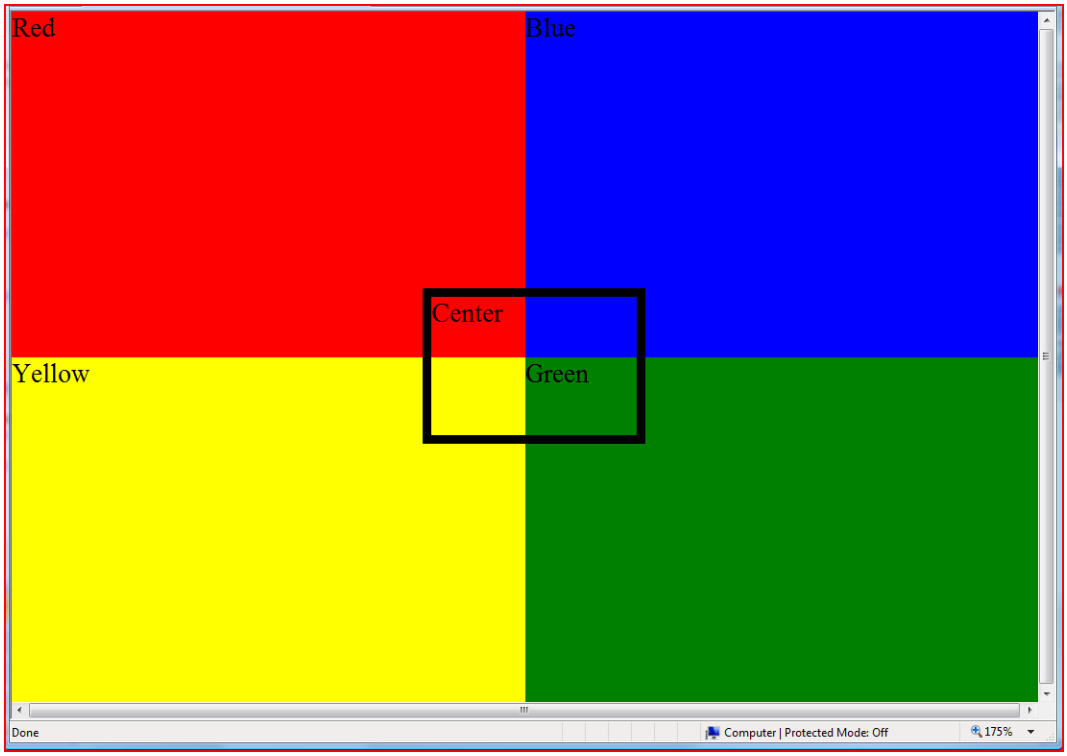
Example page:

```
<html>
	<head>
		<title>
		</title>
	</head>

	<style>

		div {
			width: 50%;
			height: 50%;
			position: absolute;
		}

		#center {
			width: 20%; height: 20%;
			left: 40%; top: 40%;
			border: 5px black solid;
		}

	</style>

	<body style="border:0; padding: 0;">

		<div id="red" style="left:0; top:0; width:50%; height:50%; background-
color:red;"> Red </div>
		<div id="blue" style="left:50%; top:0; width:50%; height:50%; background-
color:blue;"> Blue </div>
		<div id="yellow" style="left:0; top:50%; width:50%; height:50%; background-
color:yellow;"> Yellow </div>
		<div id="green" style="left:50%; top:50%; width:50%; height:50%; background-
color:green;"> Green </div>
		<div id="center"> Center </div>
	</body>
</html>
```

At 100%:



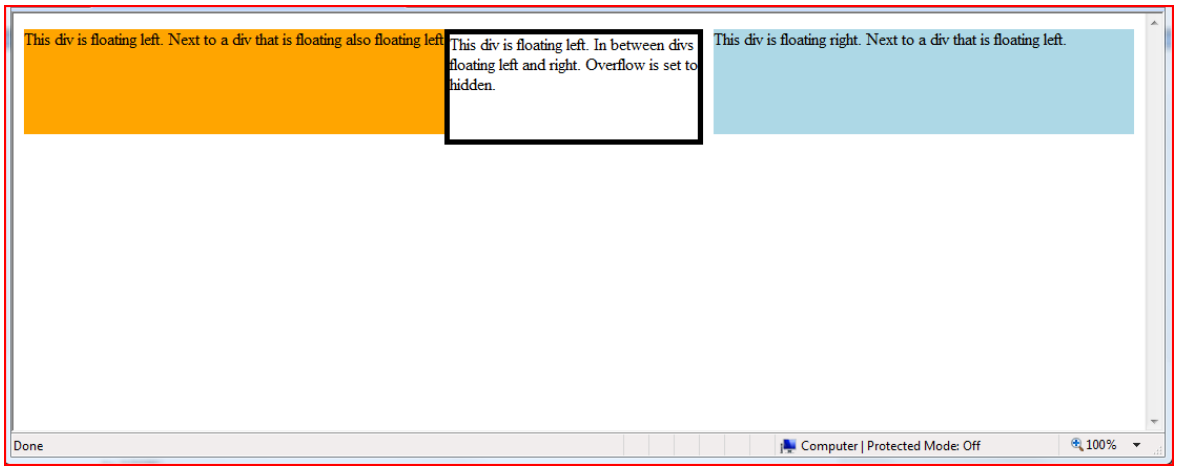
Internet Explorer 8 Beta 1 for Developers adaptive page zoom at 175%:



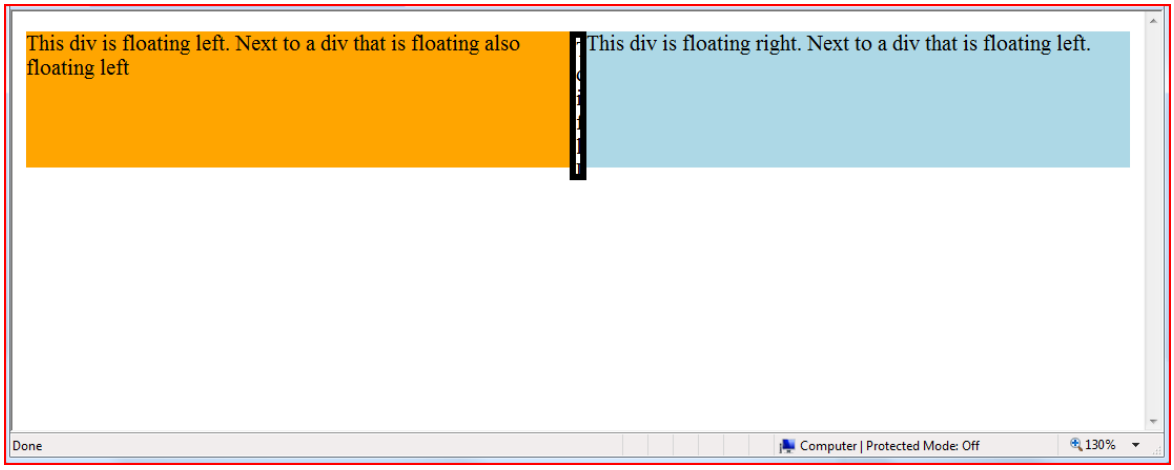
Internet Explorer 7 optical zoom at 175%:

- This example uses floats and elements with overflow to show how layout can shift with adaptive page zoom.

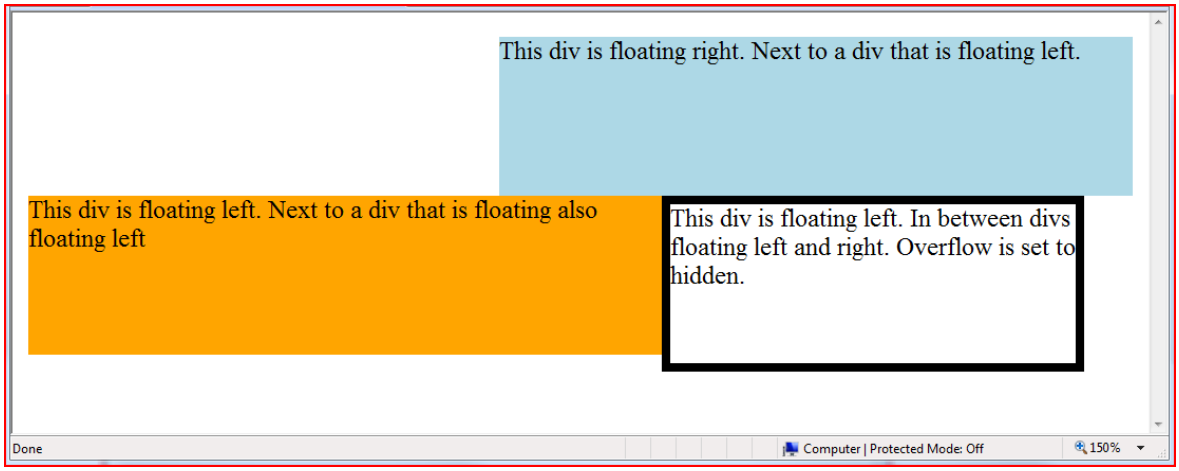
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>
	<head>
		<title>
		</title>
	</head>

	<style>
	</style>

	<body>

		<div style="float:right; width:400px; height:100px; background-
color:lightblue;">
			This div is floating right. Next to a div that is floating left.
		</div>

		<div style="float:left; width:400px; height:100px; background-
color:orange;">
			This div is floating left. Next to a div that is floating also
floating left
		</div>

		<div style="float:left; overflow:hidden; height:100px; border: 5px black
solid;">
			This div is floating left. In between divs floating left and right.
Overflow is set to hidden.
		</div>

	</body>
</html>
```


This div is floating left. Next to a div that is floating also floating left
This div is floating left. In between divs floating left and right. Overflow is set to hidden.
This div is floating right. Next to a div that is floating left.
Done
Computer | Protected Mode: Off
100%

At 130%:


This div is floating left. Next to a div that is floating also floating left
This div is floating right. Next to a div that is floating left.
Done
Computer | Protected Mode: Off
130%

At 150%:


This div is floating right. Next to a div that is floating left.
This div is floating left. Next to a div that is floating also floating left
This div is floating left. In between divs floating left and right. Overflow is set to hidden.
Done
Computer | Protected Mode: Off
150%

In sharp contrast, here is the same page zoomed to 140% in Internet Explorer 7: