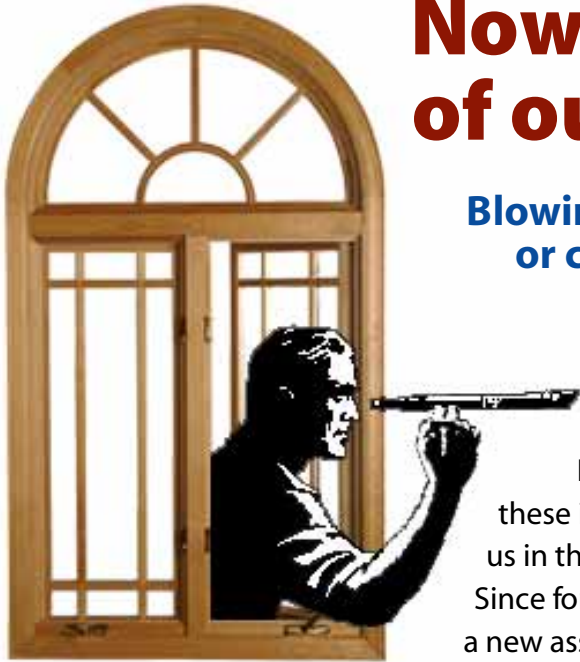


Now is the window of our discontent

Blowing up your browser— or cutting it down to size



As I write this, CNN's Web site is reporting that the U.S. Central Intelligence Agency has just arrested one of its agents on charges of spying for Russia. Now, I don't want draw too many parallels with these international goings-on, but it's clear that those of us in the spy biz are subject to a lot of moving around. Since former Master WebSpy Glenn Fleishman has taken on a new assignment at Amazon.com (see "WebSpy" for No-

ember 15, 1996), I've been recruited from the field to take his place. I'll be keeping you in the know about all things Web-related, and I'll clandestinely pass you the information you need to stay on top of Web publishing and electronic communications. Feel free to E-mail me (no encryption necessary—yet) at webspy@adobe.com with your tips, leads, scoops, and comments.

Spies do have lunch, and I had an especially interesting noontime meal recently with a potential client. As we got to know each other over burgers and fries (a real business lunch, if you ask me), we talked about online design, about where we thought the Web is headed, and about sites that had grabbed our attention in some way. After a few minutes of enthusiasm, we quickly realized that this last list was shorter than either of us was comfortable with (he, an advertising veteran and I, a designer and writer). So, in the spirit of *A Coupla Guys Eating Burgers*, we began singling out particularly bad sites and trashing the overall shortage of good design on the Web.



This approach actually helped us define ourselves to one another. Coming from a print/radio/television advertising background, he was surprised at the lack of high-quality, high-profile, premier sites; with more of a technical background, I bemoaned the lack of control that designers face when designing for the Web. We agreed that there are too many barriers between a site's visitors and its content, whether that takes the form of poor design, ineffective navigation, slow loading times, or just the wall of Plug-ins and animated GIFs that most visitors have to fight through.

Later, after returning to my office (and wishing I hadn't eaten so much), I thought about different solutions for removing those impediments. Many of them—like optimizing a graphic's color palette in Photoshop—can be found here in *adobe.mag's* "How-To" and "Helpful Hints" sections. But some, I discovered, are a little unconventional, including one that solves a pet peeve of mine: resizing the viewer's browser window.

Very few people, it seems, think about the window enclosing their Web pages. It's just a frame, a gray border, a small barrier that prevents the contents

of your Web page from spilling over onto your desktop (wouldn't that be something to see!). But having to resize the window at each Web site you go to can be time-consuming, irritating, and needlessly repetitive.

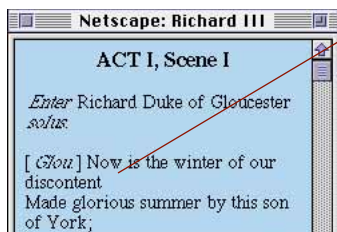
Many designers I know find themselves developing sites that fit into Navigator's default window size—generally about 475 pixels wide—so that the right sides of their graphics don't get chopped off. Others include images of opposing arrows that read, "For best viewing, adjust your browser's window to this width." Some, of course, don't care, and create images at whatever darned size suits them at the time.

However, there are ways of controlling the size of Navigator's windows, although I'll admit at the outset that currently these methods aren't perfect solutions. Weigh the pros and cons of each technique and see which works best for you.

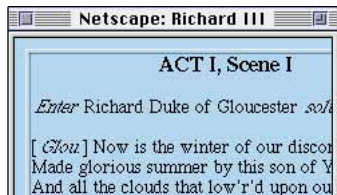
Solution #1: Tables

I've found that tables can be much more useful when they're invisible than

Figure 1



Without tables, a smaller browser window will wrap long text and graphics, ruining the intended visual effect.



Including a single-celled table, however, ensures that your content won't wrap and signals to the viewer that he or she needs to expand the window.

```
<table width=350 border=0>
<tr><td>
<H3 align=center>ACT I, Scene I</H3>
<I>Enter</I> Richard Duke of Gloucester
<I>solus.</I><p>
[ I>Glou.</I> Now is the winter of our
discontent<br>
```

Note that I've set the border to zero, so it doesn't look as if you're framing the section. For the illustrations, I've kept the border to show you where it resides.

when they're surrounding columns of data (or being used as quickie picture frames). If you build your Web page so that the entire contents are contained in a single-celled table, you can specify a width for your page (see Figure 1). This is great if you don't want to design to any default specifications. Normally, words and small graphics will wrap to the next line if your browser's window is narrower than your design. Although this may not be a problem on your 17- or 21-inch monitor, you can bet that most browser windows are not fully expanded on smaller screens, meaning your content is going to get chopped, diced, and otherwise scrambled. Place the whole thing

in a table, however, and your content remains intact and alerts the viewer that he or she needs to expand the window.

But wait, weren't we trying to avoid resizing windows? Yes, and this is where the imperfection of this solution appears ... but also where the folks at Netscape have made at least a stab at helping viewers. If the viewer clicks on the second button from the right (on Windows 95 screens) or the upper-right box (on Windows 3.1 and Macintosh screens) in the window's title bar, Navigator will expand or contract the window to fit the dimensions of the invisible table you've created. In some cases, if the window was already the entire width of the screen, clicking the zoom buttons will make the window contract only halfway to the edge of your table; successive clicks will bring it further in.

Although a step in the right direction, this method not only requires that the visitor resize the window manually (even with the help of the resizing box), but also causes the Web page to be redrawn each time, which can test your viewer's patience. This approach also doesn't give you any control over

the height of your window. However, there's another method of controlling a window's size.

Solution #2: JavaScript

Using JavaScript in your HTML file, as in Figure 2 (see next page), you can specify both the height and the width of a window. You've probably run across JavaScript window implementations. HotWired uses a variation of the JavaScript shown here to create a floating navigational bar that you can place anywhere on your screen, providing an easy method of accessing HotWired's main sections—even if your main window is displaying a different site. For a more dramatic example, check out the Nerdheaven link at the end of this article to see where JavaScripting experimentation can lead. (Many thanks, by the way, to Franke James and Bill James of Nerdheaven for their help in creating the piece of JavaScript in Figure 2.)

Granted, the JavaScript approach is more of a workaround kludge than using tables is, but it also presents some interesting possibilities in exchange

Figure 2

```
<html><head>
<title>Now Loading...</title>
<script language="javascript">
<!-- Hide from non-JavaScript browsers

// --New Window function --//
function newWindow()
{
window.open ("window.html","newWindow", 'scrollbars=no,
menubar=no,toolbar=no,location=no,directories=no,status=yes,
resizable=yes,width=175,height=225')
}

// --Name Window function-- //
function nameWindow()
{
window.name="Old_Window";
}
function closeWindow()
{
self.close()
}

// end hide -->
</script>
</head>

<!-- body content -->
<body bgcolor="ffffff" text="000000" onLoad="newWindow();
nameWindow(); closeWindow();">
<center>
<h3>Now Loading...</h3>
<p>
</center></body></html>
```

This creates a new window and sets its attributes. The "function" name can be anything you choose.

So as not to confuse the browser, we need to name the old window, then execute the "self.close" command to get rid of it.

The "onLoad" command makes everything happen. Separate multiple actions with semicolons.

for a minimum of work. For example, as you'll see in a moment, even a window's on-screen location can be controlled to a certain degree.

My original idea was to access a site that would resize my browser window upon my arrival. Unfortunately, the height and width of an open Navigator window can't be changed except by the viewer, so you have to create a new window to take advantage of this technique. If you have a brief welcoming page that's the default size, for example, you can specify a different size for the windows further into your site.

Figure 3

After the new window is created, the old window gets deleted.



Or, as in Figure 3, you can create a temporary page that gets deleted after the new, resized window appears.

Now, the darker side of kludgyness. The biggest problem with this technique is that not all browsers support JavaScript, which is yet another in the long list of reasons for offering multiple versions of your site (I'll go into this in more detail in a future column). In addition, Netscape will sometimes bring up a JavaScript dialog box that asks you to confirm that it should close the first window. This can be a pain, especially if you've implemented this effect on a corporate Web site, where image is important. Also, the creation of a new window often creates a "flashing" effect: The window is first created at the same dimensions as your existing window, then resized, forcing a screen redraw. Lastly, you can't control where on the screen the new window pops up.

Or can you? Here's a hack built on top of a kludge: When Navigator creates a new window using this JavaScript technique, it positions it below and slightly to the right of the original window's title bar so the two windows appear to cascade. If you want to move the new window closer to the center of the screen, repeat the JavaScript code enough times to open and close a succession of windows—until the final window is where you want it. Be aware that the center of your monitor might not be the center of someone else's; also, the successive flashing produced by the birth and death of many windows could stun some inexperienced viewers.

To kludge or not to kludge

As I said earlier, these solutions aren't exactly what I was looking for, but they do work for the time being. It amazes me that window control wasn't built into the original HTML specs, or at least the early Navigator tags—most applications, from word processors to multimedia applications, include window control as a basic operation.

By extending the capabilities of early HTML with tools like tables and JavaScript, however, we can experiment with the Web experience. As Nerdheaven's Franke James says, "The fun of Web design is assaulting the expectations of the viewer. Surprise them. Hit them in the face (gently, of course)." Taking control of window sizing can help you grab viewers' attention while also breaking down barriers to communication—which is what we all should be aiming for when we design our pages. ●

URLs

Nerdheaven
<http://www.nerdheaven.com/>

HotWired
<http://www.hotwired.com/>

WebReference.com, JavaScript Section (a good source of information)
<http://webreference.com/javascript/>

