

Designing and Implementing Imagemaps

Imagemaps are images with clickable regions under them. An imagemap can be a wonderful way to make a visually appealing page interactive. Or, it can be slow-to-load and confusing, if not done properly. This chapter helps you to do it right. You learn about both server-side and client-side imagemaps. You also learn about the tools you can use for creating imagemaps. Finally, you learn about testing your imagemaps and adding alternative text to increase their effectiveness.

Introducing Imagemaps

Imagemaps work by having an image with defined regions under the image. Each of those regions is associated with a link just as with the other linking elements, such as `LINK` and `A`. The browser or the server calculates the regions under the image based on the shape associated with each region, the dimensions of the region, and the anchor point of the region, which is where the region starts — usually from the upper left-hand corner of the image.

Imagemaps can be effective ways of communicating information without excessive reliance on text, possibly making a site accessible to people for whom English is a foreign language or to small children. This theory breaks down if your site design includes any destination pages that rely on text in English, which almost all sites do. In any case, you can use a large site



In This Chapter

Introducing
imagemaps

Imagemap design

Server-side
versus client-side
imagemaps

Developing graphics
for imagemaps

Using an
imagemap editor

The anatomy of
an imagemap

Adding alternate text



map for a single transit page, but you risk incurring the ire of your visitors if you put too many large, slow-to-load imagemaps on your site.

Imagemap Design

An imagemap can be a valuable addition to your site or it can be an annoyance to your site visitor. How do you know which? You can review some questions to see if your application of an imagemap is likely to elicit favorable responses or to anger visitors.

- 1. Did you keep the imagemap size to a minimum?** You might be thinking you want your initial imagemap to fill a screen. What size screen? You certainly want to design for a 640×480 screen, to keep the image size down and to guarantee everyone can see the whole image. Even if your image is well compressed, you are still talking about a large image. Why not use one or more smaller images with some white space between them and plenty of white space around the margins of the page?
- 2. Can you navigate around your site another way, other than the imagemap?** You definitely want to make sure another way exists. The simplest and least glamorous way is to enclose the names of the pages you are linking to within square brackets [like this].
- 3. Can people with their browsers set *not* to load images still navigate through your site?** This relates to the previous question. In the worst case, you will have visitors to your site who don't see any of your images. Can they navigate?
- 4. Is it obvious what the *hot* regions of your imagemaps link to?** Think of the average sign for a ladies room in a nice grocery store. It usually has the word *women* (wanting not to offend women who aren't ladies?), an illustration of a wheelchair, and an illustration of a baby in a diaper. What does this tell you? Previously the sign also had a stick figure with a skirt on, but that went the way of the word *ladies*. Those illustrations tell you the bathroom is wheelchair-accessible and it has a changing station for babies. These signs are as clear as can be. The images that are part of the hot regions of your imagemaps need to be just as clear. If you have any doubt that your hot regions accurately and adequately convey their destinations, do user testing. Find people who aren't part of the Web design group, who aren't necessarily experts on your product line, and watch them click around. Can they find what they want right away?
- 5. Is there alt text for each hot region?** The alt attribute should be populated for each region so if visitors have any doubt where a link goes, they can place their cursors over the region and read the alt text.

Server-Side Versus Client-Side Imagemaps

Imagemaps can either run on the browser or on the server. They look identical, regardless of where the processing takes place. In the earlier days of the Web, when most Web pages were published by systems administrators or UNIX gurus, server-side imagemaps were the thing to do. Today, when most pages are published by clients of ISPs, server-side imagemaps are a nightmare for systems administrators. The preferred method of delivering an imagemap is with a client-side imagemap.

Server-side imagemaps usually require root permission to write to files shared by everyone on the server. You can see why that model doesn't scale well. Server-side imagemaps also put some processing that can be delegated to the visitor's computer back on the server. This doesn't fit the model of distributed computing most systems administrators are pursuing.

Client-side imagemaps are easy to create and plenty of free or inexpensive tools can help you create them. The information the browser needs is all included in the HTML or in a separate file referred to by the HTML. The imagemapping tool creates this for you based on the regions you draw on the image.



Some of the HTML editors reviewed in Chapter 8 have imagemapping tools built into them.

Developing Graphics for Imagemaps

What kinds of images lend themselves to becoming imagemaps? You can use any kind of image, with any shape of hot regions, but the more complex the shapes of your regions, the more work you must do mapping the image. The important thing is the regions give site visitors a clear idea of where they are linking to.

There is an application of imagemaps you might not have considered. Some sites use narrow toolbars with links to the essential pages on their sites. Figure 39-1 is an example of such a toolbar.

The toolbar in Figure 39-1 is small and attractive. It fits conveniently between the heading of a page or the banner of a page, if there is one, and the content. The toolbar can also be placed at the bottom of the page, to facilitate navigation.



Figure 39-1: This is a toolbar as an imagemap.

Using an Imagemap Editor

Plenty of inexpensive imagemapping tools are available. One nice one is Mapedit, by Boutell.Com. This is shareware that can be downloaded as a 30-day trial version and as of this writing can be purchased for \$25. It runs on just about every platform including Windows 95/98/NT, Mac, and UNIX varieties.

Figure 39-2 shows Mapedit in use.

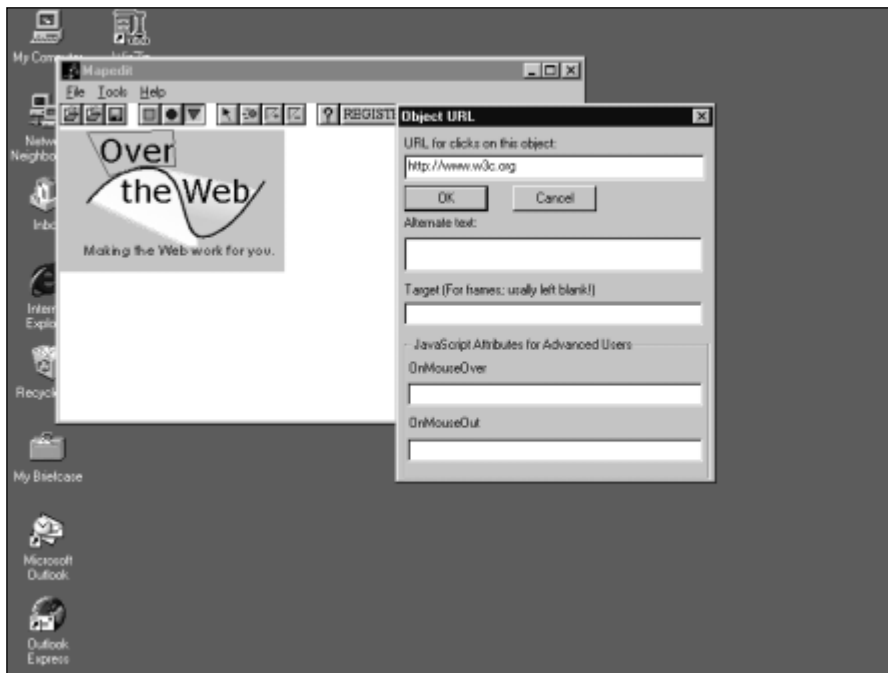


Figure 39-2: Mapedit creates clickable regions under your images.

Definition



The major drawback to the way Mapedit works is it only opens .htm, .html, and .asp files, even though plenty of other file extensions might actually be Web pages (such as .cfm, .cgi, and so forth). If a lot of your Web pages have an extension other than .html, .htm, or .asp, you will want to find a different imagemapping program.

With Mapedit, you define regions right on top of your image using either a circle drawing object, a rectangle drawing object, or a free-form drawing tool. For each region you define, you have a dialog box that requests the URL for the link, alt text for a text description of the link destination, and a target name, if the destination of the link is supposed to populate a frame.

The MAP Element

Essential to creating an imagemap are the `MAP` element and the `AREA` element. The `MAP` element tells the imagemap, “this is where information about clickable regions begins.”

Map <MAP>

Start Tag: Required

Content: `AREA` elements

End Tag: Required

Attributes: `title`, `id`, `class`, `style`, `lang`, `dir`
`name`: required; contains name associated with the usemap attribute of the `IMG` element
`alt`: alternate text

The AREA Element

The `AREA` element defines the regions within the imagemap that link to other pages.

Area <AREA>

Start Tag: Required

Content: Empty

End Tag: Forbidden

Attributes: `id`, `class`, `lang`, `dir`: defined elsewhere
`shape`: “rect”; some browsers support other shapes, but this is the only one in the HTML 4.01 spec
`coords`: coordinates that define the beginning point of the shape and the dimensions of the shape
`href`: URL of the link
`nohref`: Boolean attribute that indicates there is no link
`tabindex`, `accesskey`: see Chapter 21
`accesskey`: accessibility key character
`onfocus`: the element got the focus
`onblur`: the element lost the focus
`alt`: alternate text (recommended)
`events`: see Chapter 48

The Anatomy of an Imagemap

For the toolbar image in Figure 39-1, the HTML looks like this:

```
<A HREF="http://www.itc.virginia.edu/images/navbar.map/">
<IMG width="436" height="52" border=0
src="http://www.itc.virginia.edu/images/navbar.gif"
ismap usemap="#navbar" alt="ITCWeb" ></A>
<MAP name="navbar">
<AREA shape="rect" coords="2,11,77,43" href=" http://www.itc.
virginia.edu/home.html"
alt="ITCWeb Home">

<AREA shape="rect" coords="162,12,260,43"
href=" http://www.itc.virginia.edu/org/"
alt="ITC Organization">

</MAP>
```

You should recognize some of the elements. To begin with, an `A` element is at the very top. Within this is the `IMG` element for the graphic used, which is the same one you see in Figure 39-1. There is also an `ismap` attribute, which is Boolean (meaning if it is present, it is on; otherwise it is off), which indicates this is the image for an imagemap. The next attribute, `usemap`, gives the URL of the map to be used. You've seen the pound sign (`#`) before. It indicates the map to be used is located within the current document. It could just as easily have given a file name or any other valid URL. Finally, there is the `alt` attribute, which you definitely want to use.

Adding Alternate Text

As you have just read, it is important to include text in the `alt` attribute, describing where each of your links leads. This isn't the place to give the URL; this isn't useful. This is the place to give a short text description of what will be found at that page.

Where should you include alternate text? In the `AREA` element for each link and again in the `IMG` element. The `IMG` element's `alt` text will be shown whenever the cursor is positioned at a point that isn't part of one of the hot regions. If the cursor is over a hot region, the `alt` text for that `AREA` element will be displayed.

From Here



Go to Chapter 42 and learn about incorporating plug-ins.

Jump to Chapter 43 and learn about adding Java applets.

Proceed to Chapter 40 and learn about producing and adding sounds.

Summary

In this chapter, you learned all about imagemaps. The most important thing about imagemaps isn't where the hot regions are; it's what the image contains to direct visitors to the correct links. This chapter gave you some guidelines for designing imagemaps. It also explained how to design graphics to meet these guidelines. You learned about client-side imagemaps, server-side imagemaps, and why systems administrators much prefer you use client-side imagemaps. You learned about using an imagemap editor and about the importance of adding alternate text. Finally, you learned about the `MAP` and `AREA` elements and the anatomy of an imagemap.



