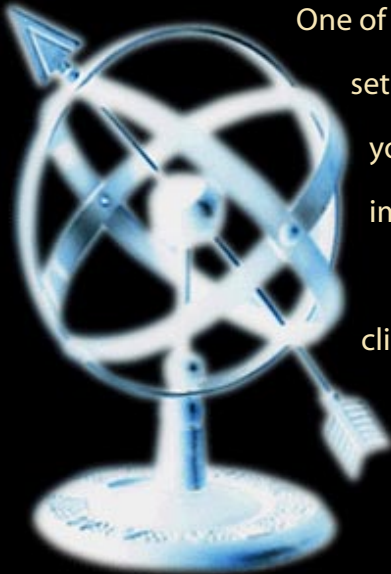


Image mapping



One of the things that mystifies newcomers to the Web is how to set up an image so that when you click on something in it, you're taken to a specific location on the Web. The answer: image mapping.

Image maps allow a user to click on an image, have that click translated into a set of (x,y) coordinates in pixels relative to the image, and then have those coordinates translated into a location or resource on the Internet.

The process only *seems* mysterious.

Setting it up



The image map goes back to the early days of the Web (i.e., when life was less complicated), so it's encoded in a simple attribute to the image (IMG) tag. Just one word with no parameters—ISMAP—gets inserted at the end of a tag, like so:

```
<IMG SRC="my.left.foot.gif" ISMAP>
```

The ISMAP attribute gets interpreted by the Web browser as a command to map out the pixel coordinates of the graphic—i.e., to assign those coordinates to something else (in this case, a URL).

A separate hypertext reference (HREF) tag needs to be set up around the IMG tag to make it work. You have to reference the program that will interpret the coordinates at the server end. This is usually called "htimage" or something similar. You also have to specify the path on the server back to the image map itself, and that's where most people stumble.

Both of these pieces of information—the path to the program and the path to the map—are items that your system administrator or other knowledgeable folk will have to give to you, if you don't know them already.

The HREF combined with IMG would be something like this:

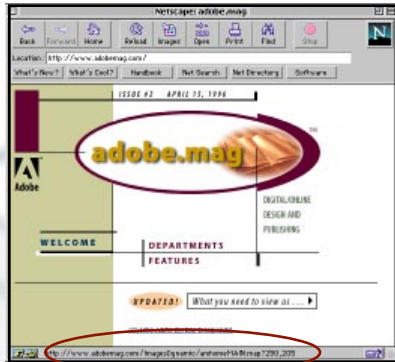
```
<A HREF="/cgi-bin/htimage/~mklauswitz/images/frankenfurter.map">  
<IMG SRC="~mklauswitz/images/frankenfurter.gif" ISMAP></A>
```

Generically, it's this:

```
<A HREF="/path to program/path to map"><IMG SRC="image" ISMAP></A>
```

The image map itself

If you're hovering your mouse over an image with ISMAP set, you'll notice that at the bottom of the Netscape Navigator or Internet Explorer window, coordinates fly by. These coordinates are transmitted to a Common Gateway Interface-enabled program, which interprets them and maps these coordinates against shapes you've defined and assigned to specific Internet resources, such as Web pages.



<http://www.adobemag.com/ImagesDynamic/amhomeMAIN.map?250,205>

The image map is really a set of shapes, locations, and coordinates. The shapes are circle, rectangle, and polygon. The locations are any kind of URL—a Web page (<http://www.bloomsbury.com>), a mailer command (<mailto:fred@wilma.com>), or even an FTP file (<ftp://ftp.lycanthrope.com/pub/update.zip>). The coordinates are the pixel positions in (x,y) on the image (with upper left being the zero point).

The image map format—the syntax for the information—differs from server to server (if you don't know what server you're using, ask your service provider or system administrator). Typical, at least in the United States, is the NCSA server's format, which has been adopted by a lot of non-NCSA servers such as WebStar and NetSite. First comes the shape, next the URL that the particular part of the image map is

pointing to, then the coordinates that define that part of the image map. Polygons consist of any number of (x,y) pairs till the shape is closed. Circles are defined by a center point in parentheses followed by the radius on its own. Rectangles are the upper-left and lower-right points, respectively.

```
poly http://www.myhouse.com/picture.gif (0,0) (150,34) (76,45) (200,16) (0,0)
circle mailto:lexluthor@dc.com (45,34) 140
rect ftp://ftp.boombox.com/ (10,10) (100,100)
```

Most servers and image map programs also allow for a “default” line, which is where users are directed if they click on an area you haven’t mapped. For example:

```
default http://www.myhouse.com/nochoice.html
```

Automating mapping and linking

So, what do you do? Get out a pencil and paper and start writing coordinates down? Well, my boy or girl, in the old days, when we had to push HTML to school uphill in five feet of snow both ways, yes, we did use the old paper

method. Now, however, we have more advanced tools at our disposal.

The best of these—I might be a little biased—is Adobe PageMill (or SiteMill). In the 'Mills Brothers, you can simply draw the appropriate shape on the image itself, type in the URLs you want, and select "Save" from the File menu. Voilà! It automatically creates the map.

Referencing the map in the 'Mills is trickier. With the image selected in "Edit" mode, click on the text icon in the Attributes Inspector palette. In the "Location" field, enter the exact path information, like this:

```
/cgi-bin/htimage/~mklauswitz/images/frankenfurter.map
```

as taken from the example earlier. Press Return. Then click on the farthest-right button in the Attributes Inspector palette. At the bottom, click the "Map" radio button. Save the page, and you're all set! (Note that, if the ".map" configuration file that contains the imagemap information needs to be located somewhere other than the location of the image you were working on, you have to manually move that file to the right place.)

For those of you without the benefit of these applications, check out

Yahoo's section on freeware and shareware utilities that perform this sole function. It's at http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Programming/Imagemaps/.

Client-side mapping

One of the innovations in Netscape 2.0 (it's been picked up by Internet Explorer 2.0) is client-side image mapping. What's described above is called server-side image mapping because the coordinates are sent off to a server, where a program is run (through your HREF reference); the result is that you are taken to the location those coordinates point to. This takes longer and sucks up a little server juice in the process; it can be tedious.

In client-side mapping, servers are omitted. Using a set of new HTML tags, you provide exactly the same kind of directives, but the browser itself recognizes the locations that the image map points to as it loads the image-map page, so there's no delay when you click on a mapped object.

An image map embedded in an HTML file needs to come *before* the image

that's being mapped. The tags `<MAP>` and `</MAP>` surround the image map. In the `<MAP>` tag, you also need to name the image map, like this:

```
<MAP NAME="menu1">
```

This name is used to reference the map later.

The tag for a line in the map is `<AREA>` and gets written generically:

```
<AREA SHAPE="shape" COORDS="0,0,0,0" HREF="URL">
```

The SHAPE is set to circle (coded as CIRCLE), polygon (POLY), or rectangle (RECT); you have to use exactly these abbreviations, not the full words. The COORDS attribute is set exactly as the NCSA map format does on page 4, except with no parentheses. So the circle referenced above would be written on the client side as:

```
<AREA SHAPE="circle" COORDS="45,34,140" HREF="mailto:  
lexluthor@dc.com">
```

The HREF can be any valid HREF. Again, there's a default option (SHAPE="default"), which should come last and which should have no COORDS attribute.

In the IMG tag, you reference the map with the attribute USEMAP, and use a pound sign before the name, just as you do with a NAME anchor:

```

```

Mix and match

You're not limited to using just one of these methods. In fact, the two can co-exist; both Netscape Navigator and Internet Explorer will enable the client-side map and ignore the server-based one if both are noted in the file.

Here's how to include both:

```
<MAP NAME="menu1">
```

```
<AREA ... >
```

```
<AREA ... >
```

etc.

```
</MAP>
```

```
<A HREF="/path to program/path to map/"><IMG SRC="image" ISMAP  
USEMAP="#menu1"></A>
```

The above will allow the seamless use of client-side mapping. See more on this at Netscape's page on the matter at http://home.netscape.com/assist/net_sites/html_extensions_3.html.

Finally, for a demo of extensive client-side mapping, tune your browser to <http://www.film.com>, one of our clients, whose recently redone site uses client-side mapping on all image maps, while preserving the server-side for older browsers.

Utility

The whole point of image maps is to provide more graphical interaction without much overhead. There are ways to create rich interfaces using tags supported only in Netscape (which I'll cover in the future), but image maps show the entire life of the Web in their code—and the client-side option updates them so they're useful with the most modern of browsers. ●