Setting up an Image Map on your web page

Mostly understandable documentation

by Todd C. Wilson
tc@galadriel.ecaetc.ohio-state.edu

Of course, I'm assuming that you have a web page. If you don't, then you're dealing with things that you can't use. Image maps and Map *THIS!* are only used with World Wide Web pages - they have no other use (currently) outside of that realm.

I'm also assuming that you're not a Net God - my documentation is geared towards those people who are just starting to deal with Image Maps, not those who understand the source code to HTTPD. If you want to nit-pick my explanations and suggestions apart, please email /dev/null.

Image map terminology that you will find useful:

"default" is what is used when the user clicks an area on the image that is not defined as an area. You might want to think of this as the "background" URL.
"rect" is short for rectangle. Square area, the most common kind.
"circle" is a circular area. It can be an oval or a true circle. Not used very much.
"poly" is a polygon. Oddly shaped area. Can be up to 64 points.
"URL" stands for Uniform Resource Locator (or Unpronounceable Rude Language). It's the way Web pages get around.

Several things are pre-required in order for your web page to show and handle maps.

First, you must have CGI access. In other words, the system administrator, or the "web master" as I shall refer to him (since it doesn't have to be the same person), must have given your account access to run CGI "scripts" from your web page. The web master does all the maintenance of the CGI scripts (be them programs, Perl files, or shell scripts). Normally, you have access to run scripts. If you think that you might not, contact the web master at your site to find out for sure.

Second, you must have a GIF (not a JPG) image to use as a map. I suggest that you create something together using Paintbrush or your favorite painting program. You can be as fancy or as simple as you like. Just keep in mind that your image must be converted to a GIF and then will be displayed using some very poor algorithms, so you might want to keep the colors to a minimum (like under 100). If you have access to some clip art, you can convert some of it to a GIF image and then pick things out there.

Third, use Map *THIS!* to set the image map data up. You don't have to go full-guns right off the bat. For each area, you need to have your URL. This is the biggest pain of the whole process - Map *THIS!* does the really hard work of letting you visually choose the area, but it can't do all of it for you!

If you're going to serve a document from your HTTP account, you should have something like this:

```
                /~wilson/bio.htm
```

This means that the system would present the file "bio.htm" from the HTTP directory "~wilson". The tilde ("~") tells the server to look in your HTTPD user directory. This varies from system to system, but usually this is your user name. It is also usually refers to your private directory.
The point here is that you must give the full PATH NAME to the file. If you try to give it

```
                bio.htm
```

Map *THIS!* won't complain (it is a legal local URL from within your web page). However, when someone clicks on that specific area of the map when you run the map page, the server would try to run something like the following URL:

```
http://galadriel.ecaetc.ohio-state.edu:80bio.htm/
```

This will confuse the server (it just put the port number with some text) and it will present the "system home" directory.

If you try to fool the server by making it "/bio.htm", that won't work - it will look for a directory called "bio.htm" in its home web directory.

This doesn't mean that you have to code web pages up like crazy just to handle all the possible spots. You can, of course, do a URL like this:

```
http://search.yahoo.com/bin/search?p=todd+wilson
```

And it will run to Yahoo and tell it to search for me.

Finally, once you get your map file set up, you need to get it into your web page. This requires you to have both the GIF and the map file on the web server (in the same directory as the page is a good choice), and to construct a special link in your page. The link takes on this format:

<a href="/cgi-bin/imagemap/**FULLMAPNAME**"><img src="**IMAGE.GIF**" ismap></a>

There are two parts that you need to change: **FULLMAPNAME** is the *complete* pathname of the image map file, starting from the web root. For example, this might be "~wilson/test.map". The **IMAGE.GIF** is the GIF image that is to be used as the background for the map.

Here's what a real map-link would look like:

```
<a href="/cgi-bin/imagemap/tc/sm/test.map"><img src="triangle.gif" ismap></a>
```

The file "triangle.gif" is in the same directory as the page this came from, as is the map file (obviously, this is /tc/sm). However, I need to give the imagemap script program the complete pathname of the map; otherwise, it will complain about (a) there being no conf/imagemap.conf file or (b) it can't find the map file. Either way, this works around the problem. BTW, the conf/imagemap.conf file is maintained by the web master, and contains an alias list of where each image map is stored where on the server. Great, but then it doesn't let users make image maps without bugging the staff, which have better things to do. The documentation from NSCA about serving image maps doesn't cover this, but the source code to imagemap reveals this "option".

That's pretty much it. You'll find that you'll create one, maybe three image maps up and that's it. Usually a navigation bar or screen, which should cut down on the system load, since it only has to serve one image as opposed to six or seven.