NCSA Imagemap Tutorial

Eternal gratitude to Kevin Hughes, *kevinh@pulua.hcc.hawaii.edu*, for his code to find intersections of points with circles and polygons.

This document is a step–by–step tutorial for designing and serving graphical maps of information resources with either the external imagemap CGI script or with the built in imagemap support in NCSA HTTPd 1.5. Through such a map, users can be provided with a graphical overview of any set of information resources; by clicking on different parts of the overview image, they can transparently access any of the information resources (possibly spread out all across the Internet).

Note: Certain newer browsers support Client Side Imagemaps. For more information on Client Side Imagemaps, check out the Spyglass tutorial[7].

Requirements

This tutorial assumes use of NCSA HTTPd[8] version 1.0a5 or later. To use the internal imagemap support, you need NCSA HTTPd 1.5. Some other servers (e.g. Plexus, Apache, Netsite) can also serve imagemaps, in server–specific ways; see the specific server's docs for more information.

Make sure you have a working NCSA HTTPd server installed and running.

This tutorial also assumes use of a browser that supports inlined images, server side imagemapping and HTTP/1.0 URL Redirection.

Compiling the Imagemap Script

If you downloaded or pre–compiled NCSA HTTPd server, or compiled your own, chances are the imagemap program was automatically compiled and added to your cgi–bin directory. If so, you can skip this step. If not, you can download the current imagemap.c from here[9].

Compile the imagemap script by first changing into your ServerRoot directory, and

1 #Requirements
2 #Compile
3 #First
4 #Internal
5 #Example
6 #Real
7 http://www.spyglass.com/techspec/img_maps.html
8 /docs/
9 imagemap.txt

then into the cgi−src subdirectory. Put the new imagemap.c source in place of the old one (if you have the old version of imagemap.c) and then, type `make imagemap` and you should be all set.

Your First Imagemap

In this section we walk through the steps needed to get an initial image map up and running.

➡ *First: create an image.*

There are a number of image creation and editing programs that will work nicely −− the one I use is called *xpaint* (you can find it on *ftp.x.org* in /R5contrib[10]; here's the tar file[11]). It's for UNIX systems running an X interface.

➡ *Second: create an imagemap map file.*

This file maps regions to URLs for the given image. For a list of tools that may help you create a map file, see Yahoo's Imagemap Directory[12]. For instance, there is a program called mapedit[13] that you could use.

A common scheme for an imagemap is a collection of points, polygons, rectangles and circles, each containing a short text description of some piece of information or some information server; interconnections are conveyed through lines or arcs. Try to keep the individual items in the map spaced out far enough so a user will clearly know what he or she is clicking on.

Lines beginning with # are comments. Every other non−blank line consists of the following:

```
method URL coord1 coord2 ... coordn
```

method is one of the following:

- **default** – For the default URL
  - Coordinates: none
- **circle** – For a circle
  - Coordinates: `center edgepoint`
- **poly** – For a polygon of at most 100 vertices
  - Coordinates: Each coordinate is a vertex.
- **rect** – For a rectangle
  - Coordinates: `upper−left lower−right`
- **point** – For closest to a point
  - Coordinate: `thePoint`

URL is one of the following:

- A Local URL
  - ex. `/docs/tutorials`
- A Full URL

---

10 ftp://ftp.x.org/R5contrib/
11 ftp://ftp.x.org/R5contrib/xpaint−2.1.1.tar.Z
12 http://www.yahoo.com/Computers/Internet/World_Wide_Web/Programming/Imagemaps/
13 http://sunsite.unc.edu/boutell/mapedit/mapedit.html

ex. `http://www.yahoo.com/`

coord#

Each coord entry is a coordinate, format x,y. The number depends on the **method**.

Notes:

- each method is evaluated in the order it is placed in the configuration file. If you have overlapping areas, such as a circle inside of a rectangle, you should place whichever one you want evaluated first before the other in the map file. In this case, we would put the circle before the rectangle.
- Also note that it does not make sense to use the default method with the point method because if even one point method is specified, anywhere you click will be considered close to the point and the URL specified by point will be serviced.
- If you will be serving access authentication[14]–protected documents through your imagemap, you MUST use fully qualified URLs, for example

  `http://your.server.com/path/to/protected/file.html`

  otherwise access will be denied.

Here is an example imagemap[15] linked from the Mosaic Demo page. Here is what a map file looks like:

```
default /X11/mosaic/public/none.html

rect http://cui_www.unige.ch/w3catalog    15,8    135,39
rect gopher://rs5.loc.gov/11/global       245,86  504,143
rect http://nearnet.gnn.com/GNN-ORA.html  117,122 175,158
```

The format is fairly straightforward. The first line specifies the default response (the file to be returned if the region of the image in which the user clicks doesn't correspond to anything).

Subsequent lines specify rectangles in the image that correspond to arbitrary URLs –– for the first of these lines, the rectangle specified by `15,8` (`x,y` of the upper–left corner, in pixels) and `135,39` (lower–right corner) corresponds to URL `http://cui_www.unige.ch/w3catalog`.

So, what you need to do is find the upper–left and lower–right corners of a rectangle for each information resource in your image map. A good tool to use on UNIX for this is *xv* (also on **ftp.x.org** in **/contrib**[16]) –– pop up the Info window and draw rectangles over the image with the middle mouse button.

It doesn't matter what you name your map file, **but it does matter where you put it!** Specifically, you cannot have your mapfile reside in the in the top–level of the

[14] /docs/info/Security.html
[15] http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Demo/metamap.html
[16] ftp://ftp.x.org/contrib

DocumentRoot[17], because the imagemap CGI program will not see a "/" in the extra `PATH_INFO` that you are referencing. If this doesn't make sense to you, then just trust me: place your map file in a subdirectory.

Third: referencing your imagemap in your HTML file.

To reference your new map, you construct URLs[18] pointing to it.

For example, if your username is `newbie` and you have created a `sample.map` file in the directory called `path` in your `public_html` home directory, and used the image `sample.gif` for the map, the following line of HTML[19] will reference it:

```
<A HREF="http://www.school.edu/cgi-bin/imagemap/~newbie/foo/sample.map">
<IMG SRC="/~newbie/gifs/sample.gif" ISMAP>
</A>
```

The extra `PATH_INFO` after the `/cgi-bin/imagemap` tells the imagemap program where to find your map file. You'll note that it is possible to request the map file itself by simply requesting:

`http://www.school.edu/~newbie/foo/sample.map`

Or, using the internal imagemap support[20] of NCSA HTTPd 1.5:

```
<A HREF="http://www.school.edu/~newbie/foo/sample.map">
<IMG SRC="/~newbie/gifs/sample.gif" ISMAP>
</A>
```

Fourth: Try it out!

Load the HTML file, look at the inlined image, click somewhere, and see what happens.

Internal Imagemap Support

With NCSA HTTPd 1.5, the server can now handle imagemap queries without launching the external imagemap script. This provides for a faster mechanism and decreased server load.

In order to enable the internal imagemap code, the server must be compiled with `IMAGEMAP_SUPPORT` defined. This is the default. Also, you need to add the magic mime type for the imagemap files in much the same way you would for CGI scripts[21]:

`AddType`[22] `text/x-imagemap .map`

[17] /docs/setup/srm/DocumentRoot.html
[18] /docs/setup/admin/AccessingFiles.html
[19] http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html
[20] #Internal
[21] cgi.html#Anywhere
[22] ../setup/srm/AddType.html

Then, you can reference the map file directly in your HTML instead of through the imagemap script, ie:

```
<A HREF="http://www.school.edu/~newbie/foo/sample.map">
<IMG SRC="/~newbie/gifs/sample.gif" ISMAP>
</A>
```

A Complete Example

The fish demo[23] in another section of this manual used the following configuration files:

Map Configuration File

The map configuration file[24] used for this picture was rather lengthy. I used `xv` to get the coordinates.

Reference the map in HTML document:

```
<A HREF="/cgi-bin/imagemap/docs/info/fish.map"><IMG
SRC="fish33.gif" ISMAP> <A>
```

Real–World Examples

Following are examples of distributed imagemaps on servers in the real world; they may or may not work at any point in time.

- Experimental Internet Resources Metamap[25] at NCSA[26].
- University of California Museum of Paleontology[27].
- National Institute of Standards and Technology[28].
- Server map[29] at NCHPC Information Server[30].

For More Information

For more information, see the NCSA HTTPd documentation[31].

Return to the tutorial index[32]

[23] /docs/info/Scripts.html#im
[24] fishmap.txt
[25] http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Demo/metamap.html
[26] http://www.ncsa.uiuc.edu/General/NCSAHome.html
[27] http://ucmp1.berkeley.edu/
[28] http://www.nist.gov/welcome.html
[29] http://info.lcs.mit.edu/Info/structure.html
[30] http://info.lcs.mit.edu/
[31] /docs/
[32] /docs/tutorials/index.html

Return to administration overview[33]

NCSA HTTPd Development Team / httpd@ncsa.uiuc.edu[34] / 11−5−95

[33] /docs/setup/admin/Overview.html
[34] ../FeedBack.html