

# Map *THIS!*

The **FREEWARE** Image Map Editor for the World Wide Web.

Written and copyrighted (C) 1995 by Todd C. Wilson / Molly Penguin Software. Please read the [legal disclaimer](#).

This program will **only** be of use to you if you are running a Web Page and if your Web Server supports "image maps". If you are not running a Web Page, have no idea what a image map is, or are just starting out with the various things you can do with HTML, **stop now!** Map This is not for you!

Table of Common Topics:

Map This:

[What is it?](#)

[Advantages over others](#)

[System Requirements](#)

[Release Notes](#)

[Limitations of Map This](#)

[Licensing.](#)

Image Maps:

[What are they?](#)

[Reasons to use and not to use maps](#)

[Setting up the server](#)

[The image map file format](#)

[Client-Side Image Maps](#)

And of course, you can perform a [keyword search](#) on this help file. Everything is cross-referenced.

## About Map This

Map This is a program to create and edit World Wide Web image maps. On your tour around the Web, you may have seen things like this:



Clicking on the different items - say, the second glyph, which looks like an "information" icon, might bring up information about the person running that particular Web page. Clicking on the first one might take you to the next page in the information base, while the third one could send email to the author of the web page.

The idea is that you (or someone else) has created a graphic - be it a simple navigation bar like this, or a larger and more complicated graphic. Obviously, it takes some skill to create a decent looking graphic. Map This can't do that, obviously (computers can't create - yet), but it can do the boring, painstaking, and prone-to-error work of "mapping out" the locations that you want the image map to respond to user's clicks. And it adds a lot of bells and whistles (what I classify as "Goodies") to the process, making it much more interesting.

The "official" method (from [NCSA's](#) and [CERN's](#) HTTPD tutorial pages, two of the most widely used HTTPD servers) of creating image maps is to use a painting program, such as Paintbrush, XV, Lview, or whatever, to find the "points" that define the areas of the map. Then you have to write them down, and type them into a specially formatted text file to place on your server. Talk about a lot of work!

Map This takes care of all the "where is this" stuff, in a graphical, user-friendly (for the most part) way. You can drag-n-plot your rectangles, circles, and polygons to your heart's content, reshape them, and then set up what they point to. And you can see what goes on, as you do it. A true WYSIWYG image map creator program!

Actually, I do know what it means. But in the spirit of not embarrassing the guilty party, I won't tell you who said it. Unless you ask.

# Advantages of Map This

There are other "mappers" out there for Windows; however, Map This has some spiffy features that the others I've seen lack:

- It's absolutely, totally, **FREE!**
- It's feature [laden](#). Whatever that means.
- It's faster than most others.
- Written as a Win32 app - runs under [Win32s](#), [NT](#), or [Win95](#).
- Supports moving and reshaping (even point addition/deletion) of Polygons.
- Handles images of almost any size (up to 1280x768).
- Supports both GIF and JPG images, including interlaced GIFs.
- Doesn't require a TrueColor video board - can operate from 16 colors to 16 million colors.
- Handles multiple open files at once.
- Keeps a tracking header comment block in the image map file.
- Keeps track of the image that goes with the map by way of a special comment line.
- Supports descriptions per each area, and one for the entire map.
- Opens existing map files that you might have.
- Docking/floating toolbars.
- Supports [NCSA](#), [CERN](#), and [CSIM](#)/HTML data file formats..
- [Listing](#) of the areas, alongside with the image.
- Allows you to save or load as either .MAP, .IMP, .HTM, or .HTML files.
- Supports user-definable [Snap-To Grids](#).
- Allows the creation of "[Guides](#)" - predefined areas suitable for image "thumbnail" maps.
- User definable fill pattern and color for the areas.
- Built-in Email Feedback allows you to contact the author with bug reports and suggestions.
- Checks your map as you work on it for missing URL's and obscured or "hidden" areas.

Win32s is Microsoft's free extension to Windows 3.1 & 3.11. It allows you to run some 32-bit apps under "regular" windows. Map This will *only* work under version **1.25** or later. [Where to find it.](#)

Win32s can be downloaded by using the following URL:

<ftp://ftp.microsoft.com/softlib/mslfiles/pw1118.exe>

If this location is unusable, you can find this file at many popular FTP systems, and on CompuServe.

Map This was developed under Windows NT Workstation version 3.51. It should work under previous versions (3.1/3.5) of NT Workstation and Server. It has been tested with 3.51, and tested under any and all combinations of the following: Service Pack #1, Service Pack #2, and NewShell Preview (October Release).

Map This runs as a native Windows 95 application. Map This has been tested using Windows 95 Final Beta April, Final Beta June, and the August 24th "Gold Master" release.



## Who wrote it?

Map This was written by Todd C. Wilson. It is Copyright © 1995 by Todd C. Wilson and [Molly Penguin Software](#).

You can contact Todd by using the built-in Email Support function, or by emailing him at:

**[tc@galadriel.ecaetc.ohio-state.edu](mailto:tc@galadriel.ecaetc.ohio-state.edu)**

And support and updates can be found at:

**<http://galadriel.ecaetc.ohio-state.edu/tc/mt>**

This program is totally **FREE!** No charge to you to use it or pass it on to others. I only ask a few things (hehehe):

- If you played with this program and hated it, [email me](#) and tell me why!
- If you used this program and loved it, [email me](#) and thank me!
- If you used this program and have a suggestion, [email me](#) and tell me!
- If you plan to put this on a [CD collection](#), or in a [book](#) on HTML or the Web, please consider giving me a copy of the CD or book - I believe it would be only fair.
- If you would like to bundle this program with your product (either as is or with modifications), [contact me](#). Im easy to deal with.

Be sure to read the [License Restriction](#).

## What is Molly Penguin Software?

It's the pseudo-company that I release freeware under. The name directly refers to my wife and my daughter. "Molly" is the phonetic pronunciation of "mali", which means "small". Therefore, Molly Penguin means "small penguin". There's a lot more story behind all of this, but I sense that you are bored enough as it is...

**Todd C. Wilson**

email: [tc@galadriel.ecaetc.ohio-state.edu](mailto:tc@galadriel.ecaetc.ohio-state.edu)

home page: <http://galadriel.ecaetc.ohio-state.edu/tc>

## Why free?

The main reason is that too many programmers believe that they can make a living writing crappy shareware and charge \$25-\$75 for it. Another good reason is that I believe that making image maps is cool, but how many times will you actually *use* this program? Enough to warrant paying \$25+ for it? I have also gotten a LOT of enjoyment out Internet over the last [fifteen years](#), and had some slack time on my hands. So I came up with something that people could possibly use.

I could go on and on about the utter abuse of shareware, on both sides of the fence (authors vs. users), but I'd probably bore you to death. If you really must hear me vent my spleen, feel free to [email me!](#)

Be sure to read the [license restrictions](#) and [copyright notice](#).

Back before there was World Wide Web, or Windows. When it was UseNet and ARPANet. When we all hacked Unix and VMS and loved it, and never dreamed of stuff like networked Doom. Strong, heady, anything-is-possible days. Ah yes...

# The Dreaded License Restriction

This program is [FREE](#). There is *no charge* to use, copy, or distribute this program. You may place this program on ANY file transfer site, of ANY kind. You may give it to your friends. You may delete it. You may use it to wrap fish in.

You may NOT attempt to turn this program into a virus carrier. If you suspect that you have an infected program, contact the person or site you got it from and then [contact me](#).

This program is provided AS-IS. See below for more legal-speak. Or you can read [my version](#) of a disclaimer.

If you wish to include this on a CD collection, or put it on a disk that is to be included with a book, please [contact me](#). I almost always say "yes", and would appreciate a free copy of the book/disk for my collection.

If you would like to bundle Map This with your software (either as-is or with suitable modifications), [contact me](#).

**Exception:** FTP sites such as Oakland, Garbo, Cica, etc., who create CD collections (or license others to create CDs of their collections) on an annual or semi-annual basis in order to cover operating costs do not have to ask permission. Permission was granted the instant the program was upload (most sites have a paragraph covering this, but why not get it all in there?).

The really ugly part (you can [skip it](#), it's there for the lawyers):

THIS PRODUCT (MAP THIS) IS PROVIDED AS IS WITHOUT ANY WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, TODD C. WILSON AND MOLLY PENGUIN SOFTWARE FURTHER DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED OR STATED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THIS PRODUCT AND DOCUMENTATION REMAINS WITH RECIPIENT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL TODD C. WILSON AND/OR MOLLY PENGUIN SOFTWARE OR THEIR SUPPLIERS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, RECURSIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, PERSONAL INJURY, DISRUPTION OF FAMILY LIFE, OR OTHER PECUNIARY LOSS) ARISING OUT OF THIS AGREEMENT OR THE USE OF OR INABILITY TO USE THE PRODUCT, EVEN IF TODD C. WILSON AND/OR MOLLY PENGUIN SOFTWARE HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO THE RECIPIENT.

LET IT BE KNOWN THAT THE USER OF THIS PROGRAM HAS PAID TODD C. WILSON AND MOLLY PENGUIN SOFTWARE THE SUM TOTAL OF **\$0.00** FOR THE UNLIMITED AND UNRESTRICTED USE OF THIS PROGRAM. ANY DAMAGES AWARDED SHALL NOT BE IN EXCESS OF SAID AMOUNT.

# My Personal Disclaimer

If the [legal disclaimer](#) leaves a bad taste in your mouth, you might prefer this one:

DISCLAIMER: This program and documentation reflects the thoughts, opinions, ideas, and body odor of myself; it does not reflect the thoughts, opinions, ideas, and/or body odor of my company, my friends, my wife, my daughter, my fish, my roses, my dog, or my trash. All rights reserved, all lefts reserved. This software is subject to change without notice. Bits are slightly enlarged to show detail. Any resemblance to actual software, running or hung, is unintentional and purely coincidental. Hand wash only, tumble dry on low heat. Do not fold, spindle, or mutilate. No substitutions allowed. For a limited time only. Type hard, you are making five copies. This software is a void pointer to null where prohibited, protected, declared private, taxed, or otherwise restricted. Software is provided "as is" without any warranties expressed or implied. User assumes all liabilities. Not liable for damages due to use or misuse or inability to understand. An equal opportunity electron employer. No shirt, no shoes, no software. Quantities are limited while supplies last. Quality may vary. Since software is hand-crafted, there will be slight differences in each object. If defects are discovered, do not attempt to fix them yourself, but return to an authorized service center. No Parking. No Standing. No Spitting. No Kidding. Posted no Bills. Parental Advisory - explicit source code. No one under 17 admitted. Keep away from sunlight, pets, and small children. Limit one per family. No money down. No purchase necessary. Cache and carry. You do not need to be present to win. Some assembly and C++ required. Batteries not included. Action figures sold separately. Available in fine shoe stores everywhere. Take a number please. Preservatives added to improve freshness. Safety goggles must be worn at all times. Hard hat area. Sealed for your protection. The buss stops here. Call before you dig. How about a nice game of chess? EXTERN use only. If a rash, redness, irritation, or swelling develops, discontinue use and consult your magic eight ball. Use only with proper ventilation. Avoid extreme temperatures. Store in a cool dry place. Refrigerate after opening. Keep away from open flames and avoid inhaling fumes. Avoid contact with eyes. Do not puncture, incinerate, or store above 120 degrees Fahrenheit. If this software begins to smoke, run, do not walk, towards the nearest exit. Do not place near any magnetic source. Smoking this software may be hazardous to your health. You are not in Kansas any more. I/O, I/O, its off to work I go. A proud sponsor of the local chapter of the old hackers home. Code used in this software was made from 100% recycled electrons. Prosecutors will be violated. No animals were used to test the runtime performance of this software. No extra salt, MSG, artificial color or flavoring added. If ingested, do not induce vomiting. If symptoms persist, delete yourself immediately. If you suspect an overloaded operator, destroy immediately. Constantly volatile when exposed to static pointers. The white zone is for passenger loading and unloading only. Do you like gladiator movies? Auto pilot is engaged during flight. Possible penalties for early withdrawal. Free software offer valid only at participating FTP sites. Slightly higher outside of the continental US. Allow four to six seconds for delivery. Motor vehicles only. Actually, I am a mouse in the middle of an incredibly complicated plot to take over the world. This disclaimer does not cover hurricane, lightning, tornado, volcanic eruption, earthquake, flood, and other Acts of Bob, misuse, neglect, repair, attempted modification, bugs in the code, damage from improper installation, incorrect line voltage, cosmic rays, missing or altered serial numbers, attempting to actually run the software, sonic boom vibrations, electromagnetic radiation from nuclear blasts, and incidents related to airplane crash, ship sinking, motor vehicle accidents, leaky roof, alien attack, broken glass, falling rocks, mud slides, forest fire, flying squirrels, verbal assaults, or house arrest. Other restrictions may apply. Your mileage may vary. This software is copynighted by Todd C++ Wilson.

## System Requirements & Etc.

Map This is a 32-bit program. This means it runs as a native Windows NT or Windows 95 application - much faster than a 16-bit program. In order to get Map This to run under Windows 3.1 or 3.11, you must have the [Win32s](#) extension installed. The normal function of Map This does not require an Internet connection or even WinSock to be installed.

However, if you wish to use the Email Support function, you must have WinSock installed and be connected to Internet in some fashion. If you don't know what WinSock is, or are confused about Internet and email and all those strange words, **STOP NOW**. You will get *NOTHING* out of this program - it is not for you!

The ETC.: Why 32 Bit?

Simple. Windows 3.1 is dead. Microsoft has shoved Windows 95 on us. Of course, the real programmers have been using Windows NT for years now, and know all about the Joys of Thirty Two (complete with the how-to photos). And there are certain programmer-level stuff in Win32 that makes life really nice for power-nuts like myself.

However, Windows 3.1 is going to be on a lot of machines for a long time. That's why I wrote Map This to use features that are common to all three 32-bit platforms: Win32s on Windows 3.1; Windows 95; and Windows NT. And since Win32s is free (like other great software I could mention), and it supports about 70-80% of the Win32 system, there doesn't seem to be a reason to make two versions.



## Limitations of Map This

Map This is a pretty decent program for what you paid for it. But let's face it, it was written by one person with input from some others, and reflects what I think is "cool" and "useful". If you have a suggestion, please, feel free to [contact me](#) and tell me!

- Map This is, as mentioned, a 32-bit program. I do not feel this is a true limit because of the proliferation of Windows NT, Win95, and Win32s.
- Map This only uses a mouse as input. It's a semi-CAD program, after all.
- Map This does not do truly stringent checking of the values for the areas. You can have "offscreen" areas that will never be hit, and illegal HTML commands.
- Map This supports up to 1024 areas per map. It will generate an error if you try to add more areas or load a map file that has more than this. This is not a true problem, since most servers will die if you have this many anyways.
- Each polygon area is allowed up to 64 points. Some Web Servers put a lower limit - I know of one that only allows up to 16 points.
- The scrolling of the image when you are zoomed-in is slow.

# Release notes for version 1.20.0

Version 1.20 adds several new features. The most important is that it now supports HTML 3.0's Client Side Image Maps. There are other things, too:

## **Heap Corrupted Error**

This error would show up if certain GIF images were loaded. The GIF decoder would overwrite it's memory area by one byte. Okay, so what - why the big error? Well, it's a lot better to have a warning and a safe exit than no warning and a crash. This has been fixed.

## **Area Fill**

Another feature added is the option to have Map This fill in the areas with a pattern and color. This is nice when you have a complicated map and need to see things. See [Preferences](#).

## **Handles**

You can now turn the handles on or off. Added because people were doing very complicated maps and having trouble. See [Preferences](#).

## **Client-Side Image Maps**

Map This now supports the *DRAFT* HTML 3.0's Client Side Image Map specification. If this specification changes, then Map This will also have to change. See [Image Map Formats](#).

## **Save As dialog box**

This file dialog is now modified by Map This to add some more controls to let you change the map information at the last moment. See [Saving an Image Map](#).

## **Cut, Copy, Paste**

You can now use the normal Edit menu options. Even between maps.

## **Test the Map**

A lot of people wanted this. Okay, here it is. The best way to test is still to put the files on your server. See [Testing the Map](#).

## **Grids**

If you entered a zero value for either the width or height, then Map This would "lock up" - forever. You would have to then manually remove some registry keys (YUCK!). Now it checks and refuses to continue. See [Snap-To Grids](#).

## **Email Support**

The counter seemed to upset some people. So I changed it.

*The following items were changed in version 1.10:*

## **Memory:**

This was the biggest improvement of the program. Under Win32s, you could easily run out of resources with a decent sized map (if you run NT like any sane person does, you'd never have this problem). Map This 1.10 now uses more "temporary" GDI objects, and does very aggressive memory management (yes,

it does take a little more code, and yes, it does slow the program down a little - very little) If you watch the resource percentage value, you may suddenly see it dip from about 70% to 54% - and then jump back up.

### **Polygons:**

Another area of work is in dealing with Polygons. Map This 1.10 allows you to add and delete points from a polygon using the right mouse button. There was also one bug fix in that if you saved a map in a zoomed-out state, the polygon values were all screwed up.

### **Grids:**

Snap-To Grids are supported by the program. You can control how big they are, where they start, and how they look like. This affects the mouse movement when you make a new area or drag an old one. See [Snap-To Grids](#).

### **Guides:**

Another feature of Map This 1.10 is that it can create "Guides" or "Template" of regular rectangular areas. I put this in because I was creating a bunch of thumbnail maps and was getting very tired of doing it by hand. See [Creating Guides](#).

### **Preferences:**

You can now customize the colors used for the lines "styles", and select the default image map type ([NCSA](#), [CERN](#), or [CSIM](#)). See [Preferences](#).

### **New Areas:**

By default, Map This does not prompt you for new area information. Some people like it this way. Some don't. You can turn this on or off. See [Preferences](#).

### **Context/Right-Click:**

Right-clicking in the map between the areas will bring up the control menu of the common things - select tool, zoom in/out, grids, etc. Right-clicking on an area will let you delete the area, move it up or down in the map list, or edit the area's information. Right-clicking on a Polygon point will do the same, but also allow you to delete the point. Right-clicking on a Polygon line gives you the option to insert a new point. See [Right Click Menu](#).

### **Background Area Checking:**

Map This will also check your areas out for you if you don't move the mouse for a tiny amount of time (this is called "idle time checking" - Word 6.0 does this with its "background repaginate"). If an area is missing a URL, you will get a warning icon in the status bar. If you have an area that is obscured, you will get an unhappy face in the status bar.

### **Status Bar:**

The status bar is now fully enabled instead of the plain vanilla one in 1.00.5 - it now shows the zoom level, the mouse position in the map, and the size of the movement. It also shows the warning icon (missing URL) and the unhappy icon (area obscured). Moving the mouse onto these icons will put up a tool-tip text box telling you what is wrong.

### **Shift key:**

In order to "grab" those obscured areas, you either have to shuffle the list or move the area covering the other one out of the way. But if you hold the shift key down, Map This will scan the area list in reverse, allowing you to grab that bad area.

**CERN map files:**

The CERN format does not support comments in the file. Map This 1.10 works around this by placing specially formatted "rectangle" areas in the list.

## Things *still* to be done to the program

1. Check the boundaries of the areas to see if they fit within the image when loading (although it is not illegal to have "offscreen" areas) and warn you.
2. Allow "points" to be used (closest-point-wins scheme) for [NCSA](#) maps. Nobody has asked for, and since [CERN](#) and [CSIM](#) don't support it...
3. Put some code in to allow Map This to mesh with the Win95 shell. This really depends on what I think of Win95 after the final release, and how sick I get of the hype.

## What is an Image Map?

An image map is a graphic that has certain areas on it defined to "do something" when the user clicks on each of the area. These areas can do whatever normal HTML links do - email, ftp, gopher, etc. Two very popular uses for image maps are navigation bars and thumbnail sheets. Another common usage is to take a graphic of a country (say, the US or UK) and define areas on the map to take you some where. For example, I could use a graphic of the United States and define an area that covers the state of Ohio to take you the web page of the Ohio State Department of Water (or whatever).

If you think youve never seen an image map before, youre wrong! Most help files for Windows applications have some sort of graphic that is a screen capture of a dialog, and have information about the buttons. In fact, the help file you are reading right now has a lot of them! The same concept carries over to image maps on the Web. However, you will not see a change in the cursor when you move over various areas of a Web-based image map (unless a future browser supports such a thing).

Next: [the Pros and Cons of using image maps.](#)

# Pros and Cons of using Image Maps

There are some very real issues to seriously consider before you start to work on creating an image map and trying to get it to work on your server.

In favor of using image maps:

- ✓ They look cool and everyone else is using them (such as company Web pages).
- ✓ They can spice up an otherwise boring page.
- ✓ They are great for letting the user interact with a catalog of images (called thumbnails, montages, or callout sheets, depending on your field).
- ✓ They also work very well for navigation bars (forward, back, home, jump, email, etc.).
- ✓ Transferring the image from the server to the browser involves only a single file as opposed to a lot of icon images. This involves less browser-to-server command request transactions.

However, there are some problems:

- ✗ If you don't have some artistic talent and/or a decent painting program, you're limited to what you can use. And nobody loves a terribly drawn picture.
- ✗ They usually require a large graphic and a fast link. This means that modem-based users will probably suffer. So the savings that you might get in browser-to-server request transactions might then get used up by the act of transferring the image.
- ✗ Text-only browsers can't see them.
- ✗ Setting up the html file on the server is a major hassle.
- ✗ Whenever the user clicks on the image, the server has to process a special program. This leads to system slowdown and (sometimes) long turn-around time. **However**, this problem has been mitigated to some degree by the rise of HTML 3.0s Client Side Image Maps.

Note that the last point can be the *major*, if not deciding, factor in determining if you can use image maps in your web page. The next section covers this and other common problems.

Next: [Should \(or can\) I use Image Maps?](#)

## Should (or can) I use Image Maps?

Should you use them? That's a more of a personal question than a practical one. If you feel that your page would benefit from them, or that you'd like to try and see, sure, do it.

Whether or not you can use image maps depends on your *server*, or more accurately, your *account* on the server.

What kind of limits your account has determines if you can deliver image maps to the public at large. There are several factors that may or may not apply to your account:

1. Does your account allow you to run "[CGI](#)" programs? CGI programs are programs that are run when certain actions happen or links are accessed. This can range from delivering a random quote to processing fill-out forms. When the user clicks on an image map, the request is normally sent to a program called "imagemap" or "htimage" to check where and what the user just did. If your account does not allow CGI programs, then you cannot serve standard image maps to your users.
2. Does your account have a storage limit? Most images that are used for maps are rather large, generally from 50k to 80k in size.
3. Does your account have a transaction limit? This may be expressed in "hits" or "accesses" per day/week/month, or total number of bytes transferred per day/week/month - or both! If you have a limited account, then you probably can't use image maps.

If your account is not allowed to use CGI programs (usually for security reasons), then you will have to limit yourself to Client Side Image Maps, which further limits your audience.

Next: [Setting up the server](#).



CGI stands for Common Gateway Interface. It basically refers to a standardized command protocol between the Web Server and a server-controlled program. These programs can do things from validating access, to keeping usage counts, and of course, processing image maps.

# How do I get image maps to work on my server?

Sadly, this is a very server-specific question. The answer comes down into two categories: will you be using either Client-Side Image Maps (which doesn't matter about the server but does matter about the browser, therefore limiting your audience), or using server-side image maps (which does matter about the server but not the browser, so it doesn't matter what your user is using to view your map). If you can, you may want support both at the same time.

## The traditional way: Server processed maps and HTML 2.0

The majority of your users will not have the latest and greatest version of whatever browser for their system for some time. So you should consider keeping support for these browsers by using the normal method of accessing image maps under HTML 2.0.

Using image maps under HTML 2.0 requires you to format the line in your HTML document like this:

```
<A HREF="themap.map"><IMG SRC="themap.gif" ISMAP></A>
```

What this means is that the browser is being told (via the `ISMAP`) that this image IS a MAP, so that when the user clicks the mouse in it, it sends back the given `HREF` value with the mouse position appended after it. For example, if the user clicks at position 50,10, the browser sends to the server something that looks like this:

```
http://www.foobar.com/directory/themap.map?50,10
```

(Where "www.foobar.com" is the name of the server, and "directory" is where the html document is stored at)

If the server you are using is one of the newer, more advanced ones, it has all the stuff built into it to understand and process image map requests. Netscape's Server is one of these, and most current commercial NT- and Windows 95-based servers are also like this. However, if you are among the many that are using older software (say, for example, your school's UNIX system), you will discover that the above example will do nothing more than display your image map file to you! Your server needs to be told to run a special program to process the map. This is usually called "imagemap" or "htimage". Contact your system admin to find out exactly what you need - and if you are allowed to use it!

Another problem with using server processed image maps is that you cannot "browse" them locally. This means that even if you have the html file, the image, and the map data on your local hard drive, you will not be able to click on an area and have it work. All that will happen is that you will see the text of the map show up in the browser's window! In order to test the map, you have to put it and all the needed data on the server - and run it live. There is no other way to do it.

The following example is a general idea of what you need:

```
<A HREF="/cgi-bin/imagemap/folder/themap.map"><IMG SRC="themap.gif" ISMAP></A>
```

(Where "folder" is the directory where the map is stored)

Wow! A real mess, right? Let's break this down a little bit to make it a bit more understandable. By the way, this line is what you would use under [NCSA](#)-based servers; [CERN](#) is very similar, but just different enough to make life interesting.

The "/cgi-bin" refers to a special directory that holds special programs that the system administrator has determined are "safe" for use from within HTML documents. The "imagemap" is the program that will be run. The rest of the line "/folder/themap.map" is the parameter that is fed to the "imagemap" program in addition to the mouse's position - sort of like `DIR *.ZIP` from the MS-DOS command line.

There is no guarantee that this example will work for you. Above all, contact your system administrator for what you need to do to get image maps to work.

## The new and improved way: Client Side Image Maps and HTML 3.0

With the creation of the new HTML 3.0 specification, one of the more useful things is that the image map data can now be processed by the browser. This has several advantages - the data is usually stored within the html document, making management simpler; since the data is in the document, and in a single, consistent format, it makes moving between servers (or from local browser to live-on-the-web) easy; and the browser does all the processing, reducing the server load. The trade off is that you can't place "hidden" areas on the map for your close friends to use (for whatever reason), and not all browsers

support HTML 3.0 yet. At the time of this writing, only Netscapes Browser 2.0 (in beta), and Microsoft's Internet Explorer (for Windows 95) supports client side image maps.

Compared to server-based image maps, Client Side Image Maps (which I shortened down to just CSIM) are much easier to set up and maintain. Since all the information is controlled by the HTML file, you can stop worrying about what software your server is running, and if your account can use image maps or not, and start worrying about whether or not your map looks good, and if you need more storage for those images or not.

Using CSIMs is almost the same as using HTML 2.0 image maps, but a lot shorter:

```
<IMG SRC="mymap.gif" USEMAP="#maptag">
```

Yes, that's right, you don't need the HREF "wrapper" around the image tag. However, you probably will want to use it, since it will allow you to define a "default" action to occur should the user's browser not support CSIMs, or if you have a hole in the image map.

The `USEMAP` refers to the name (in this case, `maptag`) of the map that is within the HTML file. This is the name you gave the map when you saved it as a CSIM file (probably into the same file you are now editing). If you want to refer to a map in another html file, you do this:

```
USEMAP="someother.html#othermap"
```

This is generally known as an "external" image map. If you want to know more about this, please refer to Advanced Client Side Image Maps.

### **Best of Both Worlds: Server-Side and Client-Side maps**

Resistance is futile - the new browsers are warping forward, and HTML 3.0 will overrun your federation. But if your server account allows server-side maps, why not use both? Thankfully, the HTML 3.0 specification on CSIMs provide for a way:

```
<A HREF="/cgi-bin/imagemap/folder/themap.map">
<IMG SRC="themap.gif" ISMAP USEMAP="#themaptag"></A>
```

What this monster does is tell the browser that it can be used as both types of image maps. The `USEMAP` tag has a higher precedence than the `ISMAP` tag, so if the browser supports CSIMs, it will use them. If the browser doesn't support CSIMs, it will ignore the `USEMAP` tag (just like Mosaic ignores Netscape's `<BLINK>` tag), but will see the `ISMAP` tag, and use that. Easy as falling off a log. And in a few years, or when you move your account, you can delete the part about the `ISMAP` and pretend you never had to deal with it.

The problem with using both formats is... using both formats! You will have to save your map in two different formats, in two different locations. The CSIM can be saved in the HTML file where it is used, but the [NCSA](#) or [CERN](#) version will have to be saved as an external file (in this example, "themap.map"). As you can see, updating the maps can become a problem, unless you keep careful track of what is what and which one is supposed to be the "master" map, and the other the "copy".

Next: [Inside an image map](#).

# The Image Map file format

If you are using Map This to create and maintain your image maps, you probably won't care what a map file looks like. If so, then you can skip this section.

Currently, there are three standards for image map files: [NCSA](#), [CERN](#), and [CSIM](#).

NCSA and CERN map files are stored separately from the parent html file, and are specially formatted text files. CSIM "files" are actually part of the parent html file.

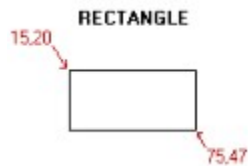
## NCSA: National Center for Supercomputing Applications

NCSA's format looks like this:

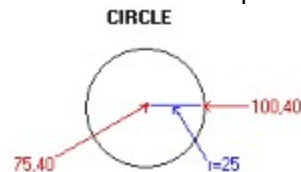
```
#comment
rect URL x1,y1 x2,y2
circle URL cx,cy x1,y1
poly URL x1,y1 x2,y2 x3,y3 ...
```

NCSA interprets any line that starts with the '#' sign to be a comment for a human, and it ignore it. Map This further extends this in a safe manner by placing a dollar sign after this, so you get the digraph #\$. Map This uses these special lines to keep tracking information about the image, author, and other information in the map file.

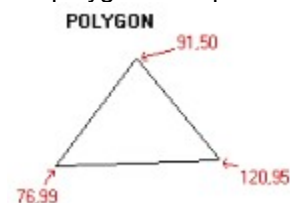
Lines that start with the word "rect" are rectangles. These are the most common area types. The [URL](#) is the item attached to this area, and the x1,y1 x2,y2 coordinates are used to define the box that covers the rectangle. The picture below show what this looks like.



circle lines look exactly like rectangle lines. However, the first pair of numbers is the center of the circle, and the second pair represents a point on the edge of the circle. Confusing, right? The picture below shows a circle with the center position given as 75,40, and a point on the edge as 100,40 - this makes the radius equal to 25.



poly means "polygon". After the URL comes several pairs of numbers. Each pair represents a vertices in the polygon. The picture below shows a simple 3-point polygon.



## CERN: European Laboratory for Particle Physics

CERN's format looks like this:

```
rect      (x1, y1) (x2, y2) URL
rectangle(x1, y1) (x2, y2) URL
circ      (cx, cy) r URL
circle    (cy, cy) r URL
poly      (x1, y1) (x2, y2) ... URL
polygon   (x1, y1) (x2, y2) ... URL
```

Servers that use the CERN format usually allow for both the shorthand (rect, circ, poly) and the longhand (rectangle, circle, polygon) of the name.

As you can see, CERN's format looks very similar to NCSA's format - the only difference is that CERN puts the numbers before the URL, and brackets the numbers with parenthesis.

CERN also defines the circle in a more understandable format - center, then radius.

However, CERN doesn't allow for comments in the file. This causes problems with Map This and other programs that rely on these comments. Map This works around this problem by placing a special area in the map that defines a rectangle that is outside of any possible image, and encoding the comment text into that "areas" [URL](#).

## CSIM: Client Side Image Maps, HTML 3.0

CSIMs are not "flat" files like NCSA and CERN - they are part of the actual HTML document. As such, they look extremely weird to the normal user.

```
<MAP NAME="refname">
<AREA SHAPE=method COORDS="positiondata" HREF="link" ALT="comment">
...more of same...
</MAP>
```

The CSIM is started with the <MAP...> opener and closed with the </MAP> - just like standard HTML.

**refname** is the reference name of this map. This is similar to a "page anchor" idea, where you refer to a jump-to link by "#whatever". All CSIM HREF callers use refname like either of these:

<IMG SRC="pic.gif" USEMAP="#**refname**"> - map is in the same file as the caller

<IMG SRC="pic.gif" USEMAP="xyz.html#**refname**"> - map is in another file.

**method** is the method of the data. CSIM supports only three methods: *RECT*, *CIRCLE*, *POLYGON*.

NetScape and Map This supports a fourth type: *DEFAULT*, which defines what to do if the user clicks outside of an area. Each method determines what the position data is. If this is missing, then the method defaults to RECT.

**positiondata** is a collection of numbers, separated by commas, and is specific for each method type:

Rectangle: x1,y1,x2,y2 (notice the comma between the pairs). Defines the top-left and bottom-right of the rectangle.

Circle: cx,cy,r - center position and radius. Just like the CERN format - true circle only.

Polygon: x1,y1,x2,y2,...xn,yn - each pair is separated by a comma. Each pair defines a vertices of the polygon.

**link** is the html link. It can be either a reference or an absolute link. References from the base document depends on either the map is internal to the caller, or external. The references are taken from where the map is stored *at*, not where it is called *from*. See [More About CSIMs](#).

**comment** is the comment for this item. This can be shown to the user by the browser, or may be used by the browser to show a list of items to choose from.



# Interesting things about image maps you should know.

## 1. The order matters.

This is important - the map is searched "top down" for an area that the mouse's position falls in. The first area will get checked before the second, etc.

## 2. Areas can overlap.

If you think about #1, then this should start to worry you. If areas overlap, then the second area (the one "underneath") will not get checked in the spot where the areas cross.

## 3. Areas can be obscured.

Since areas can overlap, and the search order matters, you can quite easily have an area that will never be accessible by the user - the areas higher up the list always get the click. Map This checks for this and displays the unhappy face in the status bar when it happens.

## 4. [NCSA](#) and [CERN](#) maps require full URLs.

This means that the URLs in NCSA and CERN maps must have the fully qualified command - for example, `http://www.big.com/joe/bob.htm` will work, and `/joe/bob.htm` will not.

## 5. [CSIM](#) maps can use partial URLs.

On the other hand, CSIMs can use the `/joe/bob.htm` just fine.

Also see [More About CSIMs](#).

## More about Client Side Image Maps

The first thing you need to understand is that the HTML 3.0 specification is still a draft specification - a work in progress, if you will. However, the CSIM subset of the specification is small enough and simple enough to expect it will not change.

The second thing is that CSIMs are just html files - in fact, they are inside of html files. It's best to think of a CSIM just like an "anchor" definition in html. For example:

```
<A NAME="download">
```

Places a "bookmark" in the file that you can then refer to from either within the document, or outside of the document. Assume that I have a file called "words.html". Inside of this document, I have an anchor that is setup as `<A NAME="jumphere">`

To go to this anchor from within words.html, I just use `<A HREF="#jumphere">`

To go to this anchor from outside of words.html, I use `<A HREF="words.html#jumphere">`

The same concept carries over to CSIMs: you can refer to them within the same document

```
<IMG SRC="something.gif" USEMAP="#themap">
```

or from another document:

```
<IMG SRC="whatever.gif" USEMAP="something.html#themap">
```

The browser is responsible for requesting the correct html file from the server and processing it's map.

Since the map is an html file, you can use based-from-current document style of commands. You can use `#jumphere` as a URL, for example - and it will work!

If you are going to use based-from-current document commands, remember if you are using the map from another document, the "base" is the file that contains the map, not the "caller" document.



# What is HTML?

HTML stands for HyperText Markup Language. HTML is the language of the Web (parley you HREF?). HyperText means text that refers to other things, and you can go there by clicking on the highlighted word. A book may have "See also: Road Rules on page 291", but with HyperText, you have "read the [Road Rules](#) for more information".

Markup is what you do with a highlighter from the bookstore - you outline text, write notes, etc. Basically, you turn a book into something that gives more information.

HTML looks like this:

```
<BODY>Four scores and seven beers ago, we witnessed the great <I>loss</I> of  
our team.<BR>It was the best of times, it was the worst of times. Get your <A  
HREF="http://www.aol.com">free disk</A> here!</BODY>
```

And the browser turns in into this:

Four scores and seven beers ago, we witnessed the great *loss* of our team.

It was the best of times, it was the worst of times. Get your [free disk](#) here!

Almost all browsers currently in use support what is called "HTML 2.0". HTML 2.0 is a specification defining the rules and operating of the language commands. A new specification, HTML 3.0, is currently in the draft stage, but several browsers are implementing some of the more stable items - client side image maps, for example.

America Online has a well-deserved reputation of giving out their sign-up diskette to everyone. I found one in my box of Fruit Loops yesterday.

Rule #1: Get out of my way.

Rule #2: Yes, I do own the road.

Rule #3: I stop for no reason.

## What is HTTPD?

HTTPD stands for http "Demon". It's the program that is run on a computer to act as the "web server". When some is referring to their "server" or "web server", they mean HTTPD. There are several flavors of web servers, but most either conform to [NCSA](#) or [CERN](#) style of image maps.

## What is a URL?

URL stands for Uniform Resource Locator. It is a "command" to the Web Server to tell it what kind of information to get and where to get it from. These commands usually start with "http" for web pages, "ftp" for file transfers, "gopher" for gopher services, "news" for UseNet news, and "mailto" to send a mail letter. [Examples.](#)

What is a URL?

URL stands for Uniform Resource Locator. It is a "command" to the Web Server to tell it what kind of information to get and where to get it from. These commands usually start with "http" for web pages, "ftp" for file transfers, "gopher" for gopher services, "news" for UseNet news, and "mailto" to send a mail letter.

[Examples.](#)

# Examples of some URL's

(you can click on each URL type to get more information):

This is a Web page run by Microsoft:

<http://www.microsoft.com/>

This is the Travels with Samantha book/photo tour:

<http://www-swiss.ai.mit.edu/samantha/table-of-contents.html>

This will FTP to the Oakland ftp server and retrieve the manual to Eudora 1.44:

<ftp://oak.oakland.edu/simtel/win3/eudora/eudrm144.zip>

Gopher to Creative Lab's server:

<gopher://gopher.creaf.com/>

Goes to the UseNet newsgroup rec.humor.funny:

<news:rec.humor.funny>

Ask for a tax break!

<mailto:pres@whitehouse.gov>

# The HTTP URL

This refers to HyperText Transport Protocol. Which is a fancy way of saying "visit the web page, stupid!".  
http always has one of these two forms:

`http://SERVER.NAME`

`http://SERVER.NAME/FILEPATH`

The "server name" is the complete name of the server. For example, **www.microsoft.com** or **galadriel.ecaetc.ohio-state.edu** (say that with a mouthful of cookies). The server name may be OPTIONALLY followed by a colon and a number - this refers to the "port" that the target system is listening for a connection on. You usually don't need to add this "port" option - most systems are set to use the default port value of 80. If you think your server is configured differently, contact your system administrator.

"File path" refers to how to get to the file or directory. This depends on where the data is set up. If you are running your own personal page out of your user directory, you will probably use:

`~joe/file.html`

"Joe" is you, "file.html" is the filename. The last part of the filename is important. Like DOS, Web Servers use the dot-name to figure out what the file really is! You usually want to keep it ".html" unless you have special requirements. However, since each server is configured differently due to the system, you should ask your system administrator about personal web pages, and how relative links work.



# The FTP URL

File Transfer Protocol. I would prefer is this was called "leech". It's just a way to retrieve a file from the sever to your machine. The format of FTP is this:

`ftp://SERVER.NAME/FILEPATH`

The "server name" is the complete name of the server. For example. **ftp.microsoft.com** or **oak.oakland.edu**. The server name may be OPTIONALLY followed by a colon and a number - this refers to the "port" that the target system is listening for a connection on. You usually don't need to add this "port" option - most systems are set to use the default port value of 23. However, sometimes the server may be running in a non-standard configuration, so you should contact your system administrator if you have problems.

"File path" refers to how to get to the file or directory. This depends on where the file is stored from the FTP root directory (usually PUB for public). For example, on Oakland, you could use `simtel/win3/network` to get into the network directory, or `simtel/win3/graphics/00index.txt` to retrieve the index listing.

## The GOPHER URL

Gopher is a yet another way to retrieve information. It arrived just before World Wide Web hit the scene, so it never really took off to the point that a lot of systems are running it. Gopher is sort of between FTP and HTTP in the way it handles information. Like FTP, Gopher presents information as files and directories. Like HTTP, it can refer to information that is elsewhere on the system or not even on the same network (pointing to a file in England from a system in Ohio, for example).

Using Gopher commands is just like FTP or HTTPD:

```
gopher://SERVER.NAME/FILEPATH
```

The "server name" is the complete name of the server.

"File path" refers to how to get to the file or directory.

## The NEWS URL

This is UseNet news. In order for this to work correctly, the USER must have news access and be able to read it over the World Wide Web; you do not. For example, Galadriel doesn't have a news feed (too much bandwidth and we get it elsewhere), but I can refer information to alt.rec.foods all I want.

The format of news is:

news:NEWSGROUP

Where NEWSGROUP is a full newsgroup name. Just copy this from the list that you are using.

## The MAILTO URL

The mailto tag is very limited. All it does is tell the browser software the user is using to start a mail message to the person specified in the mailto tag. There is no facility in the current HTML specification to supply subject or default text.

The format of mailto is:

mailto:USER@SYSTEM.NAME

Where USER is the user name (duh) and SYSTEM.NAME is the name of the system where the user has the mail account. You can usually copy-and-paste email addresses from newsgroup posts or email letters in here.

Since it is perfectly legal to give the mailto with only a user name, beware: if you do so, the system will use the SYSTEM.NAME of where the *user* is from, not where *you* are from. So if you do this:

mailto:admin

The hapless user will cheerfully send a letter to *his* admin asking why the web page is broken and where are all the dirty pictures?

**NCSA:**

The National Center for Supercomputing Applications (NCSA), located at the University of Illinois at Urbana-Champaign, is a high-performance computing and communications facility and research center designed to serve the U.S. computational science and engineering community.

**CERN:**

European Laboratory for Particle Physics, located near Geneva in Switzerland and France.  
CERN is the birthplace of the World-Wide Web.

**CSIM:**

Contraction of the phrase Client Side Image Maps. CSIMs are only usable under browsers that support HTML 3.0. CSIMs do not require any sort of special program on the server.

# Setting up the Server

*This topic only applies if you are going to use server-processed image maps (NCSA or CERN). If you are using CSIMs and have no intention of support HTML 2.0, then skip this.*

Creating the image map is only part of the job - the other part is getting the server to recognize your map and use it in the manner you expect.

It would be nice if I could just say "Put this line in your page and all will be fine", but I can't. There are at least several thousand servers running out there, and all have been configured differently.

Some general tips:

- First: Do **not** assume you can run image maps. Some systems put restrictions on maps for whatever reason. Some systems forbid image maps altogether. Just because you have a web page doesn't mean it can do image maps.
- Second: What works for one system may not work for the next one. Some server software requires the image maps to be processed through a special program. Others require the name of the image map file to end with ".MAP" (this is fairly common). Most of the more recent servers have the image map program built into the software, so you don't need to do anything special.
- Third: When in doubt, email your system administrator. He or she should be able to give you the information or point you to it.
- Fourth: Do **NOT** post messages to the newsgroups asking for help *UNTIL* you try your system admin. We don't run your system and have no clue as to what your software is. Even I cannot help you very much in this regard.



# Map This Toolbar

Map This, by design, is loaded with features and tools to make creating and maintaining image maps pretty simple. Using these tools is very straight-forward matter.

There is really only one main toolbar in use during your map editing session. This toolbar appears with each open map, and is the standard "floating" Windows toolbar. You can "grab" it with the mouse and "tear it off" or reposition it along any parts of the window.



Other related subjects: [New Image Map](#); [Right-Click](#); [Guides](#); [Grid Settings](#); [Map Info](#); [Area List](#); [Edit Area Information](#); [Using the drawing tools](#).

The rectangle tool lets you draw simple rectangle areas on the map.

The circle tool lets you draw circular or oval areas on the map.

The polygon tool allows you to draw free-form areas. These areas can contain up to 64 points.

The arrow tool lets you move areas and enables the right-click on the areas.

Select Zoom In to "get closer" to the map and the graphic. You can go up to 8:1 resolution.

Zoom Out "moves you away" from the map, until you are at the normal view.

Click the Save button or press control-S to save your work. Same thing as selecting File->Save.



The Delete tool removes the currently selected area. Once it is gone, it is GONE.

Select the Grid to turn the current grid on or off. Select Goodies->[Grid Settings](#) to define the grid.

The [Map Info](#) item lets you change the map information and settings.

This tool toggles the [Area List](#) dialog on and off.

Allows you to edit the currently selected [Area's Information](#). Same as double-clicking with the arrow tool on an area.

Allows you to [view the map](#) in a fashion similar to what a user on the Web might see.

## Using the drawing tools

Map This has three drawing tool and one general purpose editing tool for you to use. These tools are shown as different cursors on the screen.

The Rectangle tool is used to create rectangles in the map. You create new rectangles by moving the cursor to where you want one corner to be at, pressing **and holding** the left mouse button, and dragging the cursor to the opposite corner. Release the mouse button when you are done. A rectangle "ghost" is drawn on the screen in real-time as you go.

The Circle or Oval tool is used to create round areas in the map. It is used just the same as the rectangle tool.

The Polygon tool is used to create non-uniform areas. This can be used to react to something that is shaped like a boat for example. You start a new polygon area by moving the mouse to the first point, and **clicking** the left mouse button **once**. Move the cursor to the next point - you will see a "ghost" line tracking from the starting point to where the cursor is. Click the left mouse button **once** to add this point into the polygon and start the next line. You end your polygon creation by either **double-clicking** the left mouse button where you want the last point to be, or bring the cursor very close to the starting point and clicking **once**.

All three tools can be used to reshape their areas by moving the cursor over one of the "handles" (the gray or black squares on the outline of the area) and pressing **and holding** the left mouse button. Moving the mouse will then move that handle in the appropriate way and reshape the area. Release the mouse button when finished.

The Arrow tool is used to move areas in addition to allowing you to change their size and shape. To move the area, position the cursor to where it changes into a 4-way arrow. Press **and hold** the left mouse button and move the mouse to where you want the area to be moved to. Release the mouse button to "drop" the area in its new place.

The Arrow tool can also be used just like the other three tools to reshape the areas. Just grab the area's handle, and press and hold to move.

The Arrow tool also has a [right-click menu](#) when you move the cursor over an area. This area's outline changes color to indicate that this area is being looked at - this is called a "flyby". [Right-click](#) to bring up a popup menu that lets you edit this area's information, delete it, or rearrange it in the map list.

The Polygon and Arrow tool also allow you to [right-click](#) when on a polygon line or point to add new points or delete old points from that polygon.

When you are creating a new area, resizing or reshaping an old area, or moving an area, you can cancel your movement by either clicking the right mouse button or pressing the Escape (ESC) key on the keyboard. If you are holding the left mouse button down, you may have to keep holding it down and click the right mouse button at the same time. Some mice do not support this, and for some people, this is uncomfortable or physically impossible. In this case, press the Escape key on the keyboard to cancel the change or creation.

Also, Map This enforces certain minimum limits on new areas. If you try to create an area that is very small (around 4 pixels tall or wide), it will silently remove it. It does this to keep you from accidentally creating lots of little areas that you can't seem to manipulate.

# Keyboard controls

Sometimes it's easier and faster to work with the areas with the keyboard. This is especially true when you are putting in a lot of URLs all at once. Map This has some keyboard control that allows you to edit things a little easier and quicker.

**Left arrow key:**

Takes you to the previous area in the list. If you are at the first area, takes to you to the last one.

**Right arrow key or Tab key:**

Takes you to the next area in the list. If you are at the last area, brings you back to the first one.

**Return key or Spacebar:**

Brings up the [Edit Area Information](#) dialog box. Same as double-clicking the area.

**Delete key or Backspace key:**

Deletes the area. You will *not* be prompted.



## Creating a new image map

Each image map must have an image to be based upon. It would be pretty strange to have a map that consists of nothing but a black screen!

Map This supports both GIF and JPG formats for images. However, not all browsers support JPG as image maps.

When Map This saves your image map file, it places a [special tag](#) into the file that records the filename of the image, and where it was last stored at. This gives Map This the ability to find the image when you re-open the map later.

Once the image is loaded and decompressed (this is very quick, even with JPG images), the document window opens up, showing the image and the [toolbar](#). Initially, the arrow tool is selected. Click any of the other [drawing tools](#), or use the [Guides](#) feature to make some quick areas up.

## Opening an existing image map

Map This will allow you to open maps that you have created, and previously existing maps that were designed for the [CERN](#) or [NCSA](#) servers, or for HTML 3.0 browsers in [CSIM](#) format. The Map This distribution archive contains both CERN, NCSA, and CSIM example maps and images (Fish for NCSA, Dragons for CERN, Lake for CSIM).

If you open a map that Map This did not create, then you will get a message box asking you to locate the image that goes along with the map. If Map This did save the map file that you are opening, there are [special tags](#) buried in the map file to help Map This locate the image file.

## Map file not created by Map This

When Map This writes out the image map file, it includes some special [comment lines](#) in the map's data. These special lines define what image file was used and where it was last located. If you are importing an existing image map file, these lines are obviously not there, so Map This has no way of knowing which image goes with what map. It's your job to point Map This at the right file.

[CSIM](#) "files" are usually stored in the HTML file that uses them, and Map This can usually locate the correct image reference for that map, saving you this step. Not always, though!

If you point Map This at the wrong file, well, you'll have to close the map and re-open it to get the right one. Map This can't tell if its got the right image or not.

## Unable to locate Image

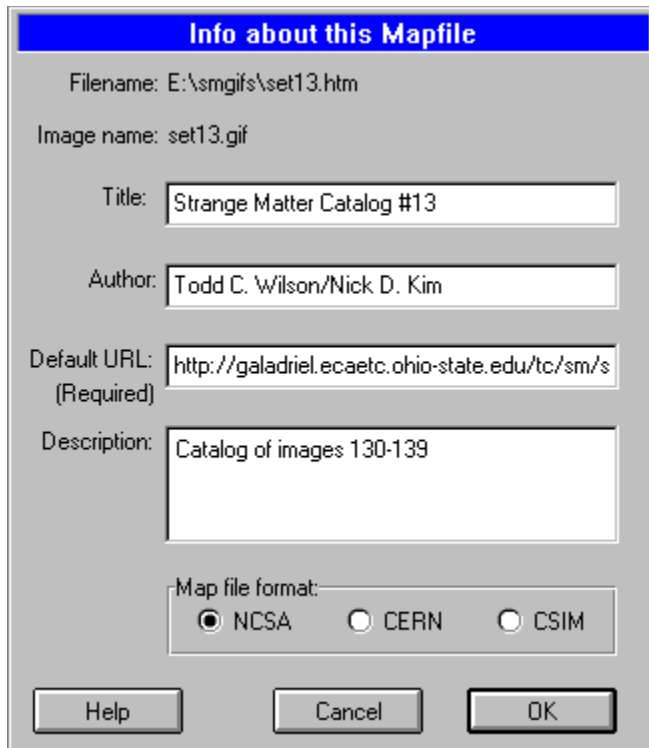
Map This maintains two special [comment lines](#) in the image map file when it writes it out to the disk: the filename of the image, and where the image was at when the map was saved. If the image or the map is moved on the disk (say, from one directory to another), Map This will try to look in the directory where the map is currently at, and where the image was last known to be at. If both of these fail, Map This gives up and asks you to find the image for it. If you manage to give it the wrong image, you will have to close the map and re-open it. The new location of the image file is not recorded unless you re-save the image map file.

## Editing the image map's general information

For each image map file, there is only one required line: the default [URL](#) to execute if the user clicks outside of any defined area. You can think of this as the "background" [URL](#), if you wish.

The other pieces of information - Title, Author, Description - are all optional, and have nothing to do with the actual function of the image map.

You can click on some areas of this graphic for more information:



The image shows a dialog box titled "Info about this Mapfile" with a blue header bar. The dialog contains the following fields and options:

- Filename: E:\smgifs\set13.htm
- Image name: set13.gif
- Title: Strange Matter Catalog #13
- Author: Todd C. Wilson/Nick D. Kim
- Default URL: <http://galadriel.ecaetc.ohio-state.edu/tc/sm/s> (Required)
- Description: Catalog of images 130-139
- Map file format:  NCSA  CERN  CSIM

At the bottom of the dialog are three buttons: Help, Cancel, and OK.

This is the filename of the map file. If nothing is here, then the map is brand-new and has never been saved. This is for informational purposes only.

This is the filename of the image file. This is for informational purposes only.

Title of the mapfile. This field is not required by the Web Server, but *is* used by CSIM files to name the map.



Author. Your name here. Optional.

Description of the map file. This field is optional, and can be used to describe what you are doing with this map file. Can be several lines long. The Web Server will not use this field.

This is the default [URL](#). Required by most Web Servers. Not required for CSIM files. You must give it the complete [URL](#) to whatever resource you wish to activate when the user clicks outside of any defined area.

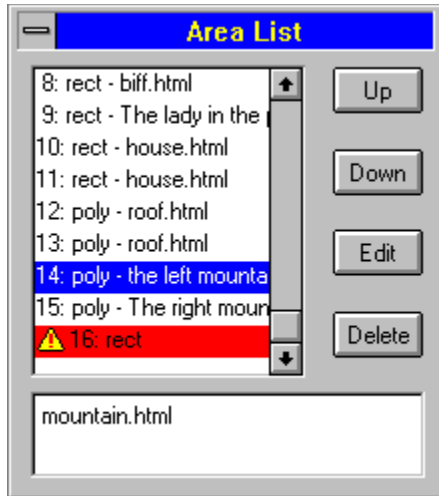
Image map format. [NCSA](#), [CERN](#), and [CSIM](#) are all supported.

Click this to cancel your input.

Click this to accept the data.

## The Area List Box

You can have Map This bring up a floating window of your areas to get an idea of what each one is. This List Box allows you to re-order the areas, delete them, and edit their information. It also shows which areas are missing URL's by placing a warning icon next to each area.



Brings up the [Area Info](#) dialog box for the currently selected area.



You can grab this to drag the window around.

Standard system menu. Double-click or choose "close" to close this window.

Shows the [URL](#) link for this area.

List of the areas.

Currently selected area.

Areas with the warning icon or highlighted in red are missing their URLs.


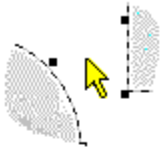










# The Right-Click Menu

The right mouse button is the "context menu" button. You click it to bring up a floating or "popup" menu. This menu changes depending on the tool and where you have clicked.

If you have the polygon or the arrow tool selected, clicking on a polygon line will give you the option to insert a new point. Clicking on a polygon's point will give you the option to remove that point.

Right-clicking inside of any area will bring up a four-item menu that allows you to edit the area information, delete the area, or move it up or down in the list.

Right-clicking outside of all the areas (in a part of the map that has no areas) will allow you to change tools, zoom in or out, edit the map information and access the other menu/toolbar items.

Tool in use	How it looks	Right-click menu
		<div data-bbox="794 667 1027 888"> <p>Map Info... Tools Zoom Grid Grid Setup... Guides...</p> </div> <div data-bbox="1044 741 1154 814"> <p>In Out</p> </div> <div data-bbox="1182 709 1369 856"> <p>Rectangle Circle Polygon • Arrow</p> </div>
		<div data-bbox="794 919 1027 1066"> <p>Edit Area Info Delete Area Move Up Move Down</p> </div>
		<div data-bbox="794 1104 1027 1140"> <p>Insert Point</p> </div>
		<div data-bbox="794 1178 1027 1213"> <p>Delete Point</p> </div>
		<div data-bbox="794 1251 1027 1430"> <p>Insert Point Edit Area Info Delete Area Move Up Move Down</p> </div>
		<div data-bbox="794 1461 1027 1640"> <p>Delete Point Edit Area Info Delete Area Move Up Move Down</p> </div>

This is a sub-menu allowing you to change the tool you are using. The currently selected tool (in this case, the Arrow tool) is indicated with a dot in front of the name. See also [Rectangle](#) tool, [Circle](#) tool, [Polygon](#) tool, [Arrow](#) tool.



Inserting a point in a polygon place another point on the line. You can then move this point to change or enhance the shape of the polygon.

Deleting a point from a polygon removes that point and causes the lines connection the point to the rest of the polygon to "snap back" to the other two points.

Placing the polygon or arrow cursor on or very close to a *polygon* line will let you make a new point on that line.

Moving the polygon or arrow cursor onto a polygon point will let you delete that point from the polygon.

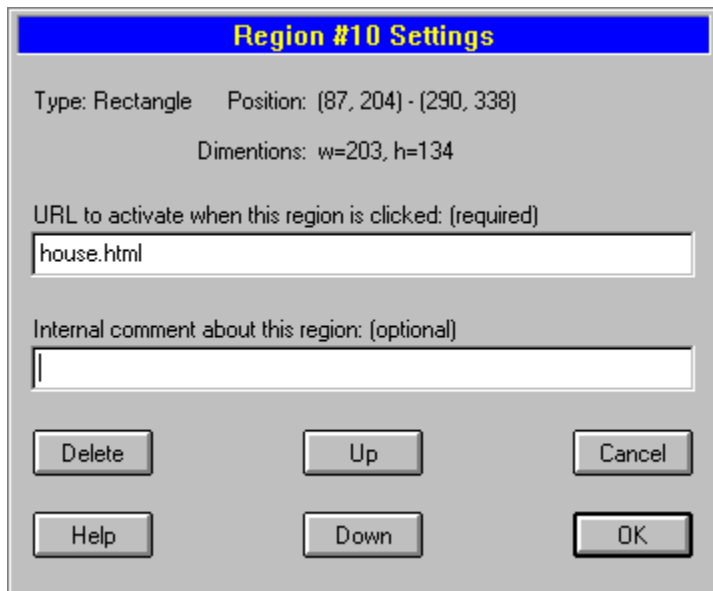
Placing the cursor (any type) between the areas gives you access to this popup menu.

When the arrow tool is in use and you place the cursor on an area, the area's outline changes color and the cursor becomes a 4-way arrow.

## Editing Area Data

When the arrow tool is selected, you can (a) double-click on an area; (b) click once and select the pencil tool; or (c) right-click to show a pop-up menu. Any of these actions will bring up the Edit Area Info dialog box. This dialog shows information about the area, such as the type, position, and size, and allows you to enter the [URL](#) for it, and the optional comment.

Click on one of the hot spots in the image below for some more information:



The image shows a dialog box titled "Region #10 Settings". It displays the following information:

- Type: Rectangle
- Position: (87, 204) - (290, 338)
- Dimensions: w=203, h=134
- URL to activate when this region is clicked: (required)  
house.html
- Internal comment about this region: (optional)  
[Empty text field]

At the bottom of the dialog box, there are six buttons arranged in two rows:

- Row 1: Delete, Up, Cancel
- Row 2: Help, Down, OK

This is the [URL](#) that is to be used when the user clicks on this area. Required by the Web Server. You must give it the complete [URL](#) to whatever resource you wish to activate.



Comment about this area. This comment is usually never presented to the user, and the Web Server will ignore it. Can be up to 200 characters long.

This shows what type of area this area is. Rectangle, circle, or polygon. The type of area defines how the server and Map This use the data that is attached to this area. The position and size also reflect the type. This is for informational purposes only.

This is the position of the area. For rectangles, this is the upper-left to the bottom right. For circles, this is the center point. For polygons, this is the smallest rectangle that covers the entire area. This is for informational purposes only.

This is the dimensions of the area. This depends upon the area type. For rectangles, this is the width and height. For circles, this the radius along both the X and Y axis's. For polygons, this is the total number of points. This is for informational purposes only.

Click this to accept the changes.

[Click this to reject the changes](#)

Click this to delete the area from the list.

Click this to move the area up in the list. Areas higher (lower numbered) are checked first.

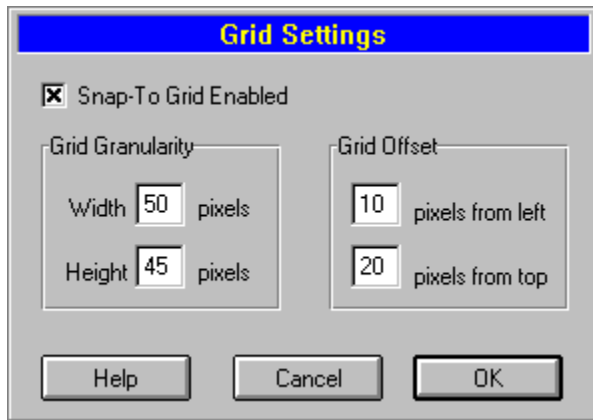


Click this to move the area down in the list. Areas lower (higher numbered) are checked last.

# Snap-To Grids

Grids are great tools in getting things lined up. Enabling the Grid causes Map This to force all your mouse input to whatever "granularity" that you specify. For example, if you tell Map This that you want a grid that is space 30 pixels across and 10 pixels down, then you will only be able to place points at 0, 30, 60, 90 etc. across the X-axis, and 0, 10, 20, 30, etc. down the Y-axis.

Map This also has a very interesting feature of specify the starting position for the grid. This means that you can have the grid at 5, 35, 65, 95 across and 1, 11, 21, 31 down if you wish.



This box turns the Grid on or off. This is the same as selecting the Toolbar button or the menu item.

This sets the width of Grid granularity. For example, if you enter 50 here, then the mouse position is constrained to 0, 50, 100, 150, etc., across the X-axis.

This sets the height Grid granularity. If you set this to 25, then the mouse position is constrained to 0, 25, 50, 75, etc., up and down the Y-axis.

This sets the starting left-side of the Grid. Using this allows you to set where the Grid starts at. If you set this to 3, and the width to 10, then the Grid is 3, 13, 23, 33, etc.

This sets the starting top side of the Grid. Using this allows you to set where the Grid starts at. If you set this to 2, and the height is 15, then the Grid is 2, 17, 32, 47, etc.

Click this to accept the changes. Your settings will be put into effect and saved in either your User Registry (Win95 or NT) or to the INI file (Win16) for later use.



[Click this to reject the changes.](#)

## Creating and using Guides

Quite often, I find that people (myself include) want to make a clickable map based on thumbnail images. While you can do this using the standard "make a rectangle" procedure, it's tiresome. And the results are not very uniform.

This is where the Guides option comes in. This dialog box allows you enter in information to tell Map This to create a series of rectangles, starting at a certain point, being so tall and so wide, so many across and down, and oh yes, space them this far apart.

The Guide box will not allow you to create Guides if the Guides will appear outside of the image, or you attempt to create more than 1024 of them.

One handy feature of the Guide dialog box is to define the Base URL for each of these new areas. This Base URL can be just the common part leading up to whatever, or you can do one very interesting thing: put the special "%d" character sequence in the line to have the program put the area number in each of the URL's that it creates.

If you have a Base URL like this:

`http://system.com/images/graphic%d.gif`

Then for each new area, the %d will be replaced with the area number. For example, area 1 will have:

`http://system.com/images/graphic1.gif`

And area 42 will have:

`http://system.com/images/graphic42.gif`

You are only allowed to have one %d in the entire Base URL. Yes, it is based on the ANSI C "printf" function, and no, you only get one, and only a number. Sorry.

(If you try to put something like %s or %dbob%d, Map This will assume that you meant to do that and will NOT put in the area numbers).

Image dimensions: 640 x 480

Resulting Guide Bounds: 10,5 to 310,75 (10 areas)

Width  Left Start at

Height  Top Start at

Horz Offset  Across

Vert Offset  Down

Base URL:

This displays the size of the image that you are working with. It is shown for informational purposes only.

This shows the area that will be "covered" by the Guide boxes. If this goes to red (or the red-compliment of your window text color), then you will not be able to create the Guides.

This is the width of each Guide box.

This is the height of each Guide box.

This is the left starting point for the Guides. You can use this to start the Guides away from the left side of the image.

This is the top starting point for the Guides. You can use this to start the Guides away from the top of the image.



This is how much additional space to put between the Guides as they go left-to-right across the image.

This is how much additional space to put between the Guides as they go down the image.

This is how many Guides to put across the image.

How many Guides to put down the image.

Click this to accept your values and create the Guides. Your settings will be saved so you can use them next time.

[Click this to not create any Guides.](#)

If you wish, you can put in a Base URL that will be put in each new Guide. To have the Guide generator put the area number in this, simply place **one** %d character sequence in the string.

## Testing the Map

Once you have your areas defined, how do you check and make sure everything will work correctly? The absolutely best way is to transfer the needed files to your Web Server and test things from there. However, you may want to just see what the map would look like without the areas getting in the way. Selecting the Test Map item will bring up a dialog box informing you about the limitations of the test, and then proceed to actually viewing the image, full-screen. This is similar to a Print Preview in word processing programs.

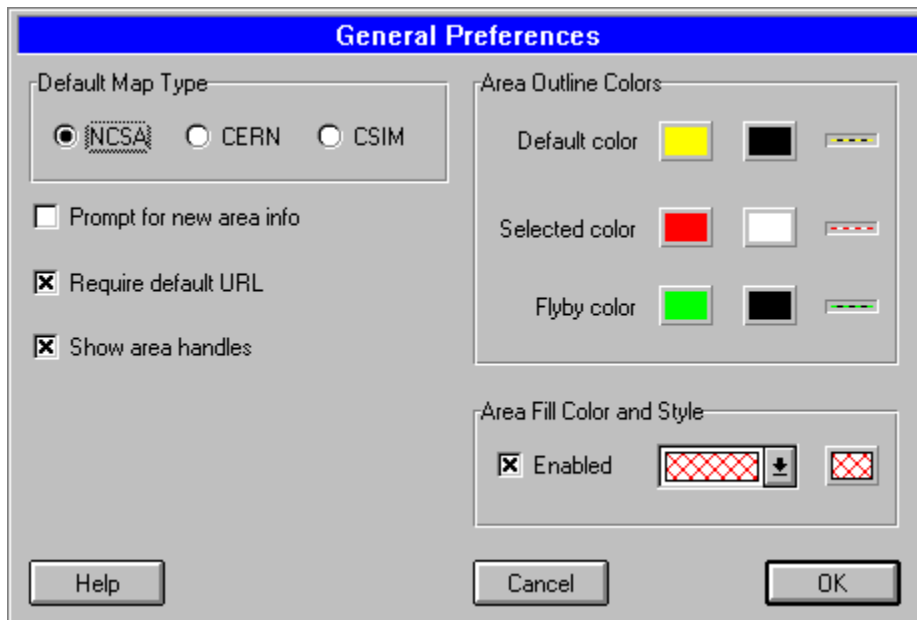
You must still transfer the map to the server in order to get it to work, and then set the server up correctly if you need to. See [setting up the server](#) for more information.



# Preferences

The Preferences dialog allows you to change the behavior and appearance of Map This. In this dialog, you can set the default map type for new image maps, whether or not to prompt for area information when you create a new area, if the default [URL](#) is really needed or not, and the colors of the outlines of the areas.

For more information, click on the areas below



Default map type for new image maps. You can change this per each map in the Map Info dialog.

If this is set, then Map This will automatically bring up the Area Info dialog each time you create a new area.

Most servers require the default URL to be specified. If yours doesn't, and you don't care if you have it or not, clear this box.

Select this to cancel your changes.

Select this to save your changes.

The default color is what is used to draw the area outlines when they are not selected nor being highlighted during a "flyby".

The selected color is what is used to draw the area outline when it is the selected area. Only used when the arrow tool is selected.



The flyby color is used to highlight an area when you move the arrow tool over an area.

This is what the line will look like when it is used.

Allows you to turn the handles of the areas on or off. Doing so may make it harder to change the shapes of areas, but will allow you to deal with close-fitting areas better.

Some NCSA servers disallow abuse of the circle format to create ovals. Set this to keep these circles as "true" circles, and not allow ovals.

If this set, then the areas of the map will be filled in using the pattern and color to the right.

This is the style of the fill-in for the areas if the checkbox is set. You can choose from several different patterns.

This is the color and a sample of the pattern used to fill in the area. Click this to change the color.

## Opening an Image File (GIF/JPG)

Since Image Maps require an Image to place the Map on, you need to select a suitable graphic for the "background" of the map. Map This supports both GIF and JPG. There are some considerations to be aware of:

GIF is the normal graphics type used for image maps. It is decoded quickly by the browser, and all browsers support it.

JPG allows more colors and gives slightly better (on average) compression with some loss of quality. However, not all browsers support this format for image maps. This format is defined as being "officially" supported by HTML 3.0 specs.



## Opening an Image Map File

Map This supports all three major types of image map files: [NCSA](#), [CERN](#), and [CSIM](#) /HTML.

NCSA and CERN maps must be stored in files that are "external" to the parent HTML file. However, Client Side Image Maps (CSIMs) are usually stored within the parent HTML file (they can, however, also be stored external if you so desire).

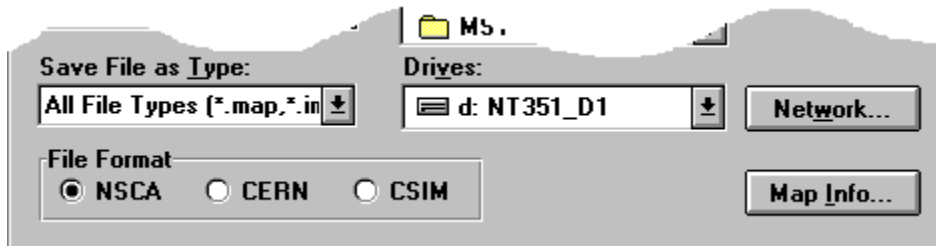
Map This can open any HTML file and look into it for an image map. If more than one map is stored within the HTML file, it will bring up a [dialog box](#) asking you which one to use.

As a standard convention, NCSA and CERN image maps are stored as filenames ending in ".map". An older, less-supported convention for NCSA maps is ".imp".

CSIMs are always stored as either ".htm" or ".html" - Map This treats both the same.

## Saving an Image Map File

If you are saving a newly created map, or saving an old map under a new name by Save As, Map This shows you the standard "file save" dialog - with some added features. Where these features are at depends on if you are using Windows 95 or not; however, they both look very similar to this:



The radio buttons determine which format you want to save the map as. The Format Type buttons and the File Type drop-down box work together: you cannot choose a format that is incompatible with the file type, and vice-versa.

The [Map Info](#) button takes you right to the overall information about the map.

Saving the image map into an existing HTML file will take you to the [Save HTML As](#) dialog box.

If you have this button, you can access your Windows network drives.

Click this to bring the [Map Info](#) dialog.

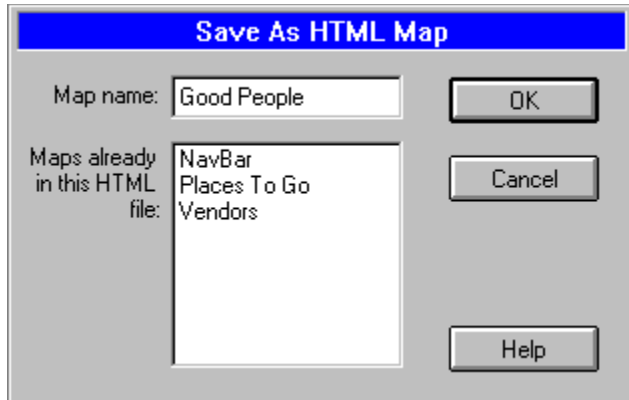
A drop-down list box containing the file types that Map This supports.

These radio buttons determine what kind of file you will be saving. Anything changed here is changed in the [Map Info](#) settings.

## Saving an Image Map into an existing HTML File

Client Side Image Maps allows you to keep more than one map in an HTML file. If you choose Save As... from the File menu, and you select an existing HTML file, Map This will prompt you for the name of the map to save "into" the file as.

Map This uses the title of the map as the name to be saved as when you are saving an HTML file.



This is the list of maps that are currently in this HTML file.



This is the name of the map to save at. This is the same as the Map Info's "Title" block. Changing it here will change it there, too.

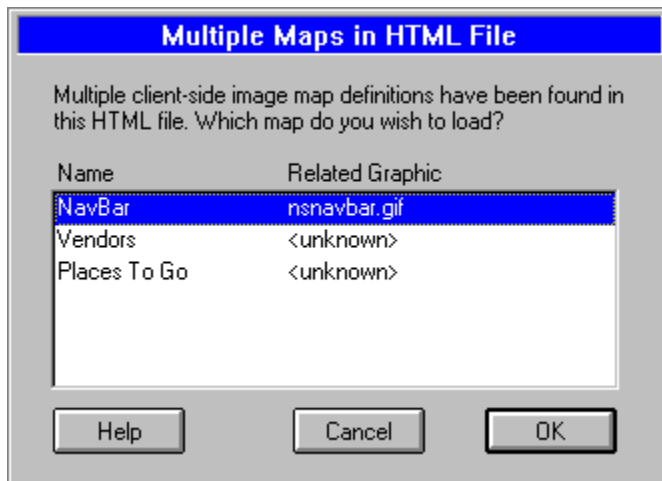
Select this to save the map.

Select this to cancel saving.

## Multiple Image Maps in a Single HTML File

One feature of Client Side Image Maps (HTML 3.0), is that they allow the possibility of putting more than one image map "file" into a single HTML file. The advantages of this is that you can place all of your image maps into one single file, and then access them as needed. The reason to use that method is that most servers are caching files (just like your own Windows machine), so it doesn't have to find un-cached files. And many browsers will cache the HTML file locally and not request it to be sent over the connection - another time savings!

When you open a HTML file that contains more than one image map definition in it, Map This will bring up a dialog asking you which map you want to load. Each map's name and associated graphic is shown (if the graphic is known). Just double-click the item, or click and select okay to load.



This map has a graphic reference that Map This was able to understand. You may still have to tell Map This where the graphic is really at.

These maps don't have a known graphic reference, so Map This will prompt you for the graphic.

Select this to load the selected map.

Select this to cancel the map load.



