


Searching the surf (Part 1 of 2)

How to find Web info more easily



The amount of information on the Net grows massively and daily. Tracking this growth is a mammoth undertaking; providing a window onto it, seemingly impossible. Yet the search engines—Alta Vista, Lycos, Excite, InfoSeek, and others—do exactly that, and millions of users visit those engines every day, attempting to surf the tidal wave of data.

As a result, these index sites (and similar review and overview sites, like Point Survey and Yahoo, respectively) generate much more



traffic than any content site could hope to achieve. It almost seems as if people have become obsessed more with metainformation—information about information, like indexes, contents, hotlists, and the like—than with information itself. Judging from public reports from search-engine sites, tens of millions of pages of search results are looked over every day—and a reasonable conclusion is that those pages drive people to billions of real content pages.

Complicating things even further, the search tools on the various engine sites have lately become increasingly sophisticated and exhaustive—that is, more pages on more sites are being indexed more frequently. There can never be complete closure because the churn of information is like motion on the ocean: some areas may be Sargassos of dead sites that float endlessly and unchangingly (we've all come across these), while others are heaving, hurricane-driven waves of information, updated dynamically or by the second.

A colleague of mine, Steve Broback, mentioned during a recent Adobe

Internet Conference that he never writes down URLs (Uniform Resource Locator Internet addresses) any more. Instead, when he gets to a computer, he types a brief description of what the person told him into the Alta Vista search engine, and the first or second match is generally what he's looking for.

This is both the power and the terror of the Net. You can find anything you can describe—as long as you can describe what you're looking for with enough detail to find it, and phrase it in such a way that a search engine can respond.

In this first part, I'll discuss making searches. In the second part, I'll show how it all comes together.

Refining searches

Typing in a few general words to make a raw (i.e., unrefined) search isn't much help unless you're already looking for very narrowly defined information, like beekeeping in Texas universities. You have to learn tricks to make a good search, either by defining the search tightly from the outset, or by refining it

progressively when you find too many matches.

All the major search engines have search languages or “advanced” search features that let you concatenate different terms. The primary terms are logical operators, like AND, OR, and NOT. Logical operators create a statement containing semantic items that you (or a program) can evaluate as a true or false statement. So if you say that you are searching for “bikini and atoll,” the search engines find only cases in which a page contains both words—if the page contains both words, the statement is “true.” The case is identical with OR or NOT; requesting “bikini and atoll or funicello” will retrieve all pages in which the statement is true that the page contains either the words bikini and atoll, or the page contains the word funicello.

You can further refine logical statements by understanding precedence and parentheses. Precedence is the order in which the statement is logically evaluated. OR is evaluated after AND, so, if you rewrote the above example as “funicello or bikini and atoll,” it still has the same meaning. Parentheses can be used either to change precedence—putting the OR before the AND—or for



clarity. So the first example might be read "(bikini and atoll) or funicello" and the second "funicello or (bikini and atoll)." In cases where sets of parentheses are "nested," the parentheses get read from the innermost level out when the system evaluates the order in which the statement should be interpreted. That is, instead of reading the whole search request in order from left to right, the software evaluates the statements in the innermost set of parentheses first, then moves up levels of matching parens until it has evaluated the entire statement. This allows you to group terms by meaning, forming more complex "thoughts." If sets of parentheses are side by side as opposed to being nested, it's not important which set gets evaluated first.

Let's look at some complex examples. Say you want to find pages about the Bikini Atoll but nothing about the associated swimwear:

```
(bikini and atoll) and not (swimwear or swimsuit or two-piece)
```

This search says, "Find only documents containing both the words bikini

and atoll, and not containing the words swimwear, swimsuit, or two-piece.” The ORs are in parentheses so that they get treated like a single phrase; without the parentheses, the search could be read as “find documents containing bikini and atoll and not swimwear; or, any documents containing swimsuit; or, any documents containing two-piece.”

To get an even better match, depending on the search engine, you might try using a word like NEAR or another such “proximity operator.” This helps you find sets of words near each other, resulting in better matches. At Alta Vista, NEAR means that the words, or sets of words, that you’re looking for must be within a few words of each other in the document. Thus your “bikini” search would be

```
(bikini near atoll) and not (swimwear or swimsuit or two-piece)
```

The first part now translates to “find only documents containing bikini and atoll within a few words of each other in the document.”

Finally, if you want the words to be one after another, in several engines you can use quotation marks. The following, for example, only finds documents where the word "atoll" follows "bikini."

`"bikini atoll" and not (swimwear or swimsuit or two-piece)`

Ego surfing

Wired magazine's Jargon Watch column defined ego surfing as using search engines to find references to yourself. It's fun and easy, and impresses your friends! The simplest thing is to go to Alta Vista and, under Advanced Search, type "glenn near fleishman." Oh, all right, substitute your own first and last name in there.

You'll find a number of unknown namesakes. My housemate discovered a namesake who's a graduate student in the apiary sciences (beekeeping) through this method. It's also a way to discover copyright infringement if someone is stupid enough to steal one of your pieces and then keep your



name on it. (Don't laugh; this happened to one of my pieces about HTML from *Adobe Magazine*, *adobe.mag's* print-based sister publication. Kids, give a hoot! Don't pirate authors' work.)

A more useful and less ego-intensive search involves the "link:" and "host:" functions on several search sites. To use Alta Vista to find all the links to your site that you didn't generate yourself, for instance, this query will do the trick (again, under Advanced Search):

<http://www.yoursite.com> and not *host:www.yoursite.com*

One of our clients generates 40,000 matches in this manner. Gratifying, to be sure.

Metacrawling

The more you learn about this, the more likely you are to ask, "Why can't I just search in the ultimate search index—a cross-connection and collation of all the search engines?" Well, Virginia, you can. Metacrawlers, as one site calls them, or agent-based client search programs are, in my humble opinion,

going to sweep the Net.

The granddaddy of such programs, Metacrawler (<http://www.metacrawler.washington.edu>), still lives at the University of Washington, where it was invented, though a company called Hotbots should be migrating it to a commercial site any day. Metacrawler takes your queries, rewrites them in the language of each of ten major search sites, sends the queries out while you wait, compiles the results, verifies (if you like) that the links are active, and summarizes the results, removing duplicates.

This is ideal for you, but less so for the companies that run the search sites. Metacrawler reduces the number of users coming to the search engines, thus reducing advertising income based on ad impressions, which is still the primary income source (outside of stock offerings) for these companies. And, since Metacrawler generates all the search requests from its Web site, locking it out would be easy for the search sites to do.

Similar in concept, and potentially more of a problem for the search sites, are local or client-based agents. These little programs, which may come in the

form of Java applets or full-fledged applications, send queries to the search engines from your own machine, which means they are impossible to block. In function, these local agents are identical to Metacrawler. (I've seen a couple of them in beta, and their speed and flexibility are remarkable.)

Getting to know you

This first part should answer the question of how to *get* where you want to go today. Using carefully constructed searches in the language of the search sites can help you zero in on the information you're looking for. In the next part, you'll learn what's happening behind the curtain while you're talking to the wizard.

