NAME

wwwseek − search multiple World-Wide Web sites in parallel

SYNOPSIS

wwwseek [**−v JOBS=***nn*] [**−v TIMEOUT=***nn*] [**−v TMPDIR=***dirname*] [**−v TRIES=***nn*]
pattern(s) >outfile

DESCRIPTION

Interactive Web searching in a browser client can be extremely tedious, since each search engine has a different user interface, and each displays a list, possibly spanning multiple screens, of hypertext links to Web documents that match the specified patterns. The user must then follow each link in turn, wait for the document text, and usually, associated images, to be downloaded, then invoke the primitive search facility provided by the Web browser to find the patterns sought.

wwwseek removes most of this tedium, produces results very much faster, and permits more powerful search mechanisms.

wwwseek queries several major Internet search engines for lists of Web documents containing user-specified patterns, then creates a temporary shell script that retrieves those Web documents *in parallel*, ignoring duplicates, and writes them to *stdout*.

The output file can later be searched in a text editor, or with pattern matching utilities like **agrep**(1) or **egrep**(1), and the searches can be repeated in the local output file, with variations, as often as needed, without having to retrieve documents from the Web again. The surrounding context options, **−A** *nnn*, **−B** *nnn*, and **−***nnn*, of the GNU implementation of **egrep**(1), or the *occur* command in GNU **emacs**(1), are particularly useful in reducing the amount of material to be looked at, and determining whether the match is useful or not.

If you find that HTML markup clutters your search output, obscuring the patterns that you are looking for, consider prefiltering it with a utility like **dehtml**(1) to remove the markup.

Because Web sites are frequently inaccessible, and potentially hundreds or thousands may be contacted by this program, the normal 15 min timeout used by **wget**(1) to contact a host to fetch a document is reduced to 15 sec. This can be changed by a command-line option.

Because **wwwseek** can sometimes take several minutes, or even hours, to run, it produces a progress report on *stderr* showing the current document number and uniform resource locator (URL) that it is fetching. Because the searches proceed in parallel at unpredictable speeds, the document numbers will often be somewhat out of order.

The output file begins with a three-line HTML comment recording the **wwwseek** command line, the current date and time, and the hostname on which **wwwseek** was run.

Each retrieved file is copied to a separate page of the output file, beginning with an ASCII formfeed (Ctl-L) character, and followed by a distinctive HTML comment of the form

<!-- wwwseek URL="..." -->

to record the origin of each document. This is convenient if you later wish to return to that site, perhaps to find other related documents.

HTML comments are preserved by **dehtml**(1), so you can still identify document origins even when **dehtml** has been used to remove HTML markup.

OPTIONS

Command-line options must precede the search patterns, and option letter case is *significant*.

−v JOBS=*nn* Set the number of parallel jobs retrieving Web documents. The default is 25, and any value outside the range 1 . . . 25 will be reset to the default. Each subprocess requires about 2MB of memory, plus space in */tmp* to hold the retrieved file. If either of these proves to be a limiting resource on small systems, then a smaller **JOBS** value can be specified on the command line. However, since the work is largely network-I/O-bound, the elapsed time for a single search is expected to be sped up by as much as a factor of **JOBS**, so large values are desirable.

−v TIMEOUT=*nn* Specify the maximum number of seconds to wait for a read request: the default is 15.

−v TMPDIR=*dirname*
Specify an alternate directory for the temporary shell script and output files from parallel retrievals. The default is */tmp*.

This variable may also be specified in the environment, as is conventional practice in the UNIX world.

−v TRIES=nn
Specify the maximum number of retries made to contact a remote host; the default is 4.

SEARCH PATTERNS

The several seach engines invoked by **wwwseek** lack a common advanced search string specification, although all treat a list of one or more words as if the words were interspersed with Boolean OR operators, so that any Web document whose contents match one or more of the words will be returned in the results list.

Alternatively, the words can be prefixed with plus signs to indicate that they are required to be found; this effectively turns the implicit OR operators into AND operators.

Search patterns should usually be entered in lowercase letters, which all engines interpret to mean matching without regard to lettercase. Uppercase letters in patterns generally request exact matching.

Some, but not all, search engines recognize a terminal asterisk in a pattern to mean zero or more following characters, so the pattern *box** would match *box*, *boxcar*, *boxed*, *boxes*, *boxing*, *boxwood*, *boxy*, . . .

A few engines recognize *altavista* advanced search strings, e.g.,

arg1 NEAR arg2
arg1 '~' arg2

arg1 AND arg2
arg1 '&' arg2

arg1 OR arg2
arg1 '|' arg2

arg1 AND NOT arg2
arg1 '&' '!' arg2

arg1 'AND' arg2 'AND' arg3
arg1 '&' arg2 '&' arg3

arg1 OR arg2 OR arg3
arg1 '|' arg2 '|' arg3

Clearly, the named Boolean operators are more convenient than the single-character ones, which need to be protected by shell quotes.

The safest common syntax is one or more words or quoted strings, each prefixed with a plus, meaning all must be found:

+arg1 +"arg 2" +arg3 +"arg 4 with more blanks"

Parenthesized expressions in search strings are not yet handled by **wwwseek**, or recognized by more than a few search engines. They can be passed through by separating them with plus signs: the *altavista* advanced search string

arg1 AND NOT (arg2 OR arg3)

can be encoded as

arg1 AND NOT '(+arg2+OR+arg3+)'

FILES

These temporary files are created in the directory defined by **TMPDIR**:

*ws*nnnnnn.*sh* Temporary shell script; *nnnnnn* is a random number intended to make name collisions with other simultaneous uses of **wwwseek** unlikely.

*ws*nnnnnn.*tm*.kk Temporary output from a parallel document retrieval; *kk* is a sequence number from 00 to **JOBS**- 1.

Unless **wwwseek** is unexpectedly terminated, these temporary files are normally deleted on completion.

ENVIRONMENT VARIABLES

TMPDIR

Directory where temporary files are to be created (default: */tmp*).

SEE ALSO

agrep(1), **amaya**(1), **arena**(1), **chimera**(1), **dehtml**(1), **egrep**(1), **emacs**(1) **grail**(1), **hotjava**(1), **ie**(1), **lynx**(1), **netscape**(1), **wget**(1), **xmosaic**(1).

AUTHOR

Nelson H. F. Beebe, Ph.D.
Center for Scientific Computing
University of Utah
Department of Mathematics, 322 INSCC
155 S 1400 E RM 233
Salt Lake City, UT 84112-0090
USA
Tel: +1 801 581 5254
FAX: +1 801 585 1640, +1 801 581 4148
Email: <beebe@math.utah.edu>
WWW URL: http://www.math.utah.edu/~beebe