wwwwais.tiff) wwwwais.c, version 2.5

Contents

€What is wwwwais?€

WWWWAIS is a small ANSI C program that acts as gateway between programs that create indexed catalogs of files and a forms-capable World-Wide Web browser. With the freely distributable **freeWAIS** or **SWISH** packages, this program, and your local Web site, you can:

- Create searchable databases of the information on your Web site
- Allow users to search multiple databases via their Web browser with customizable options
- Create a custom pop-up menu of servers to search through
- Produce hypertext search results, with file information and links directly to the relevant HTML documents
- Retrieve WAIS source descriptions and files
- Specify URLs and filters to map results to
- Only allow users from certain sites to search documents

For an example of what the program can do, try €searching for the words "office and map" at EIT€, with the results sorted by title.

This program is very loosely based on the Perl waisq interface program that comes with NCSA's httpd.

€Great! How do I get started?€

First, you need the following software:

freeWAIS or other WAIS software

- €freeWAIS 2.02€ or greater, available from €CNIDR€ (Clearinghouse for Networked Information Discovery and Retrieval)

  Note that versions of freeWAIS other than 2.02, including the commercial WAIS software, may or may not work for what you want to do. Note that all WAIS-related software has strange quirks and bugs - you may have to experiment a good deal...

  - freeWAIS is a freely available software package that makes use of WAIS (Wide-Area Information Servers), a distributed document retrieval system. See the €Dictionary of Computing WAIS definition€, or read the €comp.infosystems.wais€ €Frequently-Asked Questions file€.

- A forms-capable Web browser, such as €NCSA Mosaic for X€ 2.0 or greater or €NCSA WinMosaic€ (Mosaic for Windows) 2.0 or greater.

Once you get freeWAIS, you'll need to compile it and set it up. The following documentation may help to give you a better understanding of what you're going to do next.

· **The €Mosaic And WAIS Tutorial€**, and
· **€WAIS and HTTP Integration€**, by Marc Andreessen.
· A €page of Z39.50 resources€ (Z39.50 is a protocol that WAIS uses).

If you're only using freeWAIS to do Web searching, you may want to put it all somewhere such as `/usr/local/httpd/wais` (but that's just my preference). You'll also want to create a directory to hold the WAIS database for your site, somewhere like `/usr/local/httpd/wais/sources`.

SWISH

You can also use SWISH (Simple Web Indexing System for Humans), a program that does many things similar to WAIS but is highly simplified, so it's easy to set up and install.

· You can read the SWISH documentation at `€http://www.eit.com/software/swish/swish.html€`. The source code and related files can be downloaded from `€http://www.eit.com/software/swish/€`.

You may want to put SWISH things somewhere such as `/usr/local/httpd/swish`. You'll also want to create a directory to hold SWISH databases, somewhere like `/usr/local/httpd/swish/sources`.

€What to do after compiling€

After you've compiled (and installed) freeWAIS and/or SWISH, make sure the **waisindex**, **waisq**, **waissearch**, and **swish** programs are somewhere in your executable path (somewhere such as `/usr/local/bin`).

Now you'll want to put the following C-shell script somewhere where you can run it. This script will index your Web site into a searchable database that WAIS clients can read.

```
#! /bin/csh

set rootdir = /usr/local/www
#       This is the root directory of the Web tree you want to index.

set index = /usr/local/httpd/wais/sources/index
#       This is the name your WAIS indexes will be built under.
#       Index files will be called index.* in the /usr/local/httpd/wais/sources
#       directory, in this example.

set indexprog = /usr/local/httpd/wais/waisindex
#       The full pathname to your waisindex program.
```

```
set nonomatch
cd $rootdir
set num = 0
foreach pathname (`du $rootdir | cut -f2 | tail -r`)

        echo "The current pathname is: $pathname"
        if ($num == 0) then
                set exportflag = "-export"
        else
                set exportflag = "-a"
        endif
        $indexprog -l 0 -nopairs -nocat -d $index $exportflag $pathname/*.html
        $indexprog -l 0 -nopairs -nocat -d $index -a $pathname/*.txt
        $indexprog -l 0 -nopairs -nocat -d $index -a $pathname/*.c
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.ps
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.gif
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.au
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.hqx
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.xbm
```

```
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.mpg
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.pict
        $indexprog -nocontents -l 0 -nopairs -nocat -d $index -a $pathname/*.tiff
        @ num++
end
echo "$num directories were indexed."
```

Make sure you've configured everything correctly, then run this script. The more files your Web site has, the longer indexing will take.

Taking as an example the above configuration in the script, you'd have the directory `/usr/local/httpd/wais/sources` and a number of files with the prefix `index` in the directory. The name of the database you've just created is `index.src`.

If you're using SWISH, you'll want to create a configuration file, call it something such as **swish.conf**, and place it somewhere such as `/usr/local/httpd/swish/` (if you're using NCSA's httpd server). The configuration file below will do pretty much the same thing as the above script for waisindex:

```
# SWISH configuration file

IndexDir /usr/local/www
# This is the root directory of the Web tree you want to index.

IndexFile /usr/local/httpd/swish/sources/index.swish
# This is the name your SWISH index will be built as.

IndexOnly .html .txt .c .ps .gif .au .hqx .xbm .mpg .pict .tiff
# Only files with these suffixes will be indexed.

IndexVerbose yes
# Put this to show indexing information as swish is working.

NoContents .ps .gif .au .hqx .xbm .mpg .pict .tiff
# Files with these suffixes won't have their contents indexed,
# only their file names.
```

Now type:

```
swish -c /usr/local/httpd/swish/swish.conf
```

To run swish and index your site.

Taking as an example the above configuration in the script, you'd have the directory `/usr/local/httpd/swish/sources` and one file called `index.swish` in the directory. The name of the database you've just created is `index.swish`.

€The next step€

Now you'll need to grab the C source for **wwwwais**. You can get it at €`http://www.eit.com/software/wwwwais/`€. **Please read the license before downloading the source.** Should you have licensing or business-related questions **only**, please contact Elizabeth Batson at €batson@eit.com€.

Because wwwwais reads in a configuration file, you'll need to specify where the file exists in the source code. You might want to make the path to this file something like `/usr/local/httpd/conf/wwwwais.conf`.

Now compile **wwwwais** - it seems to compile fine with `gcc`.

Put the program in your Web server's `/cgi-bin` directory. It's a €CGI (Common Gateway Interface)€ program, so it should be in this directory. Make sure you've made it executable by everyone, if that's what you want.

€Setting up the configuration file€

The configuration file is easy to set up - it's much like NCSA and CERN server configuration files. Hash marks (#) and blank lines are ignored. To specify a variable and its value, type the variable name, a space, and its value. Here's an example configuration file:

```
# WWWWAIS configuration file

PageTitle "waistitle.html"
# If this is a string, it will be a title only.
# If it specifies an HTML file, this file will be prepended to wwwwais results.

SelfURL "http://www.eit.com/cgi-bin/wwwwais"
# The self-referencing URL for wwwwais.
```

```
MaxHits 40
# The maximum number of results to return.

SortType score
# How results are sorted. This can be "score", "lines", "bytes",
# "title", or "type".

AddrMask all
# Only addresses specified here will be allowed to use the gateway.
# For the above mask option, these rules apply:
# 1) You can use asterisks in specifying the string, at either
#    ends of the string:
#    "192.100.*", "*100*", "*2.100.2"
# 2) You can make lists of masks:
#    "*192.58.2,*.2", "*.100,*171.128*", ".58.2,*100"
# 3) A mask without asterisks will match EXACTLY:
#    "192.100.58.2"
# 4) Define as "all" to allow all sites.
```

```
WaisqBin /usr/local/bin/waisq
# The full path to your waisq program.
WaissearchBin /usr/local/bin/waissearch
# The full path to your waissearch program.
SwishBin /usr/local/bin/swish
# The full path to your swish program.

SwishSource /usr/local/httpd/wais/index/index.swish "Search EIT's Web (plain
results)"
SourceRules replace "/usr/local/www/" "http://www.eit.com/"
WaisSource /usr/local/httpd/wais/index/index.src "Search EIT's Web (plain
results)"
SourceRules replace "/usr/local/www/" "http://www.eit.com/"
WaisSource /usr/local/httpd/wais/index/index.src "Search EIT's Web (bolded resul
ts)"
SourceRules replace "/usr/local/www/" "/"
SourceRules prepend "http://www.eit.com/cgi-bin/print_hit_bold.pl"
SourceRules append "?$KEYWORDS#first_hit"
WaisSource quake.think.com 210 directory-of-servers "WAIS directory of servers"
# WAIS source file descriptions.
```

```
# For waisq sources:
#    WaisSource full_path_to_source/source.src "description"
# For waissearch sources:
#    WaisSource host.name port source "description"
# For swish sources:
#    SwishSource full_path_to_source/source.swish "description"

UseIcons yes
# Define as "yes" or "no" if you do or don't want to use icons.

IconUrl http://www.eit.com/software/wwwwais/icons
# Where all your icons are kept.

TypeDef .html "HTML file" $ICONURL/text.xbm text/html
TypeDef .txt "text file" $ICONURL/text.xbm text/plain
TypeDef .ps "PostScript file" $ICONURL/image.xbm application/postscript
TypeDef .gif "GIF image" $ICONURL/image.xbm image/gif
TypeDef .src "WAIS index" $ICONURL/index.xbm text/plain
TypeDef .?? "unknown" $ICONURL/unknown.xbm text/plain
# Information for figuring out file types based on suffix.
```

```
# Suffix matching is case insensitive.
#    TypeDef .suffix "description" file://url.to.icon.for.this.type/ MIME-type
# You can use $ICONURL in the icon URL to substitute the root icon directory.
```

Variables such as **SelfURL**, **SortType**, and **MaxHits** should be fairly explanatory; see below for a list of options. You can use the **AddrMask** option to allow only certain sites to search your site, and **WaisqBin**, **WaissearchBin**, and **SwishBin** are pointers to your `waisq`, `waissearch`, and `swish` programs, respectively.

€The WaisSource and SwishSource variables€

The **WaisSource** and **SwishSource** variables tells wwwwais the sources you want the user to be able to search. For an index that will be searched with `waisq`, specify the full pathname to the source you want to search (ending in `.src`) and a short description of the database. For WAIS servers, specify the host name, port, source name (without a `.src`) and a short description. For SWISH indexes, specify the full path to the source (ending in `.swish`) and a short description.

If you've specified more than one WAIS and/or SWISH source in your configuration file, a pop-up menu will appear on the wwwwais page with the descriptions as menu items (in the order that you specified the sources). Choose one and start searching! If you have only one source specified, no menu will appear. Just enter your search text and hit return to search that source.

€Using SourceRules€

When results are returned from WAIS servers and waisq, you may get a bunch of funny pathnames to files that you can't access. Using **SourceRules**, you can specify a series of operations to perform on the pathname result to change it into a URL, a CGI program to filter the file through, and so on.

There are three operations you can specify: **replace**, **append**, and **prepend**. They will parse the pathname in the order you've typed these commands. More than one command and its arguments can appear on the same line, but it's easier to read when commands are broken up over a few lines. You can't put a command and its argument(s) on different lines, however.

Commands apply to the WAIS source specified just before the commands. Here's the syntax:

```
replace "the string you want replaced" "what to change it to"
   This replaces all occurrences of the old string
   with the new one.
prepend "a string to add before the result"
append "a string to add after the result"
```

In any command argument, `$KEYWORDS` will be replaced with the keywords you used to search, so you can pass them to filtering programs that can use them. One good program is €print_hit_bold.pl€, a Perl program

which bolds the found text in search results.

Study the above sample configuration file and try things out. You'll find you can do a lot of nifty things with WAIS sites and filters.

€Using TypeDef€

The program often will need to know the types of files it's returning, so you won't be so confused when you get results back, and so your browser will know what to do if you ask for a file. The **TypeDef** option maps different suffixes to MIME types and short descriptions.

On a **TypeDef** line, you need to specify the suffix for the particular type(with a period), a short description to include in results (this shouldn't be any more than two or three words), the URL to the icon representing the file type (unused if you're not displaying icons), and the MIME type corresponding to the particular type.

The information specified with the suffix `.??` will be associated with all other files that wwwwais can't figure out.

€Setting up the HTML form€

In order to create an interface for your WAIS database, stick the following in any HTML page you want the search capability:

```
<form method=GET action="/cgi-bin/wwwwais">
Search for: <input type=text name="keywords" size=40> <input type=submit value="
Search ">
</form>
```

Try entering multiple words to search for (they should be separated by spaces). Using boolean operators (using **and** and **or** in searching) appears to be supported by **waisq**, so you can use them. Try it out!

€Wait! What's the difference between waisq and waissearch?€

Both the **waisq** and **waissearch** programs that come in the freeWAIS distribution search WAIS databases for the information you're looking for. However, **waisq** looks in databases on the host machine you're doing the searching on, and **waissearch** can look in databases on different machines all over the Internet. **swish** is much like **waisq** in that only indexes that are locally available to **wwwwais** can be searched.

The **waissearch** program does things remotely by contacting WAIS servers on different machines, each of which has their own databases. In order for **waissearch** to do its thing, you need to tell it a machine name and a port to connect to, and that machine needs to have a WAIS server of their own running on that port.

By telling **wwwwais** to use the **waissearch** program and specifying a host name and port in the URL (see below), you can search WAIS databases on other machines. If you wish to run your own WAIS server on your machine, make sure you have the **waisserver** program (from the freeWAIS package) and run it like this (all on one line with no returns):

```
./waisserver -p 2010 -d /usr/local/httpd/wais/sources
-e /usr/local/httpd/logs/wais.log &
```

This will run a WAIS server on your machine on port 2010. This server should log its results to a file named `wais.log` and will search through any databases located in the `/usr/local/httpd/wais/sources` directory.

Say you have a database named "index.src" in that directory and you've started your WAIS server with the line above. To search "index.src" using your server, you could call **wwwwais** with a URL like this (all on one line, of course):

```
http://foo.bar.com/cgi-bin/wwwwais?searchprog=waissearch&
host=foo.bar.com&port=2010&source=index&keywords=heart+of+gold
```

€Calling wwwwais with options€

You can call **wwwwais** with different options in the URL:

selection

```
      example: /cgi-bin/wwwwais?selection=none
      example: /cgi-bin/wwwwais?selection="My+WAIS+server"
      example: /cgi-bin/wwwwais?selection="Files+about+me"
  description: Specifies the index source to use (as set up
               in the configuration file). The argument is
               the source description. If selection
               is not defined as "none", any corresponding
               source information in the configuration file
               will override arguments and environment
               variables.
```

searchprog

```
      example: /cgi-bin/wwwwais?searchprog=waissearch
```

description: Specifies the program to do the searching.
This can be "waisq", "waissearch", or
"swish".

source

example: /cgi-bin/wwwwais?source=index.src
description: Specifies the index database to search.

sourcedir

example: /cgi-bin/wwwwais?sourcedir=/usr/local/sources
description: Specifies the directory the index database
resides in. (You can have multiple databases
in the same directory)

maxhits

example: /cgi-bin/wwwwais?maxhits=40
description: Determines the maximum number of URLs
to return after a search.

keywords

example: /cgi-bin/wwwwais?keywords=these+are+keywords
description: You can specify the search keywords by using the **keywords** label.

isindex
example: /cgi-bin/wwwwais?isindex=these+are+keywords
description: This works the same as **keywords**.

sorttype
example: /cgi-bin/wwwwais?sorttype=bytes
description: This determines how wwwwais sorts its output. Valid arguments for **sort** are "score", "lines", "bytes", "title", and "type".

version
example: /cgi-bin/wwwwais?version=true
description: This gives the version information for wwwwais and the waisq or waissearch program it runs only.

host

example: /cgi-bin/wwwwais?host=eit.com
description: This gives the host machine to search. This is only valid when using waissearch to do the searching.

port

example: /cgi-bin/wwwwais?port=2010
description: This gives the port of the wais server that will be doing the searching. This is only valid when using waissearch with wwwwais.

useicons

example: /cgi-bin/wwwwais?useicons=yes
description: This tells wwwwais to use icons for different files in the search results. This can be "yes" or "no".

iconurl

example: /cgi-bin/wwwwais?iconurl=http://www.eit.com/icons/

```
  description: This tells wwwwais the master URL at which to
               find the icons it may need.

<keywords only>
       example: /cgi-bin/wwwwais?these+are+keywords
  description: Keywords can be specified by themselves from
               "isindex" forms. This only works if no other
               options are used.

<no arguments>
       example: /cgi-bin/wwwwais
  description: Calling the program with no arguments brings
               up a blank field in which users can enter
               search keywords.
```

Examples of specifying multiple options in the URL:

```
  /cgi-bin/wwwwais?isindex=these+are+keywords&maxhits=80
  /cgi-bin/wwwwais?source=index.src&keywords=test+search
```

There are other ways wwwwais can get variable information - you can specify these variables in forms using either the GET or POST methods, and PATH_INFO is supported as well, so you can make something like:

```
<form method=GET
action="/cgi-bin/wwwwais/host=quake.think.com&port=210&searchprog=waissearch">
Search for: <input type=text name="keywords" size=40> <input type=submit value="
Search ">
</form>
```

The above form sets wwwwais up to search the WAIS server at `quake.think.com`, port 210. Note that you can use the POST method as well in this example, and the result will be exactly the same.

Environment variables are also supported - just put "WWWW_" before variables and make everything uppercase. For instance, instead of putting **searchprog=waisq** in a URL, you can type `setenv WWWW_SEARCHPROG waisq` and run wwwwais from a shell script.

€Other pointers€

Here are some other WAIS resources out there:

· €WAIS and WWW pointers€
  This is a good page full of references to other servers, gateways, and FAQs.
· €WAIS, Inc.€
  Learn about WAIS standards, browse their FTP and gopher sites, etc.

€That's it!€

You can take advantage of the command-line options by creating forms which give the user control over sorting, searching multiple databases, etc. Let me know if you make any nice interfaces with this!

If you come across any bugs or problems while searching a WAIS server, please send me the host, port, and source information so I can try things out and track down the problem.

If you know of other file indexers that can be interfaced with WWWWAIS, I highly recommend that you try modifying WWWWAIS to do so - after all, the point is to provide a single, easy to use method of searching many different types of indexes.

· As always, patches, improvements, suggestions, and corrections are gratefully accepted. Send 'em all to **Kevin Hughes** at €kevinh@eit.com€.
· Due to the inordinate amount of email Kevin gets, he makes no promises that he will have time to

respond to your message. He will eventually read everything you send him, however.

· **If you have questions about licensing**, please send email to Elizabeth Batson, €batson@eit.com€. She should only be contacted if you have licensing or business related questions.

Last update: 8/25/95