Webinator WWW Site Indexer Version 2.5

Thunderstone Software

November 6, 2007

Contents

0.1 Overview

Webinator consists of two programs. The indexing or walking program and the searching program. The gw program is the indexing portion of Webinator. It will create a database for HTML pages, retrieve pages from specified sites, and index them for searching with the Texis Web Script interface.

The search portion of Webinator is a program called Texis and a script written in the Texis Web Script language. (Support for the version 1 ”webinator” program has been dropped.)

0.1.1 Features

Here are some of its features:

- All actions are logged (see `-l`).
- One or more Web sites may be indexed into a single database.
- Multiple databases may be maintained (see `-d`).
- A Web site may be copied to the local file system (see `-c`).
- Multiple copies may be run simultaneously against the same database.
- Robots.txt is respected.
- Support for proxy servers.
- Support for meta data.
- Totally customizable search interface.

There are many more features and options to tailor its behavior to your needs.

0.1.2 Obtaining Webinator

Webinator may be obtained from `http://www.thunderstone.com/webinator/`. Follow the instructions there to acquire the package for your platform. After registering you will be given a URL to a compressed tar file containing binaries for your specified platform.

0.1.3 Technical Support

Support for Webinator is via a searchable web message board. It is located at the following url:

`http://thunderstone.master.com/texis/master/search/msgboard.html`

Anyone may read the discussions. To post you must create an account (it’s free) and be logged in. Also, once you are signed up you may ”subscribe” to periodic email notifications of new postings to the board. You may select hourly, daily, or weekly notification of new postings.

If you subscribe to periodic notifications and, at some point in the future, no longer wish to receive them you may select ”subscribe” again to enter the admin area where you may delete your subscriptions.

Do NOT attempt to get support for free Webinator by any other email or voice channel.

Other Webinator resources, such as FAQ, alternate search examples, and such may be found at Webinator's home page `http://www.thunderstone.com/texis/site/pages/webinator.html`.

0.2 Installation

0.2.1 Download and Install

Download the `webinator.tar.Z` file, from the URL given to you during the registration procedure, to a temporary directory on your machine. Then uncompress it, extract it, and run the install script.

```
   uncompress webinator.tar.Z
   tar xvf webinator.tar
   ./install
   rm install widb.tar webinator.tar
or:
   uncompress <webinator.tar.Z | tar xvf -
   ./install
   rm install widb.tar webinator.tar.Z
```

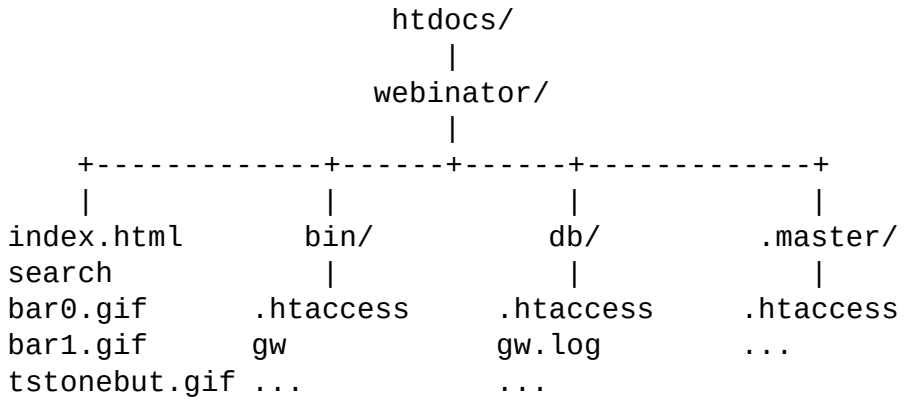
Run the install script as the user that owns your WEB server directories. Be prepared to answer questions about where your CGI programs and HTML pages reside and what user the WEB server runs as. The directory that the install is run from must be writable by yourself and WEB server user.

If you move your WEB server directories around after installing Webinator, you will have to re-install it.

The tar files and install script are not needed after installation is successfully completed.

0.2.2 Filesystem Layout

Webinator is installed underneath your HTML document tree. It consists of several subdirectories. Assuming that your HTML document root is `htdocs`, this will be the structure:



All of the directories that should not be referenced by web browsers contain a `.htaccess` file that denies all access.

The `webinator` directory contains the search interface scripts, several GIF files used by the search interfaces, and an `index.html` that contains a pointer to the search program as an example of how to start it.

The `bin` directory contains the `gw` program and any other related utilities and readme's. The `gw` program may be moved to or run from any desired directory.

The `db` directory contains the default HTML database. It is initially empty. This is the database that `gw` and `webinator` will access if none is specified. It also contains the log file maintained by `gw` when working on that database. Each database will have its own log file.

The `.master` directory contains files that are referenced when creating a new database.

All of the files under the `webinator` tree must be readable and writable by the WEB server. Also, `gw` must be run as the WEB server user or be made `setuid` to it by running commands similar to the following:

```
chown WEBUSERNAME gw
chmod u+s gw
```

The above commands will have to be run as the `root` user on most systems.

0.2.3 Customizing Webinator's Appearance

You may change any and all aspects of the search program's appearance and behavior by modifying one of the supplied scripts or writing an altogether new one.

See `http://www.thunderstone.com/webinator/example/` for some examples.

See the Texis Web Script (Vortex) manual at the Thunderstone web site, `http://www.thunderstone.com`, for details.

0.3 Operation

0.3.1 Database Usage

`gw` maintains a database that contains text from HTML pages, links to other pages, and a list of pages yet to retrieve. The list of pages yet to retrieve is called the "todo" list.

When `gw` runs it inserts any specified URL into the todo list. It then begins taking URLs from the todo list. It retrieves and stores the HTML page and its references. Each reference not seen before is also placed into the todo list. Processing continues until there is nothing left in the todo list.

If `gw` is killed it will finish the page it is working on and exit. When run again with no URL it will pick up where it left off taking URLs from the todo list.

By default `gw` operates on the database in the current directory (if there is one) or the default one as configured during installation. This may be overridden with the `-d` option discussed later.

0.3.2 Indexing

During initial database creation the indices needed for searching are not built. This will speed up the retrieval process. After the first walking session, you need to tell `gw` to create the search indices with the `-index` option. You will also need to give it the same option to update the indices after any significant additions are made to the database. Subsequent `-index` commands will update the existing indices, not rebuild them from scratch.

The database is always searchable, even if retrieval is in progress.

0.3.3 Robots.txt page exclusion

On the first access to a site the file `/robots.txt` will be retrieved, if its exists. Settings there will be respected. Anything in the todo list that is disallowed by `robots.txt` will be discarded.

0.3.4 Command Line Reference

SYNOPSIS

```
gw [options] [URL]

gw -s[X] [options] SQL
```

DESCRIPTION

In the first form the specified URL, if any, will be added to the todo list. It will then begin fetching everything from the todo list. The new HREF's from each fetched page will be added to the todo list. Processing will continue until there is nothing left in the todo list.

In the second form it will perform a SQL query against the database. Only `SELECT` and `DELETE` statements are allowed and only one statement may be executed at a time. Do not include the trailing semi-colon (`;`).

Database Tables and Fields

These are the database fields:

Table: html

```
Field     Description
--------  --------------
Url       The URL of the real HTML page
Title     The Title of the page
Body      The textual content of the page
Meta      The selected meta data from the page, separated by newlines
Visited   The date the page was fetched
Depth     The number of URLs traversed to reach the page
Dlsecs    The total number of seconds to descend all of
          the links to reach the page
id        Unique record id. Or hash code when \verb‘-unique‘ is enabled.
New       Currently unused
```

Table: refs

```
Field     Description
--------  --------------
Url       The URL of the an HTML page
Ref       The URL of a reference (link) on the HTML page
id        Unique record id.
```

Table: error

```
Field     Description
--------  --------------
Url       The URL of the an HTML page that could not be retrieved
Reason    The reason it could not be retrieved
id        Unique record id (includes timestamp info).
```

Table: querylog

Note: This table is only used if query logging is enabled in the search interface.

```
Field     Description
--------  --------------
id        Contains the date and time of the query (unique record id)
Client    The hostname of the web client that performed the query
Query     The user’s query as entered
```

Find out about fields with the following query:

```
gw -st "select TBNAME,NAME,TYPE from SYSCOLUMNS
         where TBNAME not matches ’SYS%’ order by TBNAME,NAME"
```

0.3.5 Command Line Option syntax

The ordering of multiple options on the command line is not generally important.

`gw -d/tmp/mydb -index` is the same as `gw -index -d/tmp/mydb`

Multiple options may NOT be combined. And there must be a space between each option.

Right: `gw -y -g -a`

Wrong: `gw -yga`

Wrong: `gw -y-g-a`

There must not be any space between an option and its parameter (if any).

Right: `gw -d/tmp/testdb`

Wrong: `gw -d /tmp/testdb`

Case is significant.

`-h` is different than `-H`

The URL(s) or SQL statement must be the last thing on the command line, after any options.

0.3.6 Help and Configuration

Show version (-version)

Syntax: `-version`

Displays version information and a short usage message.

Show help (-h)

Syntax: `-h`

Displays a short synopsis of all available options.

Use option file (-m)

Syntax: `-mFILE`

Read more options from the specified FILE. This allows you to specify a file that contains commonly used `gw` options. Place options in FILE one per line without the leading `-`. Blank lines and lines beginning with

are ignored. Make sure there are no leading or trailing spaces or tabs on the option lines. Do not use quotes that you would use on the command line to protect special characters from the shell.

Other options, including `-m`, may still be placed on the command line when using `-m`. This option may be used multiple times. It may even be used within an option file. Just be sure not reference an option file from within itself!

Options are read in the order specified. Options following the `-m` option may override those in the option file.

e.g.: "`gw -o -msomesite.set -v http://www.somesite.com/`" will read options from the file named `somesite.set`.

Save settings (-save)

Syntax: `-save=PROFILE`

This will save all current options and URLs to the named `PROFILE` which can then be used in later runs using the `-recall` option. By default options and URLs are saved to a profile named `lastrun`. Each save will totally replace all option settings with the current settings. URLs will be added to any existing list for the given profile.

Recall settings (-recall)

Syntax: `-recall` or: `-recall=PROFILE`

This will recall option settings and URLs from the named `PROFILE` which should be a name that was previously used with the `-save` option. `PROFILE` may also be `lastrun` indicating the options used in the last run that did not specify a `-save` option. `-recall` is shorthand for `-recall=lastrun`. `PROFILE` may also be a profile that was saved using the web interface to gw.

Don't save settings (-O)

Syntax: `-O`

Don't save options or URLs to any profile. This will prevent -recall and -rewalk from working and gw will not remember what URLs you specified for walking in the past. This is useful if you do many walks with huge lists of specific URLs.

Note: The `-d` option, if used, must be specified before `-recall`.

Set verbosity (-v)

Syntax: `-v#`

Sets the verbosity level to #. The default verbosity level is 2. Verbosity level does not effect logging (see `-l`) output.

Verbosity Levels

```
Level        Description
-----------  --------------
0            Issue no messages except errors
1            Display URLs for retrieved pages
2            Display total pages/references seen
3            Display more details about what has
             been seen
4 thru 9     Various debugging messages
```

Each level includes the previous levels.

e.g.: "`gw -v4`" will set the verbosity level to four.

Increase verbosity (-v)

Syntax: `-v`

Increases verbosity. Each `-v` increases the verbosity one level. See `-v#` above.

e.g.: "`gw -v`" will set the verbosity level to three since the default is 2.

0.3.7 Database and File Usage

Set database name (-d)

Syntax: `-dDB`

Use the database named DB. The default is to use the database in the current directory if one exists. If a database does not exist in the current directory you will be asked if it is OK to use the default search database as installed into your Web server document directory. Specifying `-d-` will use the default search database without asking.

e.g.: "`gw -d/tmp/testdb`" uses the database in `/tmp/testdb`.

e.g.: "`gw -d/usr/local/etc/httpd/htdocs/webinator/fun`" uses the database in `/usr/local/etc/httpd/htdocs/webinator/fun`.

For a database to be accessible to the Webinator search program it must be fully specified in the search script or be specified from the default search form using the `db` variable.

Examples:

In the search script: `<DB=/usr/local/etc/httpd/htdocs/webinator/fun>`

In a URL: `http://www.mysite.com/cgi-bin/texis/webinator/search?db=fun`

or: `http://www.mysite.com/cgi-bin/texis.exe/webinator/search?db=fun`

In the search form: `<input type=hidden name=db value=fun>`

Log file name (-l)

Syntax: `-lFILE`

Log to FILE. The default is to log to a file named `gw.log` in the database directory. Retrieved pages and pages that could not be accessed are logged. The log is in addition to on screen output. Control the amount of on screen output with the `-v` option.

e.g.: "`gw -l/tmp/webinator.log`" logs to the file `/tmp/webinator.log`

Set index document name (-I)

Syntax: `-IFILENAME`

Set the filename assumed for directory URLs. The default is "`index.html`". This filename will be removed from stored URLs to prevent redundant fetches of the page. So the URLs "`http://www.mysite.com/fun/`" and "`http://www.mysite.com/fun/index.html`" will be considered the same and only be fetched once.

This is also the filename that will be used for directory references when using the "`-c`" option. So the URL "`http://www.mysite.com/fun/`" will be stored in the file "`fun/index.html`".

0.3.8 Database Management

Create database (-create)

Syntax: `-create`

Creates a new empty database. This is not usually needed because the database will be created if it does not exist when you attempt to use it.

During database creation there is one table created that is for optional query logging in the search interface. You will have to edit the search interface slightly to uncomment the query logging code to enable query logging.

Wipe database (-wipe)

Syntax: `-wipe`

Wipes the existing database. This will completely empty the database. The database will be left as if it were just created. Any stored options will be forgotten. If `-unique` had been used, it will be forgotten.

The log file, `gw.log` by default, is not deleted by this operation. You may wish to delete it by hand if it's getting overly large.

Wipe todo list (-wipetodo)

Syntax: `-wipetodo`

Wipes out the list of URLs that still need to be walked. These may be left over by an interrupted walk or a walk that reached it's page limit. Normally when you restart gw after it has been interrupted, it will pickup where it left off in the todo list. This option will clear that list so you can walk something new without going after the unwalked pages.

This option appeared in version 2.5 release 19980921

Update search indices (-index or -i)

Syntax: `-index` or: `-i`

Creates or updates the search indices to reflect newly captured pages. Searches will work and be able to find the new pages before the indices are updated. It will just take longer. The indices also need to be up to date for the Ranked search to work. This is not generally needed as gw will automatically update indices upon completion of a walk unless the `-noindex` option is used.

Don't automatically update search indices (-noindex)

Syntax: `-noindex`

Don't automatically update search indices after a walk. Normally gw will automatically perform the `-index` operation upon completion of a walk.

Delete search indices (-unindex)

Syntax: `-unindex`

Drops the search indices. This will speed up downloading of a large number of HTML pages. Rebuild the indices with the `-index` option.

Delete all indices (-dropindex)

Syntax: `-dropindex`

Drop all indices. This is for emergency use only in the event that an index should get corrupted by disk errors, sudden power failures, and the like during a walk. You should use `-index` after this to rebuild the indices.

Rewalk (-rewalk)

Syntax: `-rewalk`

Rewalk the site(s) represented in the current database. The rewalk is performed to a temporary database. When the rewalk is complete, the temporary database is indexed and the current database is replaced by it. gw should not be run from within the database directory when using this option.

The options and critical files such as `top.html` and `bottom.html` are copied into the new database from the old. Otherwise, the new database is a fresh directory. The existing `gw.log` file, query log, and any non-database files will be lost.

`-rewalk` ignores all options except `-d`, `-v` and `-recall`. It uses the options from your last walk, or those specifed by `-recall`. If you wish to use different options, you need to perform a complete walk manually once with those options.

Note: This option will not work on databases created with Webinator 1.x .

Rewalk on a schedule (-rewalk)

Syntax: `-rewalk="TIME_SPEC"`

This performs the function of `-rewalk` on the schedule specified in `TIME_SPEC`. `TIME_SPEC` may consist of a time, day, and/or a repeat frequency.

A time consists of the hour, optionally followed by the minute, optionally followed by AM or PM. It may also be one of the units `hour`, `day`, or `week`.

Days may be abbreviated to the first 3 characters of the name.

Use the word `every` to get repeating behavior.

The order of terms in the time spec is not critical. Use the `-h` option AFTER the `-rewalk` to see how `gw` interprets your schedule without executing it.

When running on a schedule, gw will sleep until the first rewalk time, rewalk, then, if appropriate, sleep until the next repeat period. You may want to run gw in the background when using the scheduler.

Examples:

rewalk once at 1 am: -rewalk=”1am”

rewalk once at 1:30 PM on Saturday: -rewalk=”1:30pm saturday”

rewalk once 3 hours from now: -rewalk=”3 hours”

rewalk every day at 1 am: -rewalk=”every 1am”

rewalk every Saturday at 11 PM: -rewalk=”every 11pm saturday”

rewalk every 12 hours from now: -rewalk=”every 12 hours”

Set word expression (-k)

Syntax: `-k"expr"`

Sets the word matching expression for `-index` to EXPR. EXPR is a REX expression defining what is

considered a word during the index process. The default expression is `"\alnum{2,30}"`, which means from 2 thru 30 alphanumeric characters. You may specify `-k` multiple times to define multiple expressions if you can't define your idea of a word in one expression. When using `-k` the default expression is not used so you will have to specify it if you want it in addition to whatever you are adding.

To change the expression on an already indexed database, you will have to use `-unindex` followed by a `-index' and the desired \verb`-k' option(s).

e.g.: "`gw -k">>\alpha=\alnum{1,20}"`" will index "words" beginning with an alphabetic character followed by 1 to 20 alphabetic or numeric characters.

Execute arbitrary SQL (-s)

Syntax: `-s[X]`

This option will cause `gw` to perform a SQL query against the database instead of retrieve pages. Place a quoted SQL statement on the command line instead of a URL. Only `SELECT` and `DELETE` statements are allowed and only one statement may be executed at a time. Do not include the trailing semi-colon (`;`).

Resultant rows will be formatted according to `X` where `X` is taken from the following table.

Format Styles

```
Style  Description
-----  --------------
  h    For an HTML table
  t    For tab separated values
  c    For comma separated values
  f    For tagged values. This is the default
  q    For no formatting
```

If `X` is not specified `f` is assumed.

e.g.:

```
gw -st "select Url,Title from html where Depth=0"
```

Will result in a two column listing of URLs and Titles at a depth of 0.

Also see Section 0.3.4 about what fields are in the database.

Enable DNS caching (-dnscache)

Syntax: `-dnscache`

This option will create a hosts table that will be used during subsequent walks to speed up host name lookups. Every host name encountered will be stored in the table and subsequent lookups for any host seen before will not require querying your nameserver. This is helpful when you walk a lot of different sites and have slow name service. The hosts table will NOT be cleared when the database is wiped with `-wipe`.

Disable DNS caching (-nodnscache)

Syntax: `-nodnscache`

This option will delete the hosts table created by `-dnscache`. All entries in it will be lost and caching will be turned off for all subsequent walks.

Enable extra page duplication prevention (-unique)

Syntax: `-unique`

This option will enable extra checking for duplicate documents. Documents with the same content will only be stored once, even if their URLs are different. This is accomplished by placing a unique index on the id field of the html table and storing a hash code for the HTML source of the document there instead of the normal counter variable. All subsequent walks will perform hashing.

This option should only be used on an empty database since any existing counter id's would not be proper hash codes. This option must be respecified, if desired, after performing a database wipe with `-wipe`.

NOTE: Dynamic debugging insertions into the HTML source, such as the current time, whether visible or in comments, will change the hash thereby defeating this feature.

Disable extra page duplication prevention (-nounique)

Syntax: `-nounique`

This option will turn off the extra checking for duplicate documents previously enabled with `-unique`. The unique index on the id field of the html table will be dropped and all subsequent walks will store counters in the id field instead of hash codes. You will need to perform a `-index` to recreate the normal id index when you are ready to use the database for searching.

0.3.9 Page Fetching Behavior

Grab off-site pages (-o)

Syntax: `-o`

Allow grabbing of individual off-site pages. By default `gw` will not retrieve pages that are not on the same machine as the initial URL. With this option pages not on the initial machine will be retrieved, but none of the pages that they reference will be.

Don't add to todo (-a)

Syntax: `-a`

Don’t add pages referenced by retrieved pages to the todo list. This allows you to add a single page to the database without getting anything it refers to.

Allow an entire domain (-domain)

Syntax: `-domain=DOMAIN`

Allow walk to fetch pages from any host in `DOMAIN`. Any URL having a hostname ending in `DOMAIN` will be accepted. This option may be specified more than once to specify multiple acceptable domains.

e.g.: `-domain=othersite.com http://www.mysite.com/` will walk all of `www.mysite.com` and any URLs referring to any machine in `othersite.com`.

This option is not a “restricter”, but an “enabler”. All hosts specified will be walked and any others that match the given domain(s) will also be walked.

Note: This option will NOT hunt down and walk every web server in the specified domain. It will simply allow walking them if they are referred to.

Allow an entire network (-network)

Syntax: `-network=ADDRESS`

Allow walk to fetch pages from any host within the network specified by the numeric `ADDRESS`. This option may be specified more than once to specify multiple acceptable networks.

e.g.: `-network=192.0.2 http://www.mysite.com/` will walk all of `www.mysite.com` and any URLs referring to any machine in the network `192.0.2`.

Note: This option will NOT hunt down and walk every web server in the specified network. It will simply allow walking them if they are referred to.

Control allowed extensions (-f and -F)

Syntax: `-fEXT` or: `-FEXT`

The `-f` option will add EXT to the list of allowed file extensions. The `-F` option will remove EXT from the list of allowed file extensions. Do not include the ’.’ in EXT. It is implied. The default list of allowed extensions is:

```
html  htm  txt
```

e.g.: “`gw -Ftxt -fshtml`” allows just `".html"`, `".htm"`, and `".shtml"` files.

`-f` and `-F` may be used multiple times as needed to specify multiple extensions.

Note: The pre-version 2 behavior of wiping the entire extension list when no `EXT` was supplied to `-f` has been removed.

Define a plugin (-n)

Syntax: `-n"mime-type,extension,filter-command"`

Define a plugin for processing non-HTML file types. A plugin is basically any filter program that accepts a given file type on its input and generates a flat text version on its output. To define a plugin you specify the document's "mime" type, its extension type and the name of the filter program. Any document fetched having the specified extension or mime type will be processed through the filter command instead of the default HTML to text converter inside `gw`.

e.g.: "`gw -n"application/pdf,pdf,pdftotx"`" defines a PDF plugin that will pass documents having extension "`.pdf`" or mime type "`application/pdf`" through the filter program "`pdftotx`".

The filter program must be in the execution path or have its full path specified.

This may be used multiple times and is a good candidate for placing into an option file (`-m` section 0.3.6) because of its length. Using this option also implies "`-fextension`" and "`-Tmime-type`" so you don't have to specify them.

If your non-HTML files don't always have the specified extension and can only be identified by the mime type you might need to use the `-z` option to increase the maximum download file size (See section 0.3.10). Things like PDF documents are often much larger than the text contained in them.

Allow cgi-bin paths (-C)

Syntax: `-C`

Allow retrieval of files having paths beginning with "/cgi-bin/". These are not retrieved by default. These paths are typically not desired because they are usually dynamically generated, continuously changing, and potentially very deep.

Allow all file types (-y)

Syntax: `-y`

Retrieve all files instead of just HTML. This will turn off checking of URL types in HREF's. All URLs will be retrieved.

Add allowed MIME type (-T)

Syntax: `-Tmime-type`

Add the given mime data type to the list of allowed types. Mime types have the syntax `type/subtype`. Either `type` or `subtype` may be `*` to mean "any". By default all text types are allowed (`text/*`).

Delete allowed MIME type (-S)

Syntax: `-Smime-type`

Remove the given mime data type from the list of allowed types.

Define prefix exclusions (-x)

Syntax: `-xURL`

Excludes URLs with the prefix `URL`. This option may be specified multiple times to exclude multiple prefixes. The effect is the same as if the remote system's robots.txt file had the same entries.

e.g.: "`gw -xhttp://www.mysite.com/text`" will skip all URLs beginning with `http://www.mysite.com/text`

Define path pattern exclusions (-x/)

Syntax: `-x/EXPR`

Excludes URLs with the path component matching the REX expression `EXPR`. This option may be specified multiple times to exclude multiple patterns.

e.g.: "`gw -x/personal\digit`" will skip all URLs containing "personal" followed by a digit.

This option appeared in version 2.53 release 19990226

Define query pattern exclusions (-exquery)

Syntax: `-exquery=EXPR`

Excludes URLs with the query component (the part after the `?`) matching the REX expression `EXPR`. This option may be specified multiple times to exclude multiple patterns.

e.g.: "`gw -exquery=similar`" will skip all URLs containing "similar" in the query string.

Note: This option appeared in version 2.54 release 19990416

Define URL prefix restrictions (-j)

Syntax: `-jURL`

Excludes URLs not having the prefix `URL`. This option may be specified multiple times to include multiple prefixes. This will allow you to walk a subdirectory of a server without accidentally wandering out from under that subdirectory.

e.g.: "`gw -jhttp://www.mysite.com/docs/`" will skip all URLs NOT beginning with `http://www.mysite.com/docs/`

Ignore robots.txt (-r)

Syntax: `-r`

Ignore robots.txt. Normally `gw` will initially get `/robots.txt` from any site being indexed and respect its settings for what prefixes to ignore. This option will disable the use of robots.txt and retrieve everything. Use of this option in not generally recommended. Any URLs specified with `-x` will still be excluded when using this option.

Set login user name (-U)

Syntax: `-UNAME`

Use `NAME` as the user name when accessing protected HTML pages. The default is to skip protected pages.

Set login password (-P)

Syntax: `-PPASS`

Use `PASS` as the user password when accessing protected HTML pages. The default is to skip protected pages.

Set proxy to use (-proxy)

Syntax: `-proxy=URL`

Fetch pages through the supplied proxy URL instead of directly from the server.

Don’t lookup every hostname (-L)

Syntax: `-L`

Don’t lookup every hostname encountered to resolve all potential aliases of desired hosts. This can speed up the walk of pages with many different hosts referenced because the name server need not be consulted for each one. Not using this option will guarantee that a given host will always be treated as the same machine no matter what alias it is referenced by.

This option should not be used with `-network`.

Don’t report as Mozilla (-M)

Syntax: `-M`

Don’t report user agent as “Mozilla” to web servers. Gw will normally report it self as Mozilla, with a comment of Webinator, to web servers. Using this option will make it just report as Webinator.

Many web servers return different content based on what client is asking. You often get better pages when using Netscape (i.e. Mozilla) so gw pretends that it is.

Set reported Mozilla version number (-M)

Syntax: `-Mversion`

Set the version of Mozilla to report to web servers. Normally gw reports itself as Mozilla version 3.0. Use this option to report as a different version of Mozilla.

Set reported User-Agent (-M/)

Syntax: `-M/agentstring`

Normally gw reports itself as Mozilla version 3.0. Use this option to report as any arbitrary agent as defined by `agentstring`.

This option appeared in version 2.55 release 19990722

Index meta data (-meta)

Syntax: `-meta=NAMES`

This option tells gw to look for the specified meta data in fetched documents and store it in the database. This data will then be included in text searches. `NAMES` is a comma separated list of the meta data names to look for and store. This option may be specified more than once.

e.g.: `-meta=keywords,description` will look for the `keywords` and `description` meta-data fields in fetched documents.

e.g.: `-meta=keywords -meta=description` is the same as the above.

Meta data will be stored in the database in the order that you specify the names. During ranked user searches, hits in meta data will be ranked lower than those in the title, and higher than those in the body.

Strip query strings (-Q)

Syntax: `-Q`

Strip query strings from all URLs. Some URLs have query strings on the end indicated by a question mark (?). By default these are left alone. This option will strip them.

Don’t include alt text (-N)

Syntax: `-N`

Don’t include ALT text from IMG AREA tags in pages when storing the page text in the database. By default ALT text is respected and stored and will be searched.

Don’t store refs (-R)

Syntax: -R

Don’t add URLs referenced by retrieved pages to the refs table. This is not generally useful except for those using Webinator in conjunction with full Texis.

Force page insertion (-Force)

Syntax: -Force

Normally gw will not fetch pages it already has in the database (unless using -e section 0.3.11). This option will force gw to fetch pages that it already has. This allows you to refetch pages without having to first delete them (section 0.4.8).

This option appeared in version 2.5 release 19980223

0.3.10 Time and Size Limits

Set delay between pages (-w)

Syntax: -w#

Causes gw to wait # seconds between page fetches. The time to retrieve and process a page is subtracted from this time, so the actual sleep time will generally be a little less than that specified. The default is 0 (no delay). Versions prior to January 1998 had a default wait of five seconds.

e.g.: “gw -w2” waits two seconds between page fetches.

Set page timeout (-t)

Syntax: -t#

Causes gw to timeout after # seconds during each page fetch. The default is 30 seconds. This includes the time to make the connection to the server and download a single page. A time out will not cause the entire process to quit. That page will just be skipped and considered unavailable.

e.g.: “gw -t45” allows up to 45 seconds for a page fetch before timing out.

Limit depth of walk (-D)

Syntax: -D#

Limits the depth of page retrieval to #. Depth is determined by counting how many links were traversed to reach a particular page. Depth is normally unlimited. This option also implies the `-b` option.

e.g.: "`gw -D1`" limits traversal to the top two (0 and 1) levels.

Walk breadth first (-b)

Syntax: `-b`

Retrieve pages breadth first. Normally pages will be retrieved in whatever order they are seen in. This option will cause `gw` to retrieve all top level pages first, then second level, and so on. Depth is determined by counting how many links were traversed to reach a particular page.

Limit run time (-q)

Syntax: `-q#[m|h]` or: `-qH:M`

Causes the run to quit after # seconds. The letters `m` or `h` may be appended to specify minutes or hours respectively. Alternatively, an absolute time may be specified in the form `H:M` where `H` is the hour, specified in 24 hour format. And `M` is the minute. This is useful for starting an after hours run and ensuring it will quit before working hours.

e.g.: "`gw -q30m`" will quit no later than 30 minutes from the start time.

e.g.: "`gw -q7:30`" will quit no later than 7:30 am.

Limit retrieved pages (-p)

Syntax: `-p#`

Limits the number of pages retrieved in a run to #. Normally everything referenced will be retrieved.

e.g.: "`gw -p100`" will retrieve no more than 100 pages.

Limit retrieved page size (-z)

Syntax: `-z#`

Sets HTML page size limit to # characters. The default is 100,000 characters. Pages larger than the limit will be truncated, not discarded.

e.g.: "`gw -z20000`" will truncate pages at 20,000 characters.

0.3.11 Mode Control

Get a single page (-g)

Syntax: `-g`

Get just the single page specified by `URL` and quit. Nothing in the todo list will be processed. This is a quick way to get a single page into the database without having to process the potentially large todo list.

Reload pages (-e)

Syntax: `-e"YYYY-MM-DD[ HH[:MM[:SS]]]"`

Reload pages older than the specified date. All pages whose `Visited` date is earlier than the specified date will be retrieved and their records updated.

e.g.: "`gw -e"1995-09-01"`" will re-retrieve all pages retrieved before September 1, 1995.

The date may also be specified in relative form using the syntax:

`-e"-AMOUNT UNITS"`

where `AMOUNT` is a number and `UNITS` is one of:

`years`, `months`, `weeks`, `days`, `hours`, `minutes`, or `seconds`

e.g.: "`gw -e"-3 days"`" will re-retrieve all pages earlier than 3 days ago.

See also `-X` (section 0.3.11) and `-V` (section 0.3.11)

Delete missing pages (-X)

Syntax: `-X`

This option is only for use in conjunction with the `-e` (section 0.3.11) option.

When refetching pages with `-e` gw may encounter pages that no longer exist on the webserver. Normally, those pages will not be deleted from the database. Specifying `-X` will cause those pages to be deleted from the database.

This option appeared in version 2.5 release 19980130

Only download modified pages (-V)

Syntax: `-V`

This option is only for use in conjunction with the `-e` (section 0.3.11) option.

When refetching pages with `-e` gw normally fetches the specified pages unconditionally. This option tells gw to use the "`if modified since`" method of fetching from the webserver so it doesn't have to

transfer pages that have not been modified since the last time they were fetched.

This option appeared in version 2.5 release 19980130

Copy web site (-c)

Syntax: `-cDIRECTORY`

Causes `gw` to copy an HTML tree to directory `DIRECTORY` instead of placing it in the database. This is useful for replicating a Web site on another machine. Images will also be copied, whereas they would not be inserted into the database.

A temporary database will be generated during the copy process and deleted when the copy is finished. The location of the temporary database will be `DIRECTORY/_gwtmpdb`. The database will not be deleted if the copy is interrupted so that it may be resumed.

e.g.: "`gw -c/tmp/copy http://www.mysite.com`" will copy the HTML tree under `http://www.mysite.com` to the directory `/tmp/copy`.

Don't reauthor during copy (-A)

Syntax: `-A`

Prevents gw from re-authoring the URLs in retrieved pages when using the `-c` option. Normally gw will re-author URLs encountered in retrieved pages to be relative to the newly copied tree. Using this option prevents any changes to embedded URLs.

Reauthor to specified path during copy (-A)

Syntax: `-Aroot`

Changes the way gw re-authors URLs when using the `-c` option. It will normally try to make embedded URLs relative to the newly copied directory. This will force all embedded relative URLs to have the given `root` prefix prepended.

0.3.12 Other Arguments

URL to walk

A URL to an HTML page to put into the todo list. It should be a fully qualified URL such as `http://www.mysite.com` or `http://www.mysite.com/docs/widget.html`.

You may specify a collection of URLs by placing them in a file, one per line, and specifying the filename preceded by & instead of a URL. Place quotes (") around it on the command line since & is special to the command shell.

e.g.: "`gw "&url.lst"`" will read a list of URLs from the file `url.lst`.

gw can also read the list from its standard input (a pipe). Just use - instead of the filename.

e.g.: "`findurl | gw "&-"`" will read a list of URLs from the output of the fictional command `findurl`.

SQL statement to execute

The SQL statement to execute when using the `-s` option. Only `SELECT` and `DELETE` statements are allowed and only one statement may be executed at a time. Do not include the trailing semi-colon (`;`).

0.4 Procedures and Examples

It is imperative that the database be accessible by both `gw` and your Web server. Installation will make `gw` setuid to your Web server's user id if you provided the correct one. If it is not you must ensure that you run gw as the same user id as your Web server or that you make `gw` setuid to that user. That may be accomplished with the following commands, assuming that your Web server's user is `www`:

```
chown www gw
chmod u+s gw
```

The above commands will have to be run as the `root` user on most systems.

0.4.1 Indexing your Site

After the Webinator is installed the first thing to do is to index your Web site and try out some searches on it. This will involve retrieving all of your web pages with `gw`. Building indices for searching. And accessing the Webinator search page. Assuming that your Web home page is at `http://www.mysite.com` you would perform the following commands:

```
gw http://www.mysite.com
```

Similarly, if you have an HTML tree that is isolated from your normal home page you can index it by specifying its URL:

```
gw http://www.mysite.com/docs/widget.html
```

If you have pages with extensions other than `html`, `htm`, or `txt` you can use the `-f` option to index them. For example, to additionally index files with `asp` and `hlp` extensions use:

```
gw -fasp -fhlp http://www.mysite.com
```

0.4.2 Searching your Index

Search the pages you have indexed by entering the following URL into your favorite Web browser:

```
http://www.mysite.com/cgi-bin/texis/webinator/search/
```

```
or for Windows NT:
```

```
http://www.mysite.com/cgi-bin/texis.exe/webinator/search/
```

The above is a virtual path comprised of 2 parts. "`.../cgi-bin/texis`" is the Texis Web Script interpreter and "`/webinator/search`" is the path to the search script relative to the web server's document root.

You may have to use a slightly different URL if you specified a different cgi directory during installation.

Click `Help` on the search form that comes up to get help on constructing queries.

0.4.3 Page Exclusion and Robots.txt

If there are any HTML trees that you don't want indexed you need to setup a `robots.txt` file or use the `-x` option. For example; if you had a "text only" version of your web server that duplicated the content of your normal server you would not want to index it. (On the other hand if most of your meaningful text is contained in graphics you may want to walk the text tree instead of the normal one since graphics are not searchable.)

Suppose your "text only" pages were all under a directory called `/text`. The simplest way to prevent traversal of that tree would be to use the `-x option`, as in:

```
gw -xhttp://www.mysite.com/text/ http://www.mysite.com
```

That will prevent retrieval of any pages under the `/text` tree. It would get tedious and error prone to enter the same thing every time you indexed your site. That also does not prevent other Web robots from retrieving the `/text` tree. To setup a permanent global exclusion list you need to create a file called `robots.txt` in your document root directory. The format of that file is as follows:

```
User-agent: *
Disallow: /text
```

Where `*` is the name of the robot to block. `*` means any robot not specifically named (all robots in this case since no others are named). Or you could specify the name of the robot. For Webinator it would be `Webinator`. You may specify several "Disallow"s for any given robot.

You may also specify different "Disallow" sets for different robots. Simply insert a blank line and add another "User-agent" line followed by its "Disallow" lines.

Here's a larger example:

```
User-agent: *
Disallow: /text
Disallow: /junk

User-agent: Webinator
Disallow: /text
Disallow: /webinator

User-agent: Scooter
Disallow: /text
Disallow: /junk
Disallow: /big
```

The `Scooter` robot will be blocked from accessing any pages under the `/text`, `/junk`, and `/big` trees. `Webinator` will be blocked from accessing any pages under `/text` and `/webinator`. All other robots will be blocked from accessing pages under `/text` and `/junk`.

Use of `robots.txt` is not enforced in any way. Robots may or may not use it. `gw` will, by default, always look for it and use it if present. This may be disabled with the `-r` option. When using `-r` you may still use `-x` for manual exclusion.

0.4.4 Indexing Other Sites

You may index a site other than your own by specifying its URL just as you would for your own site.

```
gw http://www.anothersite.com
```

Please be kind when indexing other sites. Many are low bandwidth or heavily used already and won't appreciate being hit hard. If you want to index any significant number of sites, please contact Thunderstone, as we may have what you want already. Remember that we are one SQL statement away from turning off any individual free Webinator license.

0.4.5 Indexing Individual Pages

To add an individual HTML page to the database, but don't go after any of its references or any others that may be in the todo list:

```
gw -g -a http://www.mysite.com/special.html
```

This will quickly and immediately add a single page to the database without having to go through the todo list and without storing any refs on the page. To allow addition of refs on the page to the todo list, remove the `-a` option.

Normally if you were to not use the `-g` option the page would go into the todo list and everything in the todo list would be processed. That would be fine if the todo list was empty, but if the last walk was interrupted or still in progress there would be other things in the todo list and `gw` would begin processing those.

0.4.6 Reindexing on a schedule

It is often desirable to reindex a given site on a regular basis because of continuously changing content. You may specify a rewalk schedule to `gw` to handle this for you.

```
gw -rewalk="every day at 1am"
```

will rewalk the site(s) contained in the default database every day at 1:00 am.

```
gw -dmanuals -rewalk="every 11pm saturday"
```

will rewalk the site(s) contained in the database named `manuals` every Saturday at 11:00 PM.

It is also useful to perform a single rewalk at a later time or date to avoid overloading a web server during heavy use periods.

```
gw -rewalk="2am"
```

will rewalk the site(s) contained in the default database at 2:00 am.

0.4.7 Checking for WEB Server Errors

After walking a site you might want to examine the error table to find out if there were bad links or otherwise unaccessible documents. The following will report all errors encountered since the database was created or wiped:

```
gw -st "select Url,Reason from error"
```

You can restrict the report to errors occurring in a specific time interval by using the `id` field as a date. The following will limit the report to errors occurring on April 20 1996.

```
gw -st "select Url,Reason from error
        where id>'1996-04-20' and id<'1996-04-21'"
```

The following will limit the report to errors occurring in the last hour:

```
gw -st "select Url,Reason from error where id>'-1 hour'"
```

The following will report the page that each bad URL occurred on:

```
gw -s "select refs.Url Page,error.Url Error,error.Reason
       from error,refs
       where refs.Ref=error.Url"
```

0.4.8 Removing Pages from the Database

To remove a specific page from the database execute two SQL statements using the `-s` option. First delete the record from the html table, then delete all related records from the references table. For example; to remove the page `http://www.mysite.com/junk.html` use the following commands:

```
gw -s "delete from html where Url='www.mysite.com/junk.html'"
gw -s "delete from refs where Url='www.mysite.com/junk.html'
```

Note the missing `http://` prefix. It is not stored in the database.

You may wish to perform a "select" before a "delete" to see what you are going to delete before committing to it. Using the above example; use the following command to find the title of the document you are about to delete:

```
gw -s "select Url,Title from html
       where Url='www.mysite.com/junk.html'"
```

To remove a group of related pages from the database use a REX Regular Expression to specify the substring or pattern common to all of the pages you want to delete. Precede the substring or expression with slash ('/'). To delete all pages under the `/testdir` tree use the following commands:

```
gw -s "delete from html where Url like '/www.mysite.com/testdir'"
gw -s "delete from refs where Url like '/www.mysite.com/testdir'"
gw -index
```

The `like` means search the field for the argument. The leading slash (`/`) means do a regular expression match. This will delete every record whose Url contains the substring "www.mysite.com/testdir". See the query construction documentation at `http://www.thunderstone.com/webinator/` for help building "like" statements. Also remember to run with the `-index` option to update the indices after deleting numerous pages to speed up searches.

0.4.9 Erasing the Entire Database

If you decide to wipe out your existing database and start over use the following command:

```
gw -wipe
```

This will remove everything from the database and leave it ready for indexing sites (just as if it was newly installed).

0.4.10 Using Multiple Databases

Once you have a live searchable database you may want to build a separate one to contain different kinds of pages or to experiment without destroying your live database. Use the `-d` option to specify a different database.

```
gw -d/usr/local/etc/httpd/htdocs/webinator/fun
     http://www.asite.com/fun/

gw -d/tmp/testdb http://www.asite.com
```

The first pair of commands will create/use a database that is accessible to the searcher by the name of `fun`. The second will create/use a database in `/tmp` called testdb.

When you specify a database that doesn't exist you will get a message that there is no such database and it will be created automatically.

See section 0.3.7 for how to search these alternate databases.

0.4.11 Using Option Files

You may find that you use the same large set of options very often. You may place these options into a file and tell `gw` to read them from the file with the `-m` option.

Place options in a file one per line without the leading `-`. Blank lines and lines beginning with # are ignored. Make sure there are no extraneous spaces or tabs at the end of option lines.

For example, to exclude several paths and include an extra extension when walking you could use an options file like the following:

```
#
# common options for walking www.somesite.com
#
# exclude duplicate text tree
xhttp://www.somesite.com/text/

# exclude some junk
xhttp://www.somesite.com/test/
xhttp://www.somesite.com/personals/

# allow .asp files
fasp

# handle PDF files with purchased PDF plugin
napplication/pdf,pdf,pdftotx
# allow big files so PDF doesn't get truncated
z500000
```

And invoke `gw` with a command like (assuming the above file was called `somesite.set`):

```
gw -msomesite.set http://www.somesite.com/
```

Other options may still be used on the command line, as in:

```
gw -o -msomesite.set -v http://www.somesite.com/
```