

Querying Search Engines in the Command Line

Surfing on the Shell

There are literally thousands of search engines and at least as many input masks. Surfraw allows you to query the Web from the command line. Which may be just what you need when installing a server.

BY ANDREAS KNEIB

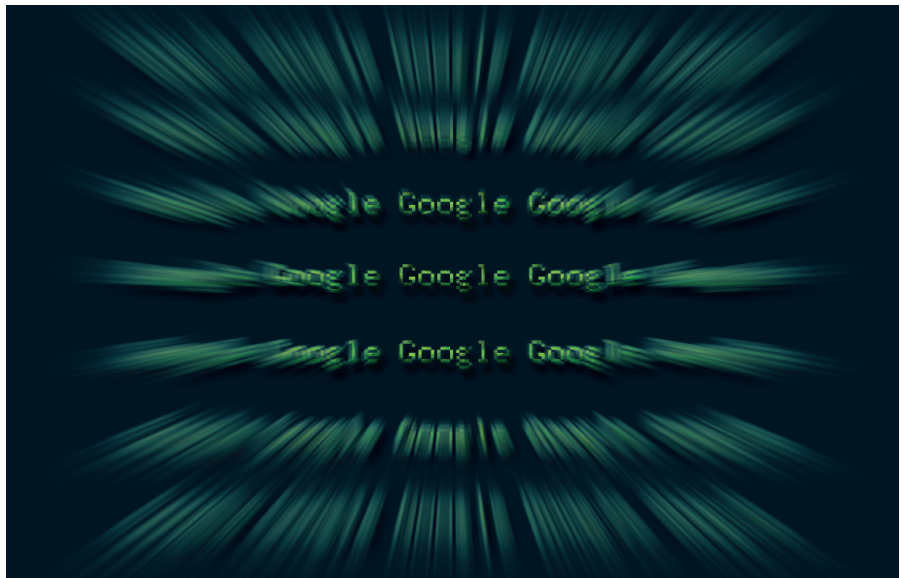
Users looking for information on the Internet will be familiar with this problem: you jump from Google via Freshmeat to Altavista, click your way through Web forms and end up with a stack of browser windows on your desktop. It is far less nerve-racking to use *surfraw*. The tool uses your search key to formulate a query and send it to your favorite search engine. The results are then displayed in a browser window.

Installation

If the program is not included with your distribution, a tarball with the source code is available from the project homepage at [1], and also to be found on this month's subscription CD. Start by unpacking the *tar* archive, before you follow the standard procedure, *configure*, *make*, and *make install* to install *surfraw*:

```
[andreas]~> tar<$$$>
xzf surfraw-1.0.7.tar.gz
[andreas]~> cd surfraw-1.0.7
[andreas]~> ./configure
[andreas]~> make
[andreas]~> su
Password: (root-Password)
# make install
```

The *INSTALL* file contains additional installation options; the file will be located in the new *surfraw-1.0.7* directory. Debian users can simply type:



```
apt-get install surfraw
```

to install the tool on their computers.

Configuration

After installing the tool, the */etc/surfraw.conf* and */etc/surfraw_elvi.list* files should be available in your */etc* directory.

Take a look at */etc/surfraw.conf* first, as this is the global configuration file. The defaults are fairly useful and the file is well documented. The header contains a note on the configuration syntax and a warning: "This file has a different format to that of *\$HOME/.surfraw.conf*! *\$HOME/.surfraw.conf* entries should always be written sh style."

In other words the syntax for the main configuration file in */etc/surfraw.conf* is not the same as *~/.surfraw.conf* in your home directory. Whereas system global configuration variables are defined using *def* and *defyn*, local preferences will follow the *VARIABLE = value* pattern.

Keeping this in mind, you can now go on to create a configuration file, *~/.surfraw.conf*, in your home directory. The personal file has the advantage of allowing you to make quick changes without assuming root privileges. It additionally provides for individual user profiles.

The following variables are of interest at this stage: *SURFRRAW_text_browser* and *SURFRRAW_graphical_browser*. Launch your favorite editor and select your favorite character based and GUI based browsers:

```
SURFRRAW_text_browser=lynx
SURFRRAW_graphical_browser=☛
phoenix
```

Now save your entries as *~/.surfraw.conf*. Congratulations, you have already completed a major part of the configuration steps. The settings in our example select Lynx as a character based browser and Phoenix as a GUI based browser.

Options

Now let's look at the second file, which is called */etc/surfraw_elvi.list*. This file contains a list of accessible search engines. For each search engine a shell script of the same name is available in */usr/local/bin* or */usr/bin*.

Calling a script in the command line with the *-help* flag set will list the parameters available to the script (*global options*). Some scripts, such as *google*, also have special *local options* (Figure 1).

The *global options* reflect a few variables from */etc/surfraw.conf*. The first

```
wfmm
Usage: google [options] [search words]...
Description:
  Surfraw search the web using Google (www.google.com)
Local options:
  -results=NUM          Number of search results returned
                        Default: 30
                        Environment: SURFRAW_google_results
                        Specialized search on topic
  -search=              BSD
                        Linux
                        Apple Macintosh
                        U.S. Government
                        Environment: SURFRAW_google_search
Global options:
  -browser=EXECUTABLE  Set browser
                        Default: opera
                        List Surfraw mechanisms for conquering evil
  -elvi                 Apply url escaping to arguments
                        Default: yes
                        Environment: SURFRAW_escape_url_args
  -escape-url-args=yes|no Get some windowed sin
                        Default: yes
                        Environment: SURFRAW_graphical
  -g | -graphical       Yes, yes, harder, deeper, faster, oh baybe
                        Default: yes
                        Environment: SURFRAW_p0rn
  -p0rn=yes|no          What you're reading now, dude
                        I can't enjoy it with you talking all the time
                        Default: no
                        Environment: SURFRAW_quiet
  -help
  -quiet=yes|no
```

Figure 1: The “google” script displays the available parameters when called with the “-help” option

item `-browser=` calls one of the browsers you selected – independently of the preferences you specified in `~/surfraw.conf`. `-elvi` allows you to access the list of search engines / shell scripts. The `-g` flag launches the GUI Web browser, assuming you have not disabled the browser using an `< SURFRAW_graphical=no` entry in `~/surfraw.conf`.

`t` is the counterpart to the `-g` flag, and will launch a small footprint text browser in an Xterm instead of a bloated GUI monster. The argument `-escape-url-args=` controls character masking in URLs for shell metacharacters.

The *local options* differ from script to script. Let’s again use *google* as an example.

The `-results=` parameter allows you to define the number of results shown for a hit.

The `-search=` flag switches the search mode to topic mode, displaying Web sites with information on BSD (*bsd*), Linux (*linux*), Apple Macintosh (*mac*), and the US Government (*unclesam*). Thus typing the following command:

```
google -results=100 -search=?
unclesam linux tcpa
```

Fig. 2: The “sourceforge mutt” argument tells *surfraw* to search Sourceforge.net for the Mutt mailer

will display what the Web Forum of the National Security Agency (NSA) has to say on the subject of the *linux* and *tcpa* for the first hundred hits for these words.

Do-It-Yourself

Of course it makes sense to customize a script based tool to suit your own requirements, although 40 search engines is no mean offering. But if your favorite search engine happens to be missing, you can create your own *Elvi* (that is a search script). The only thing you need, is some knowledge of shell programming. The *surfraw* tarball also contains a file *HACKING* with a guide to building *Elvi* files.

If this sound too much like hard work to your, why not use the script in Listing 1 as a template? You can use the template to rummage through the Linux Magazine database for tips in online articles. To save you time typing the script, it is available on the subscription CD. Copy the file to the `/usr/local/bin` directory and make it writable to *root* and executable for all users:

```
root # chmod 755 /usr/local/bin?
/linuxmagazine
```

And do have lots of fun on your command line based trip through the World Wide Web. ■

Listing 1: Simple template

```
#!/bin/sh
# $Id: linuxmagazine,v 1.0 2003/06/08 23:18:14 proff Exp $
# elvis: linuxmagazine -- Look up word in Linux Magazine
. surfraw || exit 1
w3_usage_hook () {
    cat <<EOF
Usage: $w3_argv0 [options] [search words]...
Description:
  Look up word in Linux Magazine (www.linux-magazine.com)
EOF
    w3_global_usage
}
w3_config
w3_parse_args "$@"
# w3_args now contains a list of arguments
if test -z "$w3_args"; then
    w3_browse_url "http://www.linux-magazine.com/search"
else
    escaped_args=`w3_url_of_arg $w3_args`
    w3_browse_url "http://www.linux-
magazine.com/search?config=lu&restrict=&exclude=&?
method=and&format=builtin-long&sort=score&words=${escaped_args}"
fi
```

INFO

[1] *surfraw* – Shell Users’ Revolutionary Front Against the World wide web: <http://surfraw.sourceforge.net/>