Webmaster Tools and Knowledge

April 25, 2006
Harvard University
Division of Continuing Education
Extension School
Course Web Site: http://cscie12.dce.harvard.edu/
Copyright ©1998-2006 David P. Heitmeyer
My email: david_heitmeyer@harvard.edu
Course staff email: cscie12@fas.harvard.edu

Webmaster Tools

- Site Icons ('favicon.ico')
- Web Robots
- Link Checking
- Search Robots
- Other Webmaster Tools
  - HTML/CSS Validation
  - Accessibility Compliance
  - Web Key Performance
  - Generating HTML for other devices
  - XHTML Page Performance

Site Icons

- favicon.ico at root of web site
- 'link' element in XHTML/HTML document
- Firefox uses favicon.ico or link element, rel="icon", in the location bar, bookmark list and tab display.

The code in the 'head' of the XHTML would look something like:
<link rel="icon" href="images/mozilla-16.png" type="image/png">
<link rel="shortcut icon" href="images/mozilla.ico" type="image/x-icon">

Make your site ready for search engines

- well-formed (and hopefully valid) HTML/XHTML
- use mark-up language for headings and lists
- titles that stand on their own
- "meta" keywords and description

An example using O'Reilly OnLamp.com
In "head" element of page:
<meta name="keywords" content="ONLamp.com,O'Reilly Network,oreillynet,oreillynet.com,O'Reilly,OREILLY,o'reilly network,o'reilly,onlamp.com,lamp,lampp,linux,apache,mysql,perl,python,php,linux,bsd,web development,server development reference,technical information,open source" />
<meta name="description" content="Welcome to ONLamp.com, the high performance web development site from the O'Reilly Network offering comprehensive Lamp developer information and resources. O'Reilly Network's ONLamp site features original articles, news and commentary." />

Google results for a search engine query:

```
Results 1-3

1. http://www.google.com/search?q=site%3Aexample.com
2. http://www.example.com/search?q=site%3Aexample.com
```

The code in the 'head' of the XHTML would look something like:
```
<meta name="robots" content="all">
```

The code in the 'head' of the XHTML would look something like:
```
<meta name="robots" content="all">
```
Firefox as a Web Development Tool

Web Developer Extension
Firefox Extension - Live HTTP Headers

xurl and churl

xurl
A simple Perl script that extract the links for a single page. Adapted from The Perl Cookbook.

minerva%
`xurl URL`

churl
A simple Perl script that will check the links for a single page. Adapted from The Perl Cookbook.

minerva%
`churl URL`

xurl (example)
`xurl http://www.extension.harvard.edu/`

http://www.extension.harvard.edu/2005-06/images/snaps.jpg
http://www.extension.harvard.edu/2005-06/images/veri.gif
https://dceweb.harvard.edu/prod/gowlogn3.taf
javascript:popUp('/2005-06/snapshots/default.jsp')
javascript:popUp2('/2005-06/profiles/default.jsp?n=3')
mailto:webmaster@hudce.harvard.edu

churl (example)
`churl http://www.extension.harvard.edu/`

http://www.extension.harvard.edu/: 200 OK
http://dceweb.harvard.edu/prod/sowxcrq.taf?school=EXT: 200 OK
... http://www.extension.harvard.edu/2005-06/images/go2.jpg: 200 OK
... http://www.thecrimson.com/article.aspx?ref=512654: SKIP
https://dceweb.harvard.edu/prod/gowlogn3.taf: SKIP
javascript:popUp('/2005-06/snapshots/default.jsp'): SKIP
javascript:popUp2('/2005-06/profiles/default.jsp?n=3'): SKIP
mailto:webmaster@hudce.harvard.edu
timefetch


- a Number of attempts for the second page fetch.
- b Minimum number of broken images to trigger alarm.
- d Debug: view all kinds of marginally useful output.
- f Force host: before doing recursive downloads, munge each URL and replace the host in the URL with some other host.
- h Help: print this help message.
- j Java: download java applets as well.
- F No frames: If the page is a frameset, do not fetch the frames.
- X Don't exit on errors, just try to continue.
- z Exit immediately on errors in fetching the main page.

NOTE: This program always downloads embedded frames and prints a cumulative total for frames and framesets, even if you did not specify a recursive download.

timefetch examples

timefetch is in need of updating, but can still be a useful tool. timefetch will show the actual download time (often not useful) and the total kilobytes downloaded (often useful). Warning: timefetch will not execute JavaScript, nor does it fetch images included by CSS.

minerva$ timefetch http://www.harvard.edu/
OK 0.044  20.6kb: http://www.harvard.edu/

Recursive (-r) and verbose (-v):
minerva$ timefetch -rv http://www.harvard.edu/  0.086   1.6kb: http://www.harvard.edu/images/global/shield1.gif 0.007   ... 0.025  20.4kb: http://www.harvard.edu/ 0.828 234.7kb: http://www.harvard.edu/ (incl.: img)

Web Robots

Robots, Spiders, Crawlers

As they "spider" a site, the robots can perform various actions, such as:
- Gathering content for search engines or a website mirror
- Validating, checking, or processing content

Spidering Behavior: an example with Lynx

minerva%
lynx -help
minerva%
lynx -traversal ?

-crawl ?

-realm cscie12.dce.harvard.edu

http://cscie12.dce.harvard.edu/lecture_notes/20050406/

the "lnkNNNNNNNN.dat" files contain the text dump of the pages lynx retrieved
the "traverse.dat" files contain the list of links that lynx retrieved
the "reject.dat" files contain a list of URLs that lynx did not fetch (due to the fact that they are outside the "realm" as specified on the command line).

minerva%
ls -l

total 339
... traverse.dat-rw-------   1 cscie12  courses     14318 Apr 20 17:43 traverse2.dat
Robots Exclusion Standard (RES)

The Web Robots Pages
http://www.robotstxt.org/wc/robots.html

RES provides two mechanisms to instruct robots that visit your site:
1. robots.txt file
2. robots meta tag

Two directives:
1. User-Agent
2.Disallow

Note: robots.txt must lie at the root level of the server.

Examples of robots.txt files:
http://www.fas.harvard.edu/robots.txt
http://www.foxnews.com/robots.txt
find them at a couple of your favorite web sites

Why won’t the following robots.txt files do anything useful? (they aren’t at the root level of server)
http://www.people.fas.harvard.edu/~jharvard/robots.txt
http://www.fas.harvard.edu/computing/robots.txt

Robots meta element

The Robots meta element can be used on a per-document basis.
OK to index page; OK to follow links on page
<meta name="robots" content="index,follow"/>
OK to index page; Don’t follow links on page
<meta name="robots" content="index,nofollow"/>
Don’t index page; OK to follow links on page
<meta name="robots" content="noindex,follow"/>
Don’t index page; Don’t follow links on page
<meta name="robots" content="noindex,nofollow"/>

robots.txt Examples

Disallow all robots from certain areas:
User-agent: *
Disallow: /cgi-bin
Disallow: /cgi-lib
Disallow: /ADMIN
Disallow: /test
Disallow: /computing/webstatistics/

Disallow all robots except the Harvard search engine (identified by the User-Agent string of "Harvard Homepage search engine (Ultraseek)"):
# no robots, please!
User-agent: Harvard Homepage search engine (Ultraseek)
Disallow: User-agent: *
Disallow: /
Link Checking Robots

- Check the links on a single page or on an entire site.
- If following links, will do a GET request; otherwise it should do a HEAD request.

Examples of Link Checking Robots

- churl_checklink
- webbot
- checkbot
- webcheck
- lynx

W3C Link Checker

http://validator.w3.org/checklink

Use Online:

http://validator.w3.org/checklink

Use command line:

minerva%

checklink URL

Perl, Free

checklink

minerva%

checklink --help

W3C checklink version 3.6.2.26 (c) 1999-2004 W3C

Usage: checklink <options> <uris>

Options:
- `--summary` Result summary only.
- `--broken` Show only the broken links, not all.
- `--noheader` Don't use HTTP headers in HTML output.
- `--nohead` Don't use HTML headers.

See "perldoc Net::FTP" for information about various environment variables affecting FTP connections and "perldoc Net::NNTP" for setting a default NNTP server for news: URIs.

The W3C_CHECKLINK_CFG environment variable can be used to set the configuration file to use. See details in the full manual page, it can be displayed with:

   perldoc /usr/local/bin/checklink

More documentation at: http://www.w3.org/2000/07/checklink

Please send bug reports and comments to the www-validator mailing list:

   www-validator@w3.org (with 'checklink' in the subject)

Archives are at: http://lists.w3.org/Archives/Public/www-validator/
Checkbot Example

```
minerva%  checkbot
> --url http://cscie12.dce.harvard.edu/lecture_notes/20060131/
> --verbose
```

Results are given in an HTML page: checkbot.html

```
Commercial
Watchfire has several solutions in the area of website quality assurance (links, accessibility, etc.)
```

```
For the Programmer: Writing Your Own
The Perl modules, LWP and WWW::Robot make writing robots almost trivial.
```

Examples in Perl Cookbook, published by O'Reilly
Perl and LWP by Sean Burke, published by O'Reilly
Other Webmaster Tools

- W3C HTML Validation, http://validator.w3.org/
- W3C CSS Validation, http://jigsaw.w3.org/css-validator/
- WebXact (WAI and Section 508 Compliance), http://webxact.watchfire.com/
- Watchfire.

Checking HTML Pages

- w3mir, http://langfeldt.net/w3mir/
w3mir

w3mir is an all-purpose HTTP copying and mirroring tool. The main focus of w3mir is to create and maintain a browsable copy of one, or several, remote WWW sites. Used to the max w3mir can retrieve the contents of several related sites and have the mirror browsable via a local web server, or from a filesystem, such as directly from a CDROM.

HTTP Server Stress Test

ApacheBench (ab)

minerva% /usr/sbin/ab -h
Apache JMeter
http://jakarta.apache.org/jmeter/index.html
httperf
---A Tool for Measuring Web Server Performance

Apaca Bench (ab)

This is ApacheBench, Version 1.3d <$Revision: 1.67 $> apache-1.3
Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net

Benchmarking cscie12.dce.harvard.edu (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Finished 10000 requests

Server Software: Apache/2.0.49
Server Hostname: cscie12.dce.harvard.edu
Server Port: 80
Document Path: /
Document Length: 21765 bytes
Concurrency Level: 10
Time taken for tests: 6.291670 seconds
Complete requests: 10000
Failed requests: 0
Write errors: 0
Total transferred: 220437059 bytes
HTML transferred: 0 bytes
Requests per second: 1603.38 [max 199.39, mean 199.39]
Transfer rate: 34215.08 [Kbytes/sec] received

Connection Times (ms):
min   mean[+/-sd] median   max
Connect:       0    1   0.9      1       8
Processing:     0    4   1.6      4      49
Waiting:        0    1   1.5      1      47
Total:          1    5   1.7      6      51

Percentage of the requests served within a certain time (ms):
50%      6  66%      6  75%      6  80%      7  90%      7  95%      8  98%      9  99%     10 100%     51 (longest request)

Apaca Bench (ab)

This is ApacheBench, Version 1.3d <$Revision: 1.67 $> apache-1.3
Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net

Benchmarking cscie12.dce.harvard.edu (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Finished 1000 requests

Server Software: Apache/2.0.49
Server Hostname: cscie12.dce.harvard.edu
Server Port: 80
Document Path: /tools/webcube.cgi
Document Length: 58163 bytes
Concurrency Level: 10
Time taken for tests: 98.986587 seconds
Complete requests: 1000
Failed requests: 0
Write errors: 0
Total transferred: 58323402 bytes
HTML transferred: 58171098 bytes
Requests per second: 100.59 [max 217.76, mean 100.59]
Transfer rate: 575.39 [Kbytes/sec] received

Connection Times (ms):
min   mean[+/-sd] median   max
Connect:        0     0   0.1      0       1
Processing:     191  981 966.2    719    9244
Waiting:      188  810 750.6    586    6321
Total:        191  981 966.2    719    9244

Percentage of the requests served within a certain time (ms):
50%   719  66%  1065  75%  1278  80%  1468  90%  1956  95%  2452  98%  4006  99%  5499 100%  9244 (longest request)
HTTP Server Monitoring

**Nagios**

http://www.nagios.org/

**Cricket**

http://cricket.sourceforge.net/

**Monitoring Apache with Cricket**

Web Logs

**Web Logs**

**Common Tools**

- Analog
- Analog + Report Magic
- AWStats
- WebTrends

Data in Access Logs

**Data in Access Logs**

- Time
- IP address / Hostname
- Username (if under Authentication)
- Request
- User-Agent
- Referrer URL
- Response Status
- Bytes returned

Possible Data:

- The contents of a specified environment-variable
- The request protocol
- The context of trusted and untrusted SSL certificates
- Remote hostname (from header, if available)
- The request method
- The contents of the server非常好的query
- The remote IP address
- The server name according to the Hostname extension

HTTP Server Logs

**HTTP Server Logs**

- access log
- error log (no longer common)
- referer log (no longer common)
- user-agent log (no longer common)

**access log**

- [Thu Dec  2 11:26:08 1999] [notice] Apache/1.3.9 (Unix) configured -- resuming normal operations
- [Thu Dec  2 11:27:19 1999] [notice] caught SIGTERM, shutting down
- [Mon Dec  6 19:15:04 1999] [notice] Apache/1.3.9 (Unix) configured -- resuming normal operations
- [Mon Dec  6 19:27:33 1999] [notice] caught SIGTERM, shutting down

**error log**

- [Wed Nov 13 11:21:29 1999] [error] Apache/1.3.9 (Unix) configured -- resuming normal operations
- [Wed Nov 13 11:21:29 1999] [error] caught SIGTERM, shutting down
- [Wed Nov 13 11:21:29 1999] [error] caught SIGTERM, shutting down

Typical Data:

- Time
- IP address / Hostname
- Username (if under Authentication)
- Request
- User-Agent
- Referrer URL
- Response Status
- Bytes returned

Possible Data:

- The contents of a specified environment-variable
- The request protocol
- The context of trusted and untrusted SSL certificates
- Remote hostname (from header, if available)
- The request method
- The contents of the server非常好的query
- The remote IP address
- The server name according to the Hostname extension
Log Formats

- Common Log Format (CLF):
  - host: client host
  - ident: client identification
  - auth_user: authenticated user
  - date: date and time
  - request: requested URL
  - status: HTTP status code
  - bytes: number of bytes transferred
- User-Agent Log:
  - date: date
  - user-agent: user agent
- Referer Log:
  - date: date
  - referrer-url: referrer URL
  - request-url: requested URL
- Combined Log Format:
  - host: client host
  - ident: client identification
  - auth_user: authenticated user
  - date: date and time
  - request: requested URL
  - status: HTTP status code
  - bytes: number of bytes transferred
  - referrer: referrer URL
  - user-agent: user agent

Custom Log Formats in Apache
## Complicating Issues

- HTTP is a stateless protocol
- Local Cache
- Proxy Cache
- Proxy Servers
- Shared Computers

## Log Rotation

- approximately 200 to 250 bytes per line (request)
- For example, 1,000,000 requests per day (12 requests per second)
- log grows at 2.8 kb per second
- 238 Mb for 1,000,000 requests
- compressed (gzip'ed) logs are 7 to 10% of original size!

## Tools for Log Analysis

- Analog
  - http://www.analog.cx/
  - Stephen Turner
  - UNIX, Windows, MacOS, others
  - Free!!
- Report Magic
- WebTrends Log Analyzer
  - http://www.webtrends.com/