Alkaline: a UNIX/NT Search Engine

Alkaline 1.9 Users Guide

Vestris Inc., Switzerland

Alkaline: a UNIX/NT Search Engine: Alkaline 1.9 Users Guide

by Vestris Inc., Switzerland

Published February 2002

Table of Contents

List of Tables

List of Figures

Chapter 1. Alkaline Concepts

1.1. What is Alkaline?

The Alkaline Search Engine is an all-in-one index and search server. Adding a search engine to your web site eases navigation and helps your customers locate exactly the information they are looking for in seconds.

It is, on one hand, a stand-alone web spider that can collect data from local and remote sites following a precise set of rules and using a variety of options defined by the search engine administrator. On the other hand, Alkaline offers users, navigating through a web site, a fast and relevant tool for searching hundreds of thousands indexed documents in a blink of an eye.

Alkaline is a spider. The search engine administrator defines the initial url(s), which Alkaline should index, and a set of rules such as the maximum indexing depth or links not to follow or to exclude. Alkaline will retrieve such an url, parse the document, index it’s content, and extract all links available from the HTML code. It will then follow these links and restart the same process for each new page. This allows to index a full web site, a subset or a specific set of individual documents.

Alkaline is a searcher. The webmaster can incorporate a search box with a set of advanced options to any page on the site. Users can enter words to search for. A whole set of advanced rules and options is available. Alkaline will then output results using a search template defined by the search engine administrator. These results will have the look and feel of the entire site.

1.2. Search Engines Overview

There are basically three types of search engines.

Search Scripts are written generally in Perl or C searching sites of a maximum a few thousand pages. They do not include complex algorithms to optimize searching and indexing and become unusable as the site grows over a few thousand pages. Such scripts are still good when it comes to a home-made site, where a search box is more of a gadget.

Technically, those scripts index modified pages once in a while (like once a day) in a few minutes or seconds and produce usually fixed output with little layout options as search results. As those scripts are entirely CGIs, they are slower as they never maintain persistent data in memory but store it and load or parse it each time the search engine is invoked.

Among such scripts you can find WebGlimpse (http://www.webglimpse.net/) or ht://Dig (http://www.htdig.org/).

Search Servers, ISAPI or WAI applications, sometimes mixed with CGI scripts overcome this drawback of indexes constantly re-read from the hard disk. Working as a permanently running server and answering to multiple search requests simultaneously this category of search engines requires more hardware power, more memory and is aimed to larger sites that really need a good search engine. Alkaline is designed as a server persistent search engine.

Technically, search servers maintain indexes in RAM or use some internal swap mechanism. They have complex algorithms for searching and indexing, usually jealously kept secret by their designers. Alkaline uses the concept of "cellular expansion" which gives quite an interesting performance and opens doors for future research. Cells are fast and resistant to growing data. Of course, there’s no mystery that a big server with a lot of hardware power will search faster and will be able to index a larger site. Existing Alkaline powered sites maintain an index of 500’000 pages with about 450’000 word forms and run on industry average Pentium III or Sun Ultra servers. Such a configuration can handle from two to three search requests per second.

Among such servers you can of course find Alkaline, but also Infoseek Ultraseek (http://www.ultraseek.com/ or Thunderstone Webinator (http://www.thunderstone.com/webinator/).

Finally, Distributed Servers target searching and indexing of the whole web. This is the most fierce long term fight of search engines as large companies compete for the best technology and for the most relevant search results. We plan a parallel implementation of Alkaline for a cluster over a TCP/IP platform independent network and for IBM SP2. We have already made numerous tests over a PVM network. For our distributed architecture we want Alkaline to index 5-10 million pages running fast on a cluster of 32 PII PCs. Unlike Altavista we do not plan to set search limits to Alkaline depending on the price, that is we will distribute it as one single product for the same value no matter what you search. Choosing Alkaline, you will also choose a team that works for the future.

Technically, distributed search servers perform both parallel indexing and searching. More hardware power you have, faster indexing and searching is. Of course, this depends on the network charge overhead. All major search engines use distributed architectures and can hit hundreds of requests per second.

1.3. Alkaline Features

Alkaline can be viewed as two distinct pieces: the indexer or spider and the search engine. A non-exclusive and constantly growing capabilities include the lists below.

1.3.1. Indexing

- no theoretical limits in amount of indexed documents or sites
- fully remote indexing, not just local machine or local area network
- remote URL(s) defined as a base of indexing
- indexing of local file system
- local directories defined as a base of indexing
- true spider, follows links on web pages, A HREFs, MAPs, FRAMEs, META REFRESH, etc.
- deleted pages are automatically removed and newly created pages instantly added

- grouping of multiple sites with individual options and parameters inside a same search group
- automatic support for redirected URLs, relative Location: headers, detection of circular deep redirections
- multiple indexing bases for the same index/search database
- highly configurable index/search paths, exclusion lists, index categories and file extensions
- capable of using regular expressions to define which urls to follow and what documents to index
- setting file amount, recursion and remote limits on demand
- automatic indexing of newer files only, using if-Modified-Since
- intelligent HTML parsing, link and text retrieval, supporting &...; style tags, simple error recovery
- single indexing engine for multiple search/index groups
- foreground dedicated indexing for first-time setup or fast reindexing
- multithreaded architecture with background continuous indexing
- textual cleanup, supporting accentuated characters (searching French text with or without accents for example)
- META tag support for KEYWORDS and DESCRIPTION, TITLE tag support for title
- discarding of script, style and object code
- full support for robots.txt and META ROBOTS directives, disabled on demand
- filters for indexing other formats than HTML and plain text (such as Adobe PDF)
- using external third party command line tools as filters through a documented interface
- embedded objects retrieval support for indexing other formats such as Shockwave Flash using the filter interface

- page preprocessing available through a published API before real indexing, using a filter
- Md5 document signature that identifies and ignores symbolic links and duplicate documents (such as http://www.foo.com and http://www.foo.com/index.html)
- persistent remote document retrieval, fully configurable in number of retries, etc.
- supports retrieval of secured pages on password protected sites (HTTP/1.0 BASIC authentication, NTLM support for Windows NT, no support for SSL)
- Alkaline-specific META tags to avoid indexing of individual pages, following links, excluding text portions, indexing META data or indexing parts of a document
- using the Alkaline memory mapped files swap to minimize memory usage
- using the Alkaline flat interval technology to stabilize the memory usage curve
- external lists of words to be excluded from indexing, rules for page inclusion, stop words, including regular expressions to define exclusions, etc.
- statistics on requests and traffic
- capable of adding/removing/reindexing URLs submitted online
- native server-side includes (SSI)
- full support for client non-Javascript Cookies
- fully parallel multithread configurable retrieval, concurrent indexing
- ability to run as a native Windows NT/2000 service

1.3.2. Searching

- searching remote sites
- searching any search group with a single search/index server
- searching local file system
- searching of word sub-strings and heuristics, not just full keywords

- fully configurable output (virtually any HTML layout), using user-defined templates, with the MV4 expressions mechanism for each separate search group
- multiple page results, with any amount of results per page for each separate search group
- full web server pool architecture for immediate response at search
- denial of service, server flood protection and automatic fall-off, automatic restart on resource starvation
- searching of accentuated and non-accentuated text, full support for automatic translation of accents (é, à, etc.)
- searching in META tags
- output of META DESCRIPTION and page TITLE if available
- searching in ALT image and applet tags
- no searching in scripts
- automatic selection of case-sensitive/case-insensitive search
- automatic selection of heuristics/exact search for quoted sequences
- boolean search using + and - signs
- scope restriction to host, path, url and file extension
- results sorting by date (ascending and descending), size (ascending and descending), title and url
- results grouping by domain name
- results re-sorting and re-grouping by any of the above criteria
- four level expiring cache
- user-selection of maximum amount of results
- numeric tags, combinations such as *price=345* searchable as *price<34*, *price=34* or *price>345*
- ranking weight options for titles, meta tags and document body

- weak words
- support for GET and POST methods
- wap/wml 1.1 wireless devices support

1.3.3. Online Administration

- BASIC authentication restricted administration section with various access level username/password pairs
- fully customizable administration section, using JavaScript and XML
- extended possibilities for resellers for co-branding
- extensive search statistics and performance counters
- browsing of configurations and their individual parameters
- search 4-level cache statistics per configuration
- certification embedded in the admin section
- restart the server from the admin section
- refresh templates from the admin section
- add, reindex and remove individual urls from the admin section
- produce MRTG-compliant statistics through XML queries and plot search/load averages using MRTG

1.4. History of Alkaline

Alkaline was originally developed by Daniel Doubrovkine, at that time, student at the University of Geneva, founder of Vestris Inc., Switzerland. The author was faced the particular problem of installing a working search engine to the Swiss French TV website. All existing search engines failed to perform according to the customer's requirements.

Alkaline was born. It was rather slow but sufficient. Improvements were made on a daily basis and a new C++ base library was designed as the evolution of the MV4 C++ library originally used for Alkaline. It exposed strings and arrays, implemented the full HTTP/1.0 stack, threads, URLs, parsers, had a portable file system abstraction and many more facilities. Designed from scratch, it was consistently tested across various NT and UNIX systems. Alkaline 1.0 was able to spider properly but had rather poor indexing and search performance.

While working on an unrelated project, Hassan Sultan, also a student at the University of Geneva came up with an idea to build efficient cross-reference tables and navigate through the cells matrix with an O(n) complexity without any of the parallel processing techniques. This algorithm, called the cellular expansion algorithm was implemented in Alkaline 1.1. The search engine was slowly growing from a private lab project into a full scale commercial search engine.

Today, Alkaline is version 1.9. It is a strong search engine machine running on hundreds of web sites around the world.

1.5. History of the Name

The Alkaline Search Engine is designed for a broad variety of applications but does not intend to be the very best and the most powerful search engine ever made for all purposes. Our search engine is exactly the illustration of the competition on the battery market, that's why we have called it Alkaline.

The Duracell "Comprehensive Battery Guide" lists 133 of-the-shelf batteries with descriptions like zinc-carbon, alkaline manganese, lithium, mercury, silver, zinc-air and nickel cadmium. There are even subclasses, for example Li/FeS2, Li/MnO2, Li/SO2, Li/SOCl2 and "lithium solid state". And from other manufacturers you can get sealed lead-acid and gel-type batteries. For the truly exotic application you might even want to consider fuel cells or radioactive thermal generators. What are all these batteries? How do you choose what's best for your portable widget? If you check for existing search engines, you’ll have the same inquiries.

The alkaline manganese cell, generally sold as simply "alkaline" is a better still in high-current-discharge and low temperature operation. It is inside-out compared with zinc-carbon, having the powdered-zinc negative anode and potassium hydroxide electrolyte in the middle, surrounded by a manganese-dioxide-and-carbon outer positive cathode. Because of its particular chemistry, an alkaline battery maintains a low and slowly increasing internal resistance as it discharges, compared with the rapidly rising internal resistance of both types of zinc-carbon cells. It also works better at low temperatures. Alkaline batteries have a longer life as well.

Thus, we have wanted Alkaline to hold larger indexes and a fast growing site with "little internal resistance", that is with fast search results.

1.6. People Behind Alkaline

Daniel Doubrovkine, alias dB (http://www.dblock.org) is the developer of the Alkaline core code. He has founded Stolen Technologies Inc. in 1994, which became Vestris Inc. in 1997. He owns a B.Sc. from University of Geneva. Original co-founder of Xo3 SA, he quit the group and left for Redmond, Washington in the United States where he currently works for Microsoft Corporation MSN.

Hassan Sultan is responsible for Alkaline’s search capabilities. He graduated from University of Geneva in year 2000 and works now for Microsoft Corporation in the Windows NT Sustained Engineering group.

The BASE C++ library is the foundation of Alkaline. Although most of that code has been rewritten or altered, it is worth mentionning Serge Huber, who is the author of minor parts of the original BASE core.

Alkaline authors would also like to thank all the anonymous people that have contributed to the engine’s spirit, have spent a lot of time testing the engine or have simply been there from the very first Alkaline’s steps.

1.7. Copyright Notices

Alkaline is Copyright 1994-2000 Vestris Inc., Trident Chambers, P.O. Box 146, Wickhams Cay, Road Town, Tortola, British Virgin Islands.

Vestris Inc. gratefully acknowledges the use of the following third-party components:

- MD5 Message-Digest Algorithm, Copyright © RSA Data Security, Inc.
- Regular Expression Engine, Copyright © 1986, 1993, 1995 by University of Toronto, written by Henry Spencer.
- Simple XML Processing Library, JavaScript, Copyright © 1998 Jeremie
- Morten's JavaScript Tree Menu, Copyright © 1998-2000 Morten Wang, TreeMenu.com

The documentation and the frequently asked questions are written using XEmacs in the SGML format. Both are converted to HTML and PDF using SgmlTools 2.0.2, which is a bundle of DocBook DTD, DSSSL, Jade, TeX and Python scripts. Additional formatting and processing is done using Awk scripts by Marc Vuilleumier-Stuckelberg and David Clerc from Rembo Technology Sarl.

- SgmlTools is Copyright © by Cees de Groot
- DocBook is an SGML DTD maintained by the DocBook Technical Committee of Oasis
- Python is Copyright © 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
- GNU Awk is Copyright © 1989, 1991-1997 Free Software Foundation.
- TeX is Copyright © 1999 D.E. Knuth.
- Kpathsea is copyright © 1999 Free Software Foundation, Inc.
- Jade is a DSSSL implementation by James Clark
- Web2c is written by Karl Berry, maintained by Olaf Weber

- Document Style Semantics and Specification Language, DSSSL, is an ISO standard for formatting SGML and XML documents
- OpenJade is a project undertaken by the DSSSL community to maintain and extend Jade.

Vestris Inc. also acknowledges all registered trademarks referred to in this documentation:

- Microsoft, MS, MS-DOS, Win32, Windows and Windows NT are registered trademarks of Microsoft Corporation.
- Linux is a registered trademark of Linus Torvalds.
- OS/2 is a registered trademark of International Business Machines Corporation.
- Intel and Pentium are registered trademarks of Intel Corporation.
- Unicode is a registered trademark of Unicode, Inc.
- TrueType is a registered trademark of Apple Computer, Inc.
- Arial is a registered trademark of The Monotype Corporation.
- UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.
- All other trademarks are the property of their respective owners.

Chapter 2. Alkaline Server Installation

2.1. Concepts and Requirements

Alkaline is a full HTTP/1.0 compliant server. It is not a CGI-compliant application, it is not an ISAPI/WAI filter or extension.

To successfully install and run Alkaline, a compatible operating system is required. Currently Alkaline is supported under all flavors of Linux (x86 and Alpha), Sun Solaris (Sparc and x86), FreeBSD, SGI IRIX and more. Alkaline is also built for Windows NT and takes advantage of recent Windows 2000 improvements. Alkaline is not supported under Windows 95 or Windows 98.

Alkaline is not a CGI and a cgi-bin directory to which you can upload executables and run then via a web request is not sufficient to install the server.

2.1.1. Windows NT/2000

Under Windows NT, a real access to the machine is required, such as you are able to open a command prompt and execute programs. On Windows 2000, a terminal server or rclient access is sufficient. Ability to install and run services is required to target Alkaline for deployment in production environments. IIS or other web server is absolutely not required.

2.1.2. UNIX Operating Systems

On UNIX, a telnet or ssh access is required. It is absolutely not required to have root access to the machine; a normal user is sufficient. Apache or other web server is absolutely not required.

2.1.3. Hardware Requirements

Alkaline runs on any 32bit hardware running one of the supported operating systems, including Sun SPARC Family Workstation or Server, Dec Alpha, or Intel compatible PC with a Pentium class processor. Exclusively 64bit processors are not supported although Alkaline will run on some 64bit natively capable machines.

As the cheapest hardware configuration, we advise running Alkaline under a UNIX platform rather than a Windows NT 4.0 or below. If you are choosing a server, a Pentium III Linux machine with over 128MB RAM will be a fair choice. Most Linux versions, such as Redhat 6.0, are known to be good at running Alkaline.

For Windows NT based configurations, a Windows 2000 server with 256MB RAM is an excellent choice. We have extensively tested Alkaline with Windows 2000 multiprocessor machines and have been pleasantly surprised. New versions of Alkaline (since March 28th, 2000) use a different memory manager under NT which helps avoid heap fragmentation. Performance and stability are both better than on equivalent Sun machines.

Ultimately, for intensive commercial needs, choose a Sparc Ultra Server running Solaris or a Windows 2000 multiprocessor server.

Supported operating systems currently are (this might differ slightly depending on hardware availability to us, which customers provide on a voluntary basis):

- Sun Solaris 2.5 and above / SunOS 5.5.x and above
- Windows NT 3.5 or 4.0
- Windows 2000 Workstation or Server
- Slackware, Redhat, Mandrake, Debian or Suse Linux x86 Glib or Libc
- Redhat 6.0 or Suse 6.1 Linux Alpha
- SGI IRIX 6.5 and above
- FreeBSD 2.x and above

Physical Memory requirements vary depending on the size of the index you plan to have.

- 1’000 pages / 100’000 different word forms: ~4 MB
- 10’000 pages / 200’000 different word forms: ~32 MB
- 50’000 pages / 250’000 different word forms: ~64 MB
- 50’000 pages / 450’000 different word forms: ~128 MB
- 500’000 pages / 500’000 different word forms: ~256 MB

These numbers should not be taken as absolute. Alkaline has a built-in swap which can allow it to function under tight memory constraints. For example, on a machine with only 32 MB of RAM, you can very well load and search Alkaline’s half a million pages index. If your machine has a lot of memory, you might see much higher memory usage numbers because Alkaline will always attempt to page as little as possible.

There are no special requirements for swap space, having as much physical memory as possible is strongly recommended and performance will considerably degrade with more swap space and less RAM. About 25 MB of disk space per 50’000 pages and 200’000 different word forms is required when running with the native swap enabled. About 10-20% of physical memory usage will be used by Alkaline’s private swap for temporary files in this case.

Any standard TCP/IP network is required and absolutely no web server, such as IIS or Apache, is required.

2.2. Installation on UNIX

2.2.1. Download Alkaline

Identify your operating system. Connect to your server using a telnet or an ssh session. Login as usual as a normal user. It is not necessary to have, and is not advised to use, a root access to install Alkaline. You can run the *uname -a* command which will produce something like:

```
server:~/$ uname -a
Linux ns 2.0.36 #2 Thu Nov 19 13:41:52 MET 1998 i686 unknown
```

This above operating system is obviously Linux, running on a Pentium class processor (i686).

Download an Alkaline release for your platform. Navigate to http://www.vestris.com/files/asearch-distrib/ and choose a binary distribution. You are also free to choose an Alkaline beta version, which always has the latest improvements from the live source tree.

> 💣 Although we do our best to make sure that beta versions are stable and tested, they might contain debug code, produce unexpected results or behavior specific for the beta release.

Alkaline is also available via anonymous ftp from ftp://ftp.vestris.com/pub/alkaline/distrib/.

2.2.2. Install and Test Binary

Gunzip and untar the downloaded distribution. Use *gunzip*, then *tar vfx*.

```
big-server:~/$ gunzip /tmp/asearch.1.9.Linux-x86.tar.gz
big-server:~/$ tar vfx /tmp/asearch.1.9.Linux-x86.tar
asearch.1.9/CONTRIBUTIONS
asearch.1.9/COPYRIGHT
asearch.1.9/README
asearch.1.9/asearch.Linux
asearch.1.9/admin/ ...
big-server:~/$
big-server:~/$ ls -la
total 3
drwxr-xr-x   3 dblock   users 1024 Jun  5 08:16 ./
```

```
drwx--r-x  17 dblock   users 1024 Jun  5 08:14 ../
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:16 asearch.1.9/
big-server:~/$
```

Change your current directory to asearch.1.9 and test the binary, run ./asearch.OS, for example asearch.Linux on a Linux system. Use *ls -la* to find out the binary name.

```
big-server:~/asearch.1.9$ ./asearch.Linux
==================================================
Alkaline Search Engine, Version 1.9 (May 28 2000) for Linux
(c) Vestris Inc., Switzerland - 1994-2002 - All Rights Reserved
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
==================================================
    usage: asearch [options] port path [path2 ...]
           asearch [options] path-to-asearch.cnf command ...
     port: port to bind and listen to, ex: 9999
     path: relative path(s) to asearch.cnf files
  options: please refer to the users guide
  command: one of reindex, email, emailall, remove, rx-
match, rxrepl, etc.
_______________________________________________________________
        for more information, please re-
fer to http://alkaline.vestris.com
big-server:~/asearch.1.9$
```

On Linux, Alkaline is statically linked, so you do not need any of the pthread shared libraries. On other UNIX flavors, Alkaline might complain about missing a library. See the troubleshooting section for more details.

2.2.3. Run the Demo

Alkaline is now ready for a quick demo. Run

```
big-server:~/asearch.1.9$ ./asearch.Linux 9999 demos/docs
==================================================
Alkaline Search Engine, Version 1.9 (May 28 2000) for Linux
(c) Vestris Inc., Switzerland - 1994-2002 - All Rights Reserved
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
==================================================
[checking post 1.3 configuration] {
  [verifying that (demos/docs/asearch.cnf) exists]
  [file exists (61 bytes)]
[loading global.cnf] {
  [Redirect=/admin/]
  [AdminPath=admin]
  [DocumentPath=demos/images,docs,faqs]
} loaded 31 bytes.
[warning, you should define at least a root pass-
word in global.cnf]
[Alkaline server (Apr  2 2002) running, binding to 9999]
[loading /home/asearch.1.7/demos/docs/asearch.cnf] {
  [UrlList=file:///docs/,file:///faqs]
  [Index.html=index.html]
  [RegExp][enabled for 0 element(s)]
} loaded 1 inline configuration.
[newly added to Alkaline list: Demos/docs ]
[reloading Demos/docs]
[loading indexes] {
  [no index files found (clean index)]
} done.
[indexing thread running - lazy mode]
[01:02:36 2002-04-02][building index in Demos/docs]
[01:02:56 2002-04-02][done building index in Demos/docs]
```

```
[01:02:57 2002-04-02][(re)indexing of Demos/docs pro-
cessed 292 new/modified files in 0:21 min]
```

While the engine is running, navigate to *http://localhost:9999/* with your web browser. The admin section of your search engine server will appear. You can select the documentation configuration under *Server Parameters* -> *Server Configurations* -> *demos/docs* and perform searches on it.

2.2.4. Create a Simple Configuration

Download the sample configuration files from http://alkaline.vestris.com/install/install-sample.tar.gz and gunzip/untar them to the installation directory.

```
big-server:~/asearch.1.9$ gunzip /tmp/install-sample.tar.gz
big-server:~/asearch.1.9$ tar vfx /tmp/install-sample.tar
vestris/asearch.cnf
vestris/search.html
dicos/english.txt
big-server:~/asearch.1.9$
big-server:~/asearch.1.9$ ls -la
total 6
drwxr-xr-x   6 dblock   users 1024 Jun  5 08:27 ./
drwx--r-x  17 dblock   users 1024 Jun  5 08:14 ../
-rwxr-xr-x   1 dblock   users   25 Jun  5 08:20 asearch.Linux*
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:27 dicos/
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:27 vestris/
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:27 admin/
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:27 demos/
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:27 docs/
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:27 faqs/
drwxr-xr-x   2 dblock   users 1024 Jun  5 08:27 tools/
-rwxr-xr-x   2 dblock   users   31 Jun  5 08:27 global.cnf
-rwxr-xr-x   2 dblock   users  720 Jun  5 08:27 CONTRIBUTIONS
-rwxr-xr-x   2 dblock   users 2107 Jun  5 08:27 COPYRIGHT
```

```
-rwxr-xr-x   2 dblock   users  419 Jun  5 08:27 README
big-server:~/asearch.1.9$
```

Use vi, joe or any other text editor such as XEmacs to modify the downloaded files as follows.

In global.cnf: if necessary, create or modify the *root* password; do not use your machine's administrative password

```
big-server:~/asearch.1.9$ cat global.cnf
Pass root=test
```

In vestris/asearch.cnf: modify the server you wish to index, you can try the demo with http://www.vestris.com/.

```
big-server:~/asearch.1.9$ cat vestris/asearch.cnf
UrlList=http://www.vestris.com/
Remote=N
ExcludeWords=dicos/english.txt
```

In vestris/search.html: modify the server name (*localhost*) by your server's IP number or server name. You can leave localhost for your local machine.

```
big-server:~/asearch.1.9$ cat html/search.html
...
    <form action="http://www.big-
server.com:9999/vestris/search.html"
        method="post">
...
big-server:~/asearch.1.9$
```

Make sure that the search.html file is readable by everybody (run *chmod 604 html/search.html*). This is required to make the template page accessible.

2.2.5. Index a Site

You can now create a first time index. Alkaline will spider the site defined in asearch.cnf following the rules of the same configuration file. This is also called a clean index. When you run Alkaline in production, it is capable of indexing in background while search is enabled.

To index the site defined by data/asearch.cnf, type *./asearch vestris reindex*.

```
big-server:~/asearch.1.9$ ./asearch.Linux vestris reindex
======================================================
Alkaline Search Engine, Version 1.9 (Jul 18 2000) for Linux
(c) Vestris Inc., Switzerland - 1994-1999 - All Rights Reserved
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
======================================================
[loading /home/dblock/server-prod/t/vestris/asearch.cnf]{
  [UrlList=http://www.vestris.com/]
  [Remote=N]
  [Excludewords=dicos/english.txt]
} loaded 1 inline configuration.
[loading indexes (zero length files are ok)] {
        [vestris/siteidx1.urt][0 lines]
        [vestris/siteidx1.inf][0 lines]
        [building the md5 tree][*****]
        [vestris/siteidx1.lnx][0 lines]
        [checking cross-references, ok]
        [vestris/siteidx1.ndx][0 lines]
} done.
[parsing extensions: htm,html,shtml,txt]
[processed 0/0 resource locators]
[calculating aim [**********] - 0 indexed URLs]
[robots re-
quest for http://www.vestris.com/robots.txt][122 bytes]
[/cgi-bin/][/db-cgi/][/cgi-agnes/][/~dblock/][/alkaline/dicos/]
```

```
  [ExcludeWords][loading di-
cos/english.txt][35666 bytes][8937 elements]
[http://www.vestris.com/] (-1) - [1228 bytes][0]
  [inf][lnx][md5][vix][keys][mta][ndx][ok]
...
[http://www.vestris.com/sti/company.html] (-3) -
[5982 bytes][294]
  [inf][lnx][md5][vix][keys][mta][ndx][ok]
```

This might take some time. You can safely interrupt Alkaline with Ctrl+C after you see

```
[writing databases] {
        [inf][**********]
        [lnx][**********]
        [url][**********]
        [ndx][**********]
} done.
```

The following files will be created in the vestris directory: siteidx1.urt, siteidx1.lnx, siteidx1.ndx, siteidx1.inf.

```
big-server:~/asearch.1.9$ ls -la vestris/
total 205
drwxr-xr-x   2 dblock   users        1024 Jun  5 08:47 ./
drwxr-xr-x   6 dblock   users        1024 Jun  5 08:27 ../
-rwxr-xr-
x   1 dblock   users         429 May 12  1999 asearch.cnf*
-rw-r-r-
1 dblock   users       38201 Jun  5 08:47 siteidx1.inf
-rw-r-r-
1 dblock   users        2205 Jun  5 08:47 siteidx1.lnx
-rw-r-r-
1 dblock   users      147102 Jun  5 08:47 siteidx1.ndx
-rw-r-r-
1 dblock   users       13592 Jun  5 08:47 siteidx1.urt
big-server:~/asearch.1.9$
```

2.2.6. Search Your Index

Run the Alkaline server daemon. It is up to you to detach or not the process by using the Unix & on the command line.

```
big-server:~/asearch.1.9$ ./asearch 9999 vestris
=====================================================
Alkaline Search Engine, Version 1.9 (May 28 2000) for Linux
(c) Vestris Inc., Switzerland - 1994-1999 - All Rights Reserved
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
=====================================================
[checking post 1.3 configuration] {
        [verifying that (vestris/asearch.cnf) exists]
        [file exists (77 bytes)]
[surveillance thread running]
[(re)reading global.cnf]{
  [setting password for {root}]
} loaded 15 bytes.
Alkaline server (Jul 18 2000) running.
[loading /home/dblock/server-prod/t/vestris/asearch.cnf]{
  [UrlList=http://www.vestris.com/]
  [Remote=N]
  [Excludewords=dicos/english.txt]
} loaded 1 inline configuration.
[newly added to Alkaline list: vestris ]
[reloading vestris]
[loading indexes (zero length files are ok)] {
        [vestris/siteidx1.urt][1234 lines]
        [vestris/siteidx1.inf][1234 lines]
        [building the md5 tree][*****]
        [vestris/siteidx1.lnx][1234 lines]
        [checking cross-references, ok]
        [vestris/siteidx1.ndx][45678 lines]
} done.
[indexing thread running - lazy mode]
[parsing extensions: htm,html,shtml,txt]
```

This will run Alkaline in a so called lazy mode, responding to requests on port 9999 and indexing your site again and again in background.

Try to connect to the administrative pages. Navigate to http://www.your-server:9999/ and login using the username *root* and the password you have entered in global.cnf above.

Now try to search. Open the search.html page using a normal browser such as MS Internet Explorer or Netscape Navigator, and try to search something - you should get immediate results.

You can also try to navigate to the following URL: http://www.your-server:9999/vestris/search.html?search=test directly.

2.2.7. Stopping Alkaline

It is safe to terminate Alkaline with a Ctrl-C if it is running in a non-detached mode. You can stop Alkaline from the online management console. Also, the Alkaline tools set contains a perl script, AlkalineStop.pl. Alternatively, to terminate Alkaline, you can kill all Alkaline processes. You can use a variant of: *ps ex | grep asearch | cut -b-6 | sort -rn | xargs -l kill -9*.

2.3. Installation on Windows NT/2000/XP

2.3.1. Download Alkaline

Download Alkaline for Windows NT/2000/XP. Navigate to http://www.vestris.com/files/asearch-distrib/WinNT/ and download asearch.1.9.WindowsNT-ix86.zip. You are also free to choose a current Alkaline beta version which always have the latest improvements from the live source tree. Windows NT for Dec Alpha is not supported, you must have an x86-based configuration.

2.3.2. Install and Test Binary

Connect to your Windows NT server using Terminal Server or login normally as a normal user or an Administrator. The administrative privileges are not required unless you want to run Alkaline as an NT service.

Alkaline has no graphical user interface. It is a direct port from the UNIX version. You must open a DOS prompt to execute the commands explained below. To do that, click on Start -> Run, type in *cmd.exe* and click OK. A black DOS prompt windows will appear.

Use WinZip, pkunzip or any other compressor program to unzip asearch.1.9.WindowsNT-ix86.zip into any directory. Several files will be extracted in the Alkaline.1.9 directory, among them the Alkaline executable, *asearch.exe* and the administration pages directory, *admin*.

```
06/13/00  08:06a 483,328 asearch.exe
02/23/00  12:51p   1,414 CONTRIBUTIONS
02/23/00  12:51p   1,438 COPYRIGHT
02/23/00  12:51p     720 README
02/23/00  12:51p   <DIR> admin
02/23/00  12:51p   <DIR> demos
02/23/00  12:51p   <DIR> docs
02/23/00  12:51p   <DIR> faqs
02/23/00  12:51p   <DIR> tools
        5 File(s)  486,900 bytes
        7 Dir(s)
```

Test the executable. Run *asearch.exe*.

```
N:\asearch.1.9> asearch
========================================================
Alkaline Search Engine, Version 1.9 (Jul 18 2000) for Windows NT
(c) Vestris Inc., Switzerland - 1994-1999 - All Rights Reserved
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
========================================================
    usage: asearch [options] port path [path2 ...]
           asearch [options] path-to-asearch.cnf command ...
     port: port to bind and listen to, ex: 9999
     path: relative path(s) to asearch.cnf files
  options: please refer to the users guide
  command: one of reindex, email, emailall, remove, rx-
match, rxrepl, etc.
_______________________________________________________________
        for more information, please re-
fer to http://alkaline.vestris.com
N:\asearch.1.9>
```

2.3.3. Run the Demo

Alkaline is now ready for a quick demo. Run

```
N:\asearch.1.9> ./asearch.exe 9999 demos\docs
==================================================
Alkaline Search Engine, Version 1.9 (May 28 2000) for Windows NT
(c) Vestris Inc., Switzerland - 1994-2002 - All Rights Reserved
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
==================================================
[checking post 1.3 configuration] {
  [verifying that (demos\docs\asearch.cnf) exists]
  [file exists (61 bytes)]
[loading global.cnf] {
  [Redirect=/admin/]
  [AdminPath=admin]
  [DocumentPath=demos/images,docs,faqs]
} loaded 31 bytes.
[warning, you should define at least a root pass-
word in global.cnf]
[Alkaline server (Apr  2 2002) running, binding to 9999]
[loading N:\asearch.1.9>\demos\docs\asearch.cnf] {
  [UrlList=file:///docs/,file:///faqs]
  [Index.html=index.html]
  [RegExp][enabled for 0 element(s)]
} loaded 1 inline configuration.
[newly added to Alkaline list: Demos/docs ]
[reloading Demos/docs]
[loading indexes] {
  [no index files found (clean index)]
} done.
[indexing thread running - lazy mode]
[01:02:36 2002-04-02][building index in Demos/docs]
[01:02:56 2002-04-02][done building index in Demos/docs]
[01:02:57 2002-04-02][(re)indexing of Demos/docs pro-
cessed 292 new/modified files in 0:21 min]
```

While the engine is running, navigate to *http://localhost:9999/* with your web browser. The admin section of your search engine server will appear. You can select the documentation configuration under *Server Parameters* -> *Server Configurations* -> *demos/docs* and perform searches on it.

2.3.4. Create a Simple Configuration

Download the sample configuration files from http://alkaline.vestris.com/install/install-sample.zip and unzip it into the installation directory. Make sure you preserve subdirectories. The resulting directory is available at http://alkaline.vestris.com/install/.

Use notepad or any other text editor to modify the downloaded files as follows.

In global.cnf: if necessary, create or modify the *root* password; do not use your machine's administrative password

```
N:\asearch.1.9> type global.cnf
Pass root=test
```

In vestris\asearch.cnf: modify the server you wish to index, you can try the demo with http://www.vestris.com/.

```
N:\asearch.1.9> type vestris\asearch.cnf
UrlList=http://www.vestris.com
Remote=N
ExcludeWords=dicos/english.txt
```

In vestris\search.html: modify the server name (www.vestris.com) by your machine's IP number or server name. You can use localhost for your local machine.

```
N:\asearch.1.9> type vestris\search.html
...
    <form action="http://www.big-
server.com:9999/vestris/search.html"
        method="post">
...
```

2.3.5. Index a Site

You can now create a first time index. Alkaline will spider the site defined in asearch.cnf following the rules of the same configuration file. This is also called a clean index. When you run Alkaline in production, it is capable of indexing in background while search is enabled.

To index the site defined by vestris\asearch.cnf, type *asearch.exe vestris reindex*.

```
N:\asearch.1.9> asearch.exe vestris reindex
======================================================
Alkaline Search Engine, Version 1.9 (Jul 18 2000) for Windows NT
(c) Vestris Inc., Switzerland - 1994-1999 - All Rights Reserved
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
======================================================
[loading /home/dblock/server-prod/t/vestris/asearch.cnf]{
  [UrlList=http://www.vestris.com/]
  [Remote=N]
  [Excludewords=dicos/english.txt]
} loaded 1 inline configuration.
[loading indexes (zero length files are ok)] {
        [vestris/siteidx1.urt][0 lines]
        [vestris/siteidx1.inf][0 lines]
        [building the md5 tree][*****]
        [vestris/siteidx1.lnx][0 lines]
        [checking cross-references, ok]
        [vestris/siteidx1.ndx][0 lines]
} done.
[parsing extensions: htm,html,shtml,txt]
[processed 0/0 resource locators]
[calculating aim [**********] - 0 indexed URLs]
[robots re-
quest for http://www.vestris.com/robots.txt][122 bytes]
[/cgi-bin/][/db-cgi/][/cgi-agnes/][/~dblock/][/alkaline/dicos/]
  [ExcludeWords][loading di-
cos/english.txt][35666 bytes][8937 elements]
[http://www.vestris.com/] (-1) - [1228 bytes][0]
  [inf][lnx][md5][vix][keys][mta][ndx][ok]
...
[http://www.vestris.com/sti/company.html] (-3) -
[5982 bytes][294]
  [inf][lnx][md5][vix][keys][mta][ndx][ok]
```

This might take some time. You can safely interrupt Alkaline with Ctrl+C after you see

```
[writing databases] {
        [inf][**********]
        [lnx][**********]
        [url][**********]
        [ndx][**********]
} done.
```

The following files will be created in the vestris directory: siteidx1.urt, siteidx1.lnx, siteidx1.ndx, siteidx1.inf.

```
N:\asearch.1.9> dir vestris\

 Directory of N:\asearch.1.9\vestris

06/14/00  09:35a                    33 asearch.cnf
06/14/00  09:36a                69,404 siteidx1.inf
06/14/00  09:36a                 2,953 siteidx1.lnx
06/14/00  09:36a               183,827 siteidx1.ndx
06/14/00  09:36a                15,760 siteidx1.urt
               7 File(s)        271,977 bytes
```

2.3.6. Search Your Index

Run the server by typing *asearch.exe 9999 vestris*. Ultimately, you will have to install Alkaline as an NT service.

```
N:\asearch.1.9> asearch.exe 9999 vestris
=====================================================
Alkaline Search Engine, Version 1.9 (May 28 2000) for Linux
(c) Vestris Inc., Switzerland - 1994-1999 - All Rights Reserved
```

```
written by Daniel Doubrovkine and Hassan Sultan, Univer-
sity of Geneva
http://alkaline.vestris.com -
dblock@vestris.com / hsultan@vestris.com
=====================================================
[checking post 1.3 configuration] {
        [verifying that (vestris/asearch.cnf) exists]
        [file exists (77 bytes)]
[surveillance thread running]
[(re)reading global.cnf]{
  [setting password for {root}]
} loaded 15 bytes.
Alkaline server (Jul 18 2000) running.
[loading vestris/asearch.cnf]{
  [UrlList=http://www.vestris.com/]
  [Remote=N]
  [Excludewords=dicos/english.txt]
} loaded 1 inline configuration.
[newly added to Alkaline list: vestris ]
[reloading vestris]
[loading indexes (zero length files are ok)] {
        [vestris/siteidx1.urt][1234 lines]
        [vestris/siteidx1.inf][1234 lines]
        [building the md5 tree][*****]
        [vestris/siteidx1.lnx][1234 lines]
        [checking cross-references, ok]
        [vestris/siteidx1.ndx][45678 lines]
} done.
[indexing thread running - lazy mode]
[parsing extensions: htm,html,shtml,txt]
```

This will run Alkaline in a so called lazy mode, responding to requests on port 9999 and indexing your site again and again in background.

Try to connect to the administrative pages. Navigate to http://www.your-server:9999/ and login using the username *root* and the password you have entered in global.cnf above.

Now try to search. Open the search-demo.html page using a normal browser such as MS Internet Explorer or Netscape Navigator, and try to search something - you should get immediate results.

You can also try to navigate to the following URL: http://www.your-server:9999/vestris/search.html?search=test directly.

2.3.7. Stopping Alkaline

It is safe to terminate Alkaline with a Ctrl-C. You can stop a running server from the online administrative console. Alternatively, the Alkaline tools set contains an AlkalineStop.pl perl script. You can also kill Alkaline from the Task Manager although this doesn't give Alkaline a chance to properly cleanup temporary files or make sure that it is not in the middle of writing an index (which would corrupt it).

2.4. Troubleshooting

2.4.1. Segmentation Fault (core dumped).

If you are unable to run Alkaline and this is the only message that you are getting, then you have downloaded a binary of a wrong architecture.

Check what operating system you are running and download the appropriate binary. If you are able to run Alkaline and can reproduce when the segmentation fault occurs more than once, please file a bug report on http://www.vestris.com/.

2.4.2. ./asearch.1.9.OS: can't load library 'xxx.ld.so'.

Your library xxx is missing or is incorrect.

If you're a superuser, find a library distribution and install it. If you are not a superuser (and thus cannot install the library in the /lib or /usr/lib path) you can still try to set the environment variable LD_LIBRARY_PATH to the path of xxx.ld.so - that should fix the problem on some platforms. Refer to your closest unix guru for more details.

2.4.3. asearch.exe has provoked an unexpected error in ???.dll

Please try to reproduce this error and file a bug report at http://www.vestris.com/ and provide as much information about your configuration as possible as well as the exact steps that lead to this error.

2.4.4. Internal Server Error or binary code in browser

http://www.mysite.com/cgi-bin/asearch.exe shows an Internal Server Error or dumps some binary code. Alkaline is not a CGI and you cannot run it with a web request.

You will have to obtain a real access to the server, at least a telnet access or a terminal server access. Under Windows NT you must be able to open a command prompt window and execute programs such as Alkaline. This usually implies that you must be "sitting" on the server or you must use software such as PCAnywhere. Under UNIX, an ftp access is not enough, you must have a telnet access (a root access is absolutely not required).

2.4.5. Browsing to http://server:port/ does not work

Make sure that Alkaline is running. If you are presented with a login dialog when you connect to http://server:port/, this means that Alkaline is running. When you supply the username *root* and the password from the global.cnf file, the dialog should not appear again. If it does, check the password in the global.cnf file again.

If you get an error such as *404 File not Found* or *the Admin directory does not exist*, this means that Alkaline was unable to find the admin directory. It must be in the same path from where you launch Alkaline and must be readable by the asearch process.

2.5. Upgrading Alkaline

2.5.1. Finding the Current Software Version

To find the current version of Alkaline, run the *asearch* binary with no parameters. The header includes the exact version of your currently running software. If the version is not present in the header, you are running Alkaline 1.1 or earlier.

On Windows, you can select the *asearch.exe* program in the File Explorer and choose the Version tab on the Properties sheet. This provides, among others, the exact version of your program. If the Version property sheet is not present, you are running Alkaline 1.3 or earlier.

With Alkaline 1.4 and above, the Administration pages show the product version in the *Performance Counters* section.

2.5.2. Before You Begin

Backup your entire Alkaline installation. Make sure that if for whatever reason the upgrade process fails, you can revert to the previous configuration. This involves copying the entire directory containing the Alkaline binary or executable, the indexes and the *admin* directory if any.

2.5.3. Upgrading to Version 1.7

Alkaline 1.7 fixes several bugs related to the usage of the GET method. It is necessary to modify search templates and custom scripts to reflect the following changes:

- *start* and *quant* parameters are passed as name=value pairs, such as start=10 rather than start10
- options on the query string are separated with & and not +, the later now equivalent to a space

2.5.4. Upgrading from Version 1.4 and Above

Download the version of Alkaline you wish to upgrade to from http://alkaline.vestris.com/. Uncompress the contents of the distribution into a temporary directory.

Stop the Alkaline service as usual. Please refer to the Stopping Alkaline FAQ for details on various methods of shutting down the service.

Replace your current installation *asearch* binary and *admin* directory with those from the new distribution. There should be no additional permissions or settings to apply.

Restart the service or daemon. Follow the same procedure as you usually use. Test that all your regular operations work properly.

2.5.5. Upgrading from Earlier Releases

First, check whether you have an *equiv* directory in your folder where Alkaline is installed. If you don't, follow the instructions for versions 1.4 and above.

Although new versions of Alkaline are backward compatible to the *equiv* structure format, it is recommended that you follow the procedure described below as the equiv structures are deprecated.

Download the version of Alkaline you wish to upgrade to from http://alkaline.vestris.com/. Uncompress the contents of the distribution into a temporary directory.

Stop the Alkaline service as usual. Please refer to the Stopping Alkaline FAQ for details on various methods of shutting down the service.

Replace your current installation *asearch* binary and copy *admin* directory from the new distribution. There should be no additional permissions or settings to apply.

Examine your *equiv/equiv.struct* file. For each group or alias that is searchable (the first parameter in the url query for search), make sure you already have a subdirectory at the current level. For example, if your search query was *http://server:9999/?Foo+FooHTML*, you probably have a directory *foo*.

For all HTML aliases in the *equiv/equiv.struct* file, create a .aln or an html file under the alias subdirectory. For example, if your search query was http://server:9999/?Foo+FooHTML, and FooHTML is an alias for a remote url, such as *http://server/alkaline/search.html*, create foo/search.aln with a single line of contents, *http://server/alkaline/search.html*. If the alias was pointing to a local file, copy that local file under the alias subdirectory. For example, if FooHTML was an alias for */home/www/search.html*, copy this file to *foo/search.html*.

Examine other options in the *equiv/equiv.struct* file. They include *Proxy* and other options. Create a global.cnf file in the current directory and populate it with equivalents. For example, *Proxy,proxy.server.com:8080* becomes an entry in a simple text file, global.cnf, as *Proxy=proxy.server.com:8080*. Global.cnf options and parameters are detailed in *Appendix II: Known global.cnf Variables*.

Passwords are scrambled in equiv/admin.struct. You must create new passwords in the global.cnf file. Typically, add the *root* password. For example, add *Pass root=test* in your global.cnf file.

Replace the search queries in your search templates and throughout your site. For example, if your search query was *http://server:9999/?Foo+FooHTML*, it becomes *http://server:9999/foo/search.html* or *http://server:9999/foo/search.aln*.

Delete the equiv subdirectory.

Restart the service or daemon. To start the daemon now, you need to specify subdirectories rather than aliases. For example, *./asearch 9999 foo bar*. Test that all your regular operations work properly.

Chapter 3. Alkaline Server Configuration

3.1. asearch.cnf Configuration File

This file defines various configuration options that drive the indexing process for each individual site. The file name, *asearch.cnf* cannot be changed.

Each configuration option has to be written on a single line and has the following format.

```
Name=Value
```

- The order of options does not have any significance.
- It is possible to specify comments by starting a line by a # sign.

For example:

```
UrlList=http://www.foo.com
Remote=N
```

Directives supported in this file are described in *Appendix I: The asearch.cnf Configuration Reference*.

The *asearch.cnf* file can contain multiple *virtual* configurations, separated by * (star) characters. Each virtual configuration can have separate options and url lists. For example:

```
UrlList=http://www.foo.com
Remote=Y
UrlInclude=foo.com

* Virtual Configuration Bar
UrlList=http://www.bar.com/
Remote=N
```

3.2. Server Paths, Aliases and Templates

When running Alkaline, it is possible to specify multiple paths containing different asearch.cnf files. For example:

```
./asearch 9999 foo bar
```

will load both configurations and indexes from foo/asearch.cnf and bar/asearch.cnf. Each path, such as *foo* is called an alias, since it defines a particular configuration. It must contain an asearch.cnf file.

Each path can contain templates in a form of html files with specific Alkaline tags. To specify a remote file, you must create a file with a *.aln* extension containing the url to query on a single line.

For example, the directory foo can contain a file called search.html containing the search box and Alkaline tags for mapping search results. To search the foo index, navigate to *http://server:port/foo/search.html*.

If the search.html template is in fact at http://www.bar.com/search.html, a file called search.aln can be created in the directory foo with a single line in it:

```
http://www.bar.com/search.html
```

To search the foo index and display results using this template, navigate to *http://server:port/foo/search.aln*.

3.3. The global.cnf file, Server Passwords

Alkaline has such global settings as the *LogPath* or the HTTP proxy. It also has password protected areas, such as the online management. Passwords and additional global configuration options are defined in the global.cnf file that has to be created in the same directory where Alkaline is run from. The syntax for passwords, for example, is:

```
Pass name=password
```

Alkaline predefined global.cnf options and passwords are fully described in *Appendix II: Known global.cnf Variables* and *Appendix III: Known global.cnf Passwords*.

Chapter 4. Searching with Alkaline

4.1. Simple Search

The Alkaline search engine finds documents on an internet site, several unrelated internet sites or an intranet domain. To search for any information you have to type in a sequence of words that define what you are looking for. The search engine will output a list of results, best results first. Alkaline searches exact words and word heuristics (parts of words) only. It does not do fuzzy or misspelled words search. Alkaline does not search phrases (yet).

A simple search is done by typing a word. Searching for *light* will find all pages containing *light*, *light*ning, de*light*ed, etc. It will also find pages with *Light* and *Light*ning because searching is case-insensitive by default.

Searching multiple words is done by typing a sequence of words separated by spaces. Searching for *ricky blue* will find all pages containing *Ricky*, t*ricky*, *blue*, *blue*s, etc. Pages containing both words will be shown first in the results. Pages with these two words in the title or in meta tags will be more relevant.

Case-sensitive search can be enabled by using a single capital letter inside a word. For example, searching for *Intranet let* will find all pages containing *Intranet* and *let*ter, but will not list pages with just *intranet*. Case-sensitivity applies to the word with capital characters only.

Entire words can be searched by using quotes. Searching for *"net"* will find pages containing *net* and *Net*, searching for *"Net"* will of course find only pages containing *Net* because of the capital N enabling the case-sensitive search.

4.2. Boolean Search

Boolean search allows to lookup pages containing some word and not containing other. To express the fact that a page must contain a word, a + sign must be placed in front of the word. To search for all pages not containing a word, a - sign should be used. For example, you can search *+net -"Internet"* which will show pages containing *net*, *network*, etc., only if they don't contain *Internet*. Note that searching for *-word* or *+foo -foo* will produce no results.

Refined boolean search can be done by mixing a boolean expression with normal words. Searching *+net +Foo bar* will sort results showing pages containing *net*, *Foo* and optionally *bar* first.

4.3. Meta Data Search

Meta search can be done by specifying a meta tag followed by a column, for example *author:foo*. You can get refined meta results too, for example *author:+"Foo"* and *+author:"Foo"* will produce same output. Searching *author:"Foo Bar"* is equivalent to *author:"Foo" author:"Bar"*.

All case-sensitivity, full and partial word rules apply with the meta data search. The meta tag name is never case-sensitive and is always exact.

4.4. Numeric Data Search

Alkaline will index words, such as *price=234*, in a special manner. You can consequently perform a numeric data search: *price<234*, *price=234* or *price>234*.

All rules about case-sensitivity are preserved with the meta search. There is no partial word support for numerical meta tags, if the words *"price=234"*, *"aprice=234"* and *"priced=234"* are in the index, searching for *price=234* will only find *"price=234"*.

4.5. Returning all pages

Searching for the word *"*"* will produce a list of all indexed documents.

You can use this search with the search scope features to display all the indexed pages in a specific directory, or all the indexed pages dated within certain time constraints.

4.6. Restricting Search Scope

To *define a scope* means to specify a more precise location of a document or to restrict the search query to a particular range. Url scoping is particularly useful as Alkaline is capable of indexing multiple sites with the same daemon.

Host scope is specified by adding the rightmost part of a host entry, such as *host:.host.com* to the search query. For example, to search *foo* in documents at http://www.foo.com, the following command should be issued: *foo host:.foo.com*. It is possible to specify multiple hosts by writing more than one *host:* elements in your search string. For example: *host:.bar.com host:.foo.com +foo -bar* will return all documents indexed at *.foo.com, *.bar.com containing "foo" and not containing "bar".

Path scope is specified by adding the leftmost part of a path without the leading slash, such as *path:foo/bar* to the search string. For example, to search *foo* in documents at http://www.foo.com/foo/bar, the following command should be issued: *host:www.foo.com path:foo/bar foo*.

It is possible to mix *host:* and *path:* as it is possible to specify multiple path entries.

Url scope is specified by adding the leftmost part of a complete url, including the leading *http://*, such as *url:http://www.foo.com/bar* or *url:file:///foo/bar*to the search string. If you choose to omit the protocol, http:// is assumed. For example, to search *foo* in documents at http://www.foo.com/bar, the following command should be issued: *url:www.foo.com/bar foo*. To specify a scope to urls with spaces and other special characters, use the encoded url value, such as *url:www.foo.com/foo%20bar*.

File extension scope is specified by adding the rightmost part of a filename without the leading dot, such as *ext:cpp,h* to the search string. Multiple extensions can either be specified separated by commas or by adding more than one *ext:* parameter to the search string. For example to search *foo* in all *.bar* documents, the following command should be issued: *ext:bar foo*.

Alkaline will return documents matching the search query and *any* of the host:, path: or url: scope options *and* the ext: scope delimiter, if present. As usual if no scope is specified, the entire indexed domain is searched. It is possible to specify multiple values in all scope delimiters by separating them with commas or by adding multiple scope options to the search string.

Meta scoping is specified by adding *meta:name1[,name2]* to the search string. Multiple meta tag names can either be separated by commas or by adding more than one *meta:* parameter to the search string. For example to search *foo bar* in all keywords only, the following command should be issued: *foo bar meta:keywords*, which is equivalent to *keywords:foo keywords:bar*. Meta scoping was added in version 1.6.

4.7. Restricting Date Scope

To search documents modified after or before a specified date add *before:* and/or *after:* to the search string followed by a valid date of one of the following forms DDMMYY, DD/MM/YY, DD.MM.YY, DD-MM-YY, DDMMYYYY or DD/MM/YYYY.

> You can specify both *before* and *after*, for example *before:12.05.1999 after:01.05.1999* will return all documents between these two dates. Note that the bounds are not included in the search results.

4.8. Forcing Search Options

By default, Alkaline will choose a case-sensitive search when at least one upper-case letter is present in a word. To search all words case-sensitive, *opt:case* should be added to the search string. To search all words case-insensitive, *opt:insens* should be used.

To search all pages containing all words, *opt:and* should be added to the search string. The default behavior of Alkaline is to search all pages containing any of the words and producing best results first.

To force searching of whole words only, *opt:whole* should be added to the search string. The default behavior of Alkaline is to do partial matches.

It is of course possible to specify more than one such option by separating them by commas or by adding multiple *opt:* entries to the search string, for example: *foo opt:whole,case* will return all pages containing the exact word "foo".

4.9. Hints and Techniques

Punctuation is not indexed, but future versions might use it for phrasal search. Searching a word with a comma will be unsuccessful because the comma is used as any other character together with other punctuation marks.

A dash "-" is a common character, except placed in the beginning of a word that is not quoted (boolean search). Words with a dash can be searched and indexed. Alkaline can search and index numbers and words with digits.

Accentuated characters, such as in French, are translated into respective variants (é to e, à to a, etc.) at both search and index. Multilingual servers using Alkaline might have this feature removed to enable searching of Russian text for example.

Chapter 5. Running Alkaline

5.1. Running Alkaline as a Daemon

In order to launch Alkaline as a daemon, you must run it from the directory that contains the *aliases* subdirectories, which in their turn contain asearch.cnf files and siteidx indexes. The command line syntax is:

```
asearch [options] [host:]port path1 [path2 [...]]
```

For example:

```
asearch 8080 foo bar
```

The *port* must be a positive number. It must be unused by any other process. If you have multiple network cards, you can specify a host entry. For example

```
asearch foo.bar.com:8080 foo bar
```

The *paths* must be relative and be subfolders of the current directory. If this is not the case, they will not be searchable.

On UNIX, you can detach the Alkaline process (so it is not killed after you close the terminal or telnet session) by running:

```
./asearch 8080 foo &
```

5.2. Reindexing a Database

Reindexing a database can be done from command line in a fully verbose mode. The command line syntax is:

```
asearch [options] path-to-asearch.cnf reindex
```

For example:

```
asearch /usr/alkaline/data/ reindex
```

5.3. Gather Email Addresses

Using a standard Alkaline configuration, you can gather all email addresses found on a site. The command line syntax is:

```
asearch [options] path-to-asearch.cnf email
asearch [options] path-to-asearch.cnf emailall
```

For example:

```
asearch /usr/alkaline/data/ email
```

Using the *email* option will output each email address once, the *emailall* option will output all email addresses found.

The email and emailall directives will not output any copyright or information notice, the output will be exclusively an unsorted list of email addresses and the url they have been found at, separated by tabs. No database files are written, Alkaline will simply spider the given site page after page following links as usual.

You might find it useful to process this output:

```
big-
server$ ./asearch data/ email | awk ’{print "Email:("$1") Server:("$2")"}’
Email:(webmaster@foo.com) Server:(http://www.foo.com/index.html)
Email:(foo@foo.com) Server:(http://www.foo.com/foo.html)
Email:(admin@foo.com) Server:(http://www.foo.com/foo.html)
```

5.4. Remove Urls from a Database

It is possible to remove individual urls from an existing database. The command line syntax is:

```
asearch [options] database-
path remove url1[*] [url2[*]][url3[*] ...]
```

For example:

```
asearch /usr/alkaline/data/ remove http://www.foo.com/file.html
```

You can remove an entire site or subdirectory by passing an url ending with a *. For example:

```
asearch /usr/alkaline/data/ remove http://www.foo.com/*
```

5.5. Merging Databases

It is possible to merge multiple databases into one single database. The command line syntax is:

```
asearch [options] target-database-
path merge database1 [database2][database3 ...]
```

For example:

```
asearch /usr/alkaline/foo/ merge /usr/alkaline/bar/
```

The target database must exist and is being merged into. When merging, warnings will be issued when same urls are present in both source and target databases. The resulting database is a true union (a real complex merge), each url or indexed word is present once only as if the indexing was done from scratch.

5.6. Exclude Words from an Existing Database

It slows the indexing process to use an ExcludeWords dictionary. It is possible to apply an existing exclusion dictionary of regular expressions or simple words to a written database. The command line syntax is:

```
asearch [-regexp] database-
path excludewords file1 [file2][file3 ...]
```

For example:

```
asearch /usr/alkaline/data/ excludewords bar/words.txt
```

This feature was added 03-Aug-2000. The regular expressions extension was added in version 1.5.

5.7. Testing Regular Expressions

You can test regular expressions with Alkaline in order to save time when building the appropriate word exclusion dictionaries or url exclusion lists. It also allows to test regexp replacements for options supporting it. The command line syntax is:

```
asearch [string] rxmatch [regexp]
asearch [string] rxrepl [source regexp] [target regexp]
```

For example:

```
$ ./asearch "http://server/whatever.cgi?name=value" rx-
match "(http://server/whatever\.cgi\?)(.*)(=value)"
[http://server/whatever.cgi?name=value] positively
 matches [(http://server/whatever\.cgi\?)(.*)(=value)]

$ ./asearch "http://server/whatever.cgi?name=value" rx-
match "(http://server/whatever\.cgi\?)(.*)(=other)"
[http://server/whatever.cgi?name=value] does not
 match [(http://server/whatever\.cgi\?)(.*)(=other)]

$ ./asearch http://www.vestris.com rxrepl "(.*)(www)(.*)" "\1ns\3"
```

The rxmatch feature was added 02-Jul-2000. The rxrepl feature was added 12-Jul-2000. For more information about the regular expressions features, please refer to the RegExp asearch.cnf directive reference.

5.8. Querying Available Settings

It is possible query your Alkaline for a structured list of supported options and settings in global.cnf and asearch.cnf. The produced output contains all available options, types, defaults, descriptions and boundary values, if any.

The command line syntax is:

```
asearch options
```

For example:

```
# ./asearch options
[global.cnf options]
CacheTemplates: boolean, default=true - [cache search templates]
KeepAlive: boolean, default=false - [allow to keep-
alive clients]
Nagle: boolean, default=true - [disable nagle algorithm]
...
[asearch.cnf options]
WriteIndex: number, default=100 - [database write interval]
Retry: number, default=3, min=1, max=3 -
[retry count for a timed-out connection]
Timeout: number, default=30, min=1, max=30 -
[network timeout period]
...
```

This command is supported since version 1.41.0421.0.

5.9. Parsing Html Documents

It is possible parse html documents using Alkaline in order to track html errors and typos. You can parse local and remote documents alike.

The command line syntax is:

```
asearch parse [-auth:[domain\]username[:password]] [-
verbose] url [url [...]]
```

Where the *auth:* option specifies optional credentials to use and the *verbose* option enables verbose output of headers.

For example:

```
# ./asearch parse -
auth:foo:bar http://www.vestris.com /home/www/index.html
```

The output includes the complete dump of all structured tags that Alkaline finds, but does not show raw html content.

This command is supported since version 1.41.0424.0.

5.10. Command Line Options

Each option must start with a dash or a double dash. The order of options is insignificant.

Table 5-1. Command Line Options

l (Log)	Show query strings as they are requested by a client.
v (Verbose)	Show progress of the background indexing thread in lazy mode. Will produce same output as with the reindex command when running Alkaline as a daemon.

sf=X (SleepFile=X)	Force the reindex thread to sleep X seconds between two files being reindexed. X must be a positive integer or 0.
sr=X (SleepRoundtrip=X)	Force the reindex thread to sleep X seconds after each indexing roundtrip. X must be a positive integer or 0.
No404	Skip pre-processing (verification of removed files). This option can be used when you know that no full sections, but at most individual documents have been removed from the site. The HTTP/404 verification step is necessary when a page foo.html references bar.html and the latter references dummy.html. Removing bar.html and dummy.html means that the link path to dummy.html is broken and the page will never be removed from the index as it cannot be reached any longer.
Reindex	Force the Reindex=N flag in all configurations during background indexing.
Expire	Force the expire option in the configuration files; treat all documents as out-of-date.
Once	Perform background indexing only once upon daemon launch, newly runtime added groups into the equiv/equiv.struct will be reindexed once only as well.

NewOnly	Force the NewOnly=Y option in the configuration files; index only those files that are not already in the index, this also allows to restart indexing from the interrupted point.
Ssi	Enable server-side includes on Alkaline template pages.
Exv	Track which urls are included or excluded, which pages are indexed or skipped and for what humanly readable reason.
Exx	Dump html parser tags for tracking of html errors.
d (Daemon)	Run Alkaline server as a true daemon, suppress all output; note that in this mode restarting the server is not possible (you need to kill it explicitly). This option will not detach Alkaline under Windows NT.
mt=X (Maxthreads=X)	Specify X as the maximum amount of threads in the search thread pool; each thread executes a single search and will remain idle till a new search operation is requested. A thread that was inactive for a certain amount of time will die. If more requests than the maximum number of threads are made, they are queued and processed as a thread becomes available. Default value is 100.

mi=X (Maxindexthreads=X)	Specify X as the maximum amount of threads in the index thread pool; each thread executes a single index operation and will remain idle till a new search operation is requested. Default value is 10.
ai=X (AcceptInterval=X)	Specify X as the interval in milliseconds to sleep after each new connection is accepted. Heavily flooded servers have seen stability greatly improved with this option and values around 200-300 ms. The operating system maintains a list of incoming connections that have not been accepted yet. After accepting, Alkaline places the connection into a queue and a thread from the thread pool will pickup the connection and service it. Accepting connections more often than what the server can physically process will lead to a growing thread pool, more elements in the Alkaline’s service queue and overall worse performance. Added 03-Jul-2000. Default value is 0.
EnablePing	Enable the self ping thread that makes sure that the engine is still running every 15 seconds. If Alkaline dies, the ping thread will attempt to terminate the dead process. Replaced DisablePing in 1.31.0907.0.

EnableSwap[=filename]	Enable the swap mechanism and use less physical memory. On a UNIX system, you can optionally specify a file name. For more information about the swap, please consult the Alkaline Virtual Memory and Swap section. This option replaced *DisableSwap* on 04-Aug-2000.

The THREAD_ options set the priority value for the background reindex thread on Windows NT. This value determines the thread’s base priority level. The system uses the base priority level of all executable threads to determine which thread gets the next slice of CPU time. Threads are scheduled in a round-robin fashion at each priority level, and only when there are no executable threads at a higher level does scheduling of threads at a lower level take place.

When manipulating priorities, be very careful to ensure that a high-priority thread does not consume all of the available CPU time. Using REALTIME_PRIORITY_CLASS may cause disk caches to not flush, hang the mouse, and so on. Also using THREAD_TIME_CRITICAL may have the same disastrous effect.

Don’t mix up class and process priorities. The class priority is the base priority of Alkaline and cannot be changed by Alkaline options. The process priority is relative to the class priority and is set for the background indexing thread using the THREAD set of options.

Table 5-2. Windows NT Specific Options

THREAD_BOOST	When a thread is running in one of the dynamic priority classes, the system temporarily boosts the thread’s priority when it is taken out of a wait state.

THREAD_NOBOOST	Default behavior for thread priority boosting.
THREAD_ABOVE_NORMAL	Indicates 1 point above normal priority.
THREAD_BELOW_NORMAL	Indicates 1 point below normal priority.
THREAD_HIGHEST	Indicates 2 points above normal priority.
THREAD_IDLE	Indicates a normal priority unless Alkaline's priority class is manually changed. Alkaline's main thread always runs in NORMAL_PRIORITY_CLASS. You may change that using the Task Manager. IDLE indicates a base priority level of 1 for IDLE_PRIORITY_CLASS, NORMAL_PRIORITY_CLASS, or HIGH_PRIORITY_CLASS processes, and a base priority level of 16 for REALTIME_PRIORITY_CLASS processes.
THREAD_LOWEST	Indicates 2 points below normal priority. We advise to use this option in general when you want Alkaline to work as little as possible when running on a general purpose web server.
THREAD_NORMAL	Indicates normal priority.
THREAD_TIME_CRITICAL	Indicates a base priority level of 15 for IDLE_PRIORITY_CLASS, NORMAL_PRIORITY_CLASS, or HIGH_PRIORITY_CLASS processes, and a base priority level of 31 for REALTIME_PRIORITY_CLASS processes.

Chapter 6. Customizing Search Results

6.1. Creating Search Templates

A template is almost nothing more than an HTML document with special tags used to output Alkaline search results. On UNIX, a template must have public read permissions in order to be used. Type *chmod 604 search.html* (replace by your file name and location), if you get *401 Unauthorized* error messages when trying to perform a search operation. Various sample templates are available at http://www.vestris.com/alkaline/asearch.cnf/.

Virtually any HTML layout can be specified using templates. It is possible to mix script code, embedded objects, etc. Templates can have tags, extended tags and options. Simple regular expressions are available with some options.

All possible tags and options and their usage are detailed in the Search Templates Tags and Options appendix.

6.1.1. Simple Tag

A simple tag is always of format:

```
<!-NAME->
```

Tags define the location of some predefined entity in the resulting document. Each tag will be replaced by the appropriate content dynamically generated by Alkaline.

6.1.1.1. Example

```
<!-SEARCH-RESULTS->
```

6.1.2. Extended Tag

An extended tag is always of format:

```
<!-NAME-expression->
```

Extended tags define the location of some predefined entities in the resulting document and parameters or format for the output of the later. Each extended tag will be replaced by the appropriate content dynamically generated by Alkaline.

6.1.2.1. Example

```
<!-SEARCH-GENERAL-$total results found->
```

6.1.3. Option

An option is always of format:

```
<!-SET NAME-expression->
```

Options define the global behavior of the search engine output.

6.1.3.1. Example

```
<!-SET DATE-$Day:$Month:$Year->
```

6.2. Creating Search Input Forms

6.2.1. Prerequisites

Alkaline is an HTTP/1.0 compliant server. It requires you to specify what database you want to search and what template you wish to use to output search results. This is done by specifying a server path when performing a query. For example http://foo.com:9999/bar/search.html mean that the bar alias has to be searched and results should be shown using the search.html template. Alternatively, you can specify a *searchconfig* parameter and thus use the same template for multiple configurations.

Forms can use both the POST and the GET methods.

In order to search a database, Alkaline must be running as a daemon.

6.2.2. A Simple Search Form

A simple form is:

```
<form method="post"
 action="http://server:port/foo/search.html">
<input type="text" name="search">
</form>
```

The above form assumes that Alkaline is running on a *server*, bound to *port* and that the relative path *foo* contains the asearch.cnf file, indexes and the search template *search.html*.

If either the path does not exist or the search template cannot be retrieved or is invalid, Alkaline will respond with a 400/Bad Request.

The search form can contain the following additional fields:

Table 6-1. Predefined Form Fields

host	append a host: scope restriction entry to the search string
path	append a path: scope restriction entry to the search string
url	append an url: scope restriction entry to the search string
other	append any additional value to the search string

quant	specify the maximum number of results per page to show
before	append a before: time scope restriction entry to the search string
after	append an after: time scope restriction entry to the search string

Example:

```
<form ac-
tion="http://www.foo.com:8080/foo/search.html" method="post">
 <input type="text" name="search" size="20">
 <br>in
  <select name="other">
   <option value="">The Entire Site</option>
   <option value="url:www.foo.com/doc/">Documentation</option>
  </select>
  <input type="submit" value="Search">
  <br>modified after: <in-
put type="text" size="10" name="after"> (example: 15.05.1999)
  <br><input type="checkbox" name="other" value="opt:and"> match all terms</
   <in-
put type="checkbox" name="other" value="opt:case"> case sensitive</input>
   <in-
put type="checkbox" name="other" value="opt:whole"> whole words only</input>
  <br>Show <select name="quant">
   <option selected value="10">10</option>
   <option value="20">20</option>
   <option value="-1">all</option>
  </select> results per page.
</form>
```

6.3. Writing Expressions

6.3.1. Introduction

The Alkaline simple regular expressions extend mapping possibilities for search results. Extended tags, such as <!–name expression–> and options, such as <!–SET name–expression–> allow values in a special simple regexp form.

Expressions allow, for example, to show different output when no results were found. They also enable such advanced features as inserting a picture for search results from a particular location in the site. In either way, expressions help search templates designers to give their Alkaline users a look and feel of the site being searched.

6.3.2. Expression Variables

A variable name is a well defined entity beginning with a $ sign, followed by a series of alphanumeric elements, dots and dashes. Variables are used in Alkaline specific extended tags and options. For example, *$url* is a variable and is being translated into an url when used in a <!–SET MAP–expression–> option. For example:

```
<!-SET MAP-this is an url: $url->
```

Variables use the following general syntax:

```
{$£}{name}[|COMMAND{PARAMETER}]~[Prefix]#[Postfix]^[Elsefix]
```

To output a variable value, simply use it with a $ sign, for example *$url*. To evaluate a variable, use it with a £ sign, for example *£url~[Url is not empty!]*. As you might have guessed, the final result will be "Url is not empty!" if the variable *url* contains any text and nothing will be output when it is empty.

Variable names can contain spaces and special characters. Such a variable should be used between brackets, for example *$[Custom Meta Tag]*.

In general, if the variable is empty, *Elsefix* is output. Otherwise, *Prefix* is inserted before and *Postfix*, after the variable value. Any of the three sections *Prefix*, *Postfix* or

Elsefix can me omitted.

6.3.3. Expression Commands

Commands allow to pre-operate on the variable. This means that the variable value will be affected before it is checked for having any data for the *Prefix*, *Postfix* or *Elsefix* output. You can chain commands by separating them with a comma. Commands can be one, or a combination of the following:

Table 6-2. Expression Commands

TRIM	remove leading and trailing spaces
TRIM32	remove leading and trailing spaces, carriage returns, tabs and line feeds
LEFT{SIZE}	leave leftmost SIZE characters
RIGHT{SIZE}	leave leftmost RIGHT characters
IS{STRING}	clear if not STRING (case-sensitive)
NOT{STRING}	clear if STRING (case-sensitive)
HAS{STRING}	clear if does not contain STRING (case-insensitive)
STARTS{STRING}	clear if does not start with STRING (case-insensitive)
ENDS{STRING}	clear if does not end with STRING (case-insensitive)
REPLACE{SOURCE:TARGET}	replace SOURCE with TARGET (case-sensitive) (added in version 1.7)
REPLACI{SOURCE:TARGET}	replace SOURCE with TARGET (case-insensitive) (added in version 1.7)
UPCASE	convert to upper-case
LCASE	convert to lower-case
REVERSE	reverse value

MORE{NUMBER}	clear if term is smaller or equal to NUMBER (leave unchanged if more than NUMBER)
LESS{NUMBER}	clear if term is bigger or equal to NUMBER (leave unchanged if less than NUMBER)
URLENCODE	encode as an URL (eg. space becomes %20)
URLDECODE	decode as an URL
HTMLQUOTE	encode as html (eg. & becomes &amp;)
HTMLDEQUOTE	decode as html
CLEFT{NUMBER}	cut the NUMBER leftmost characters
CRIGHT{NUMBER}	cut the NUMBER rightmost characters
URLSCH	url scheme, such as http (added 02-Jul-2000)
URLHOST	server name (added 02-Jul-2000)
URLDIR	full document path, cannot be empty (added 02-Jul-2000)
URLFILE	file name (added 02-Jul-2000)
URLARG	parameters after ? (added 02-Jul-2000)

6.3.4. Examples

The easiest way to understand how expressions work in Alkaline is to look at examples with the following data:

Table 6-3. Variable Values

Variable Name	Variable Value

search	foo
url	http://www.foo.com/bar/
dummy	
quant	10

Table 6-4. Expression Examples

Expression	Output
$dummy~[Dummy is empty...]^[Dummy is not empty...]	Dummy is empty...
Searching for "$search", $quant results found.	Searching for "foo", 10 results found.
$quant~[Found]#[documents.]^[No documents found.]	Found 10 documents.
£search~[Searching $search.]^[Nothing to search!]	Searching foo.
$url|TRIM,LEFT10	http://www
$url|[REPLACEfoo:bar]	http://www.bar.com/bar/
£url|[HASftp://]~[Ftp!]^[£url|[STARTShttp://]~[Web!]]	Web!

6.4. Server Side Includes

6.4.1. What is SSI?

Server-Side includes is an NCSA standard that allows users to create documents that provide simple information to clients on the fly. Such information can include the current date, the file’s last modification date, and the size or last modification of other files. In it’s more advanced usage, it can provide a powerful interface to CGI and

/bin/sh programs.

SSI means basically that your document will be parsed by the server for special SSI tags before producing output to the client. Alkaline partially supports the SSI standard.

Full details about the original server-side includes can be found in the NCSA SSI tutorial at http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html.

6.4.2. Alkaline SSI Support

Alkaline has a limited support for SSI. This support is only partially compliant with the SSI standard for various reasons, mainly because Alkaline’s template usage policy is already very restrictive and only the administrator can decide which person/document to trust.

Several security issues that might have heavy impact should be considered and it is advised that SSI are enabled in last resort with a permanent consideration about what enabling SSI will imply in terms of server stability and security compromise. It is still clear that SSI is not a security issue if everything is done properly.

To enable SSI with Alkaline, you should include –ssi or –EnableSsi to the command line. The default behavior of Alkaline is that SSI are disabled.

If SSI are enabled, Alkaline will look for special SSI tags and attempt to execute them on all generated template pages (search results). Thus SSI tags can of course be included on template pages only.

The fact that searched documents have SSI code in them will not influence any behavior in Alkaline as such code is discarded by the parser.

SSI tags can contain Alkaline search variables from the <!–SEARCH-GENERAL expression–> tag. Alkaline will process the full SSI string between <!–# and –> for options such as $search, which will be replaced by their values before being handled to the SSI processor.

6.4.3. Valid SSI Tags

An SSI tag has the following format:

```
<!-#name operation=command ...->
```

6.4.3.1. include virtual=url file=local

Include a document contents into the resulting page at the location of the include tag. If the document does not exist, an error will be produced.

To specify a proxy server for the retrieval of a virtual document you must use the Proxy option in global.cnf.

Unlike the SSI definition, Alkaline will not check for localness of documents, the URLs and the local paths can be anything. Alkaline will neither verify the content of the documents.

Example:

```
<!-#include file="c:\alkaline\asearch.cnf"->
<!-#include virtual="http://www.vestris.com/index.html"->
```

6.4.3.2. echo var=name

Echo an environment variable or variables.

Example:

```
<!-#echo var="PATH"->
```

6.4.3.3. fsize list-of-files

Show a formatted size for each file in the list.

Example:

```
<!-#fsize /bin/asearch->
```

6.4.3.4. fcreated/fmodified list-of-files

Similar to fsize, show the creation/modification date for a file. Unlike under the SSI definition, only the fixed locale formatting is available.

Example:

```
Alkaline server compiled: <!-#fcreated /bin/asearch->
```

6.4.3.5. exec cmd=cmdline cgi=url

Execute a local command or retrieve a remote document/cgi.

This is a dangerous command. The user that can modify the template file can run any command with the rights of the running Alkaline. Such a user can destroy valuable data!

Unlike the SSI definition, Alkaline will not check for localness of documents, the urls and the local paths can be just about everything. But unlike for the include command Alkaline will verify that remote documents return a text/* mime format.

Example:

```
<!-#exec cmd="chkdsk"->
<!-#exec cgi="http://server.com/cgi-bin/test"->
```

6.5. WAP/WML Templates and Wireless

Support

The Wireless Application Protocol (WAP) is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly. You can find more information about WAP at http://www.wapforum.org.

WML is a markup language that is based on XML (eXtensible Markup Language). The official WML specification is developed and maintained by the WAP Forum, an industry-wide consortium founded by Nokia, Phone.com, Motorola, and Ericsson. This specification defines the syntax, variables, and elements used in a valid WML file. The actual WML 1.1 Document Type Definition (DTD) is available for those familiar with XML at: http://www.wapforum.org/DTD/wml_1.1.xml.

Alkaline 1.7 supports serving the WML text/vnd.wap.wml content-type. This means that wireless clients can perform search operations on Alkaline-powered servers when directed to a WML template.

6.5.1. WML Search Form

A WML search form is similar to an html form. Before you start designing an Alkaline WML search form and template, please make sure you read any WML reference and get familiar with concepts of decks and cards.

A typical wml search form for a configuration *foo* is:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
   <card id="searchcard" title="Search">
    <p>
     <do name="back" type="prev" label="back">
      <prev/>
     </do>
     <big>Search</big><br/>
```

```
      <input name="search" size="15"/>
      <select name="quant">
       <option value="-1">all results</option>
       <option value="10">10 results</option>
       <option value="50">50 results</option>
      </select>
      [ <a href="http://server:port/alias/search.wml?search=$(search)&quant=$
     </p>
   </card>
</wml>
```

6.5.2. WML Search Template

A sample WML template (search.wml) can be found in the admin directory of the Alkaline distribution. Such a template constructs a card of search totals followed by a deck of cards containing search results.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="results" title="Search Results">
   <p>
    <b>WAP Search</b>: <!-SEARCH-GENERAL $post.searchconfig -
> <br/>
    <!-SEARCH-GENERAL Alkaline has found $to-
tal page(s) in $time seconds, showing $quant results.-><br/>
    <!-SEARCH-RESULTS->
   </p>
  </card>
  <!-SET MAP-
    <br/>[ <a href="#res$index">next</a> ]
   </p>
  </card>
  <card id="res$index" title="$title">
   <p>
```

```
    <do name="back" type="prev" label="back">
      <prev/>
    </do>
    <do name="link" type="accept" label="$url">
     <go href="$url"/>
    </do>
    <a href="$url">$title</a><br/>
    <small>$header ...</small>
    <br/>
    size: <b>$size</b> bytes<br/>
    modified: <b>$modif</b><br/>
    relevance: <b>$quality</b>%<br/>
  ->
</wml>
```

Chapter 7. Advanced Alkaline Features

7.1. Indexing Other Document Formats

7.1.1. Introduction

Alkaline features include indexing special document formats, such as Adobe PDF or Microsoft Word. To index document formats other than HTML, a *filter* is required.

Alkaline has the ability to preprocess any data retrieved before it is indexed. A document of any format can be passed to any external piece of software, called a filter, transformed by this filter and indexed. Alkaline can perform various tasks using filters. It can obviously index a site with documents of a different format. But as a filter can be invoked on any indexed document, Alkaline offers the unique possibility of implementing such features as mirroring sites or gathering site statistics and information.

Both document and object filters take the contents of a temporary file created by Alkaline, process such contents to produce html output into a second temporary file, read by Alkaline.

It is necessary to instruct Alkaline to retrieve a document of a different type. This is done by using the *ExtsAdd* directive in the asearch.cnf file. For example:

```
ExtsAdd=pdf,doc
```

There’re two sort of Filters in Alkaline: *document filters* and *object filters*. Document filters process documents directly linked from HTML pages. Object filters process embedded objects.

```
Simple PDF document:
<a href="docs/document.pdf">pdf document</a>

Embedded Shockwave Flash object:
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
```

```
 codeBase="http://active.macromedia.com/flash2/cabs/swflash.cab#version=3,0,
 height="100%" id="navig" width="100%">
  <param name="movie" value="navig.swf">
  <param name="loop" value="false">
  <param name="quality" value="autohigh">
  <param name="menu" value="false">
</object>
```

7.1.2. Document Filters

Document filters are defined in the asearch.cnf files by:

```
Filter Extension/Mime Type=command line
```

It is possible to preprocess all documents by omitting the Extension/Mime Type parameter, for example:

```
Filter=/bin/filter $1 $2
```

Some documents are returned without a mime type or have no extension, for example http://www.server.com/ does not imply any extension and might be an HTTP/0.9 compliant server returning HTML contents without the Content-type header. Specifying a filter with no type will catch all these special cases.

To index pdf documents, you must tell Alkaline to retrieve pdf files by adding ExtsAdd=pdf to the asearch.cnf file. An Adobe pdf filter would be used like this (notice that the extension is case-sensitive):

```
Filter PDF=/bin/pdftotext $1 $2
Filter pdf=/bin/pdftotext $1 $2
```

Specifying a case-sensitive extension is not very convenient; you can also specify mime types, for example:

```
Filter Application/Zip=/bin/specialunzip $1 $2
Filter Application/Pdf=/bin/pdftotext $1 $2
```

The variables such as $1, $2 are used to pass parameters to the filter. Available variables are:

Table 7-1. Document Filter Automatic Variables

$0	filename extension without the leading dot (ex: pdf)
$1	temporary file name that contains data remotely retrieved
$2	temporary file name that should contain results generated by the filter
$3	url of the file retrieved, not quoted
$4	data retrieved (use a temporary file, $1, instead)
$5	mime type of the file retrieved if any (such as application/zip), not quoted

7.1.3. Object Filters

Alkaline will process objects of the following format found in the retrieved HTML documents:

```
<object classid="clsid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx">
 <param name="name1" value="value1">
 <param name="name2" value="value2">
 ...
</object>
```

Such objects are embedded in the document. After document filter processing, the resulting output of the filter will be embedded into the document as well.

To make a filter work properly for an embedded object, the following must be included into the asearch.cnf file:

```
# command line to execute for each object of type ClassID
```

```
Object ClassID=command line
# param value to use to retrieve the document
ObjectDocument ClassID=parameter name
```

Here is a real example for filtering Shockwave Flash embedded objects. The shockwave flash CLSID (unique class ID) is clsid:D27CDB6E-AE6D-11cf-96B8-444553540000 and the embedded object at http://www.foo.com/bar/ looks like this:

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
 codeBase="http://active.macromedia.com/flash2/cabs/swflash.cab#version=3,0,
 height="100%" id="navig" width="100%">
  <param name="movie" value="foo.swf">
  <param name="loop" value="false">
  <param name="quality" value="autohigh">
  <param name="menu" value="false">
</object>
```

The document defined for the Shockwave Flash object is in the variable movie, thus the following should be added to asearch.cnf:

```
ObjectDocument clsid:D27CDB6E-AE6D-11cf-96B8-444553540000=movie
```

This will tell Alkaline to retrieve http://www.foo.com/bar/foo.swf as defined by the movie variable in the object with the CLSID "clsid:D27CDB6E-AE6D-11cf-96B8-444553540000".

For the filter to be executed, it is necessary to define a valid command line. Mapping for the command line for objects is more complete than for document filters. All variables defined by the *param* tags are available in addition of

Table 7-2. Object Filter Automatic Variables

sourcefile	filename that contains the retrieved document
targetfile	filename that should contain the filter results
url	URL where the OBJECT tag was found

base	the BASE HREF of the document where the OBJECT tag was found

For example:

```
# place on one single line
Object clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000=/usr/local/bin/swf-filter
 $sourcefile -menu="$menu" > $targetfile
```

The following output was produced by Alkaline when an object filter is invoked:

```
[http://www.foo.com/bar/] (-1) - [639 bytes][0]
 [clsid:D27CDB6E-AE6D-11cf-96B8-444553540000]
 [http://www.foo.com/bar/foo.swf][200 OK][510879 bytes]
 [inf][lnx][md5][vix][keys][mta][ndx][ok]
```

7.1.4. Writing Filters

An Alkaline filter is a simple command line program that takes at least two arguments in any order or format: a file name of the original document and a file name of the output. The output should be text or (partial) HTML. Your filter can generate TITLE and META tags. Punctuation and formatting output by the filter are ignored by Alkaline.

With Alkaline, you can specify a chain of filters or any kind of command processing for your filter files. You can also make a script that will choose whether to process a file or not. In the case when the filter(s) should not translate the file, simply output the exact copy of the original document.

If you plan to write a new filter, make sure you visit http://www.wotsit.org. It is definitely the best source for document formats and available technical resources and format specifications. As you test or write a new filter, please email admin@vestris.com with a detailed description, examples, source and/or binary availability, licensing information, and all other useful links and comments.

7.2. Available Filters

7.2.1. Adobe Pdf (pdf2text and pdf2html)

The *pdf2text* Adobe pdf filter has been successfully tested. It is provided by Derek B. Noonburg <derekn@foolabs.com> from the xpdf tool under the GPL license. You should get xpdf which contains pdftotext and pdfinfo from the FooLabs Site at http://www.foolabs.com/xpdf/.

The pdftotext program accepts two parameters: a source and a target filename. Thus the line to add to your asearch.cnf for the pdf filter looks like this:

```
Filter pdf=/bin/pdftotext $1 $2
```

You can avoid seeing filter errors by adding > *null* on Windows NT or > */dev/null 2>&1* on Unix to the command line:

```
Filter pdf=/bin/pdftotext $1 $2 > /dev/null 2>&1
```

You can use pdftotext along with pdfinfo in order to generate HTML content and benefit from meta keywords, author and document title. Derek’s pdftohtml shell script for UNIX implements those features and is available at http://alkaline.vestris.com/filters/pdftohtml.

You can pickup an executable of pdf2text, gzip and a pdf2html for Windows NT in the Alkaline distribution directory at http://alkaline.vestris.com/files/asearch-distrib/WinNT/pdf2text.zip. The asearch.cnf configuration directives must use fully qualified Windows paths:

```
Filter pdf=c:\tools\pdftotext.exe $1 $2
```

or, if you are using pdf2html:

```
Filter pdf=c:\tools\pdf2html.bat $1 $2
```

7.2.2. Microsoft Word (vwHtml)

The *wvWare* Microsoft Word filter has been successfully tested. It is also known as former MsWordView and is provided by Caol´n McNamara <caolan.mcnamara@ul.ie>. It can be found at http://www.wvware.com/ under the GPL license. WvWare can load and parse the Word 2000, 97, 95 and 6 file formats.

The filter syntax that should be employed is simply

```
Filter doc=/bin/wvHtml $1 > $2
```

7.2.3. Microsoft Rich Text Format (rtf2html)

Rtf2Html is a commercial software from Chris Hector <chris@sunpack.com> and is available at http://www.sunpack.com/RTF/. This filter has not been tested.

7.2.4. LaTex / Tex (LaTex2Html)

TeX is a typesetting system written by Donald E. Knuth. LaTeX is a TeX macro package, originally written by Leslie Lamport, that provides a document processing system. A great Tex FAQ can be found at http://www.tex.ac.uk/cgi-bin/texfaq2html.

A LaTex2HTML converter is available from Nicos Drakos <nikos@mpn.com> of the University of Leeds at http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html. This filter has not been tested.

7.2.5. Word Perfect, AmiPro, Wang WPS (Plus), etc.

A commercial software for Windows, called WebConvert, available from http://www.webconvert.com/, promises to convert most of the major document formats. It can be used from the command line, so it can run as an Alkaline filter. It has not been tested.

7.2.6. Shockwave Flash

You can grab the source code in C of a tested Shockwave Flash decoder at http://alkaline.vestris.com/filters/swfparse.cpp. A volunteer is welcome to transform this into a real filter that would extract text and links.

7.2.7. Extensible Markup Language (Xml)

Xml documents are structured differently and need some processing in order to be indexed and useful. This can be done with a freeware tool Xml2, by Dan Egnor, at http://ofb.net/~egnor/xml2/. Also make sure to check Pixie, by Sean McGrath, at http://www.xml.com/pub/a/2000/03/15/feature/, an open source XML processing library.

To generate meta tags from the output of xml2, use the following xml2html.awk script:

```
/[^\=]*/ {
    start = index($0, "\=")
    NAME = substr($0, 2, start - 2)
    gsub("/", "-", NAME)
    gsub("\@", "-", NAME)
    gsub("-", "-", NAME)
    CONTENT = substr($0, start + 1, length($0))
    print "<meta name=\"" NAME "\" content=\"" CONTENT "\">"
    next
}

{
    print
}
```

For example

```
Filter xml=/usr/bin/xml2 < $1 | awk -
f /usr/bin/xml2html.awk > $2
```

7.2.8. MPEG Layer 3 Music (Mp3)

MPEG layer 3 is a type of audio codec where processed by significant compression from the original audio source with very little loss in sound quality.

Mp3 files have a blob of data associated to them, called ID3. This information can contain the song title, artist, album and more. There’re hundreds of programs that provide extraction of those tags. For the simplest Mp3 indexing we suggest Id3Tool by Chris "Crossfire" Collins (http://kitsumi.xware.cx). You can download this tool as source code or binary format at http://freshmeat.net/projects/id3tool/

In addition, Mp3 encoding information can be retrieved using Mp3Header by Owen Llyod (http://owl.yi.org), available in source or binary format at http://owl.yi.org/programs/#mp3header.

The output of id3tool and mp3header is typically:

```
$ id3tool "02 Les Enfoires.mp3"

Filename: 02 Les Enfoires.mp3
Song Title:     Quand on n’a que l’amour
Artist:         Céline Dion & Mauranne
Album:          La soirée des enfoirés 96
Note:           Profits aux Restos du coeur
Year:           1996
Genre:          Chanson (0x66)
```

```
$ mp3header "02 Les Enfoires.mp3"

02 Les Enfoires.mp3 - File Data
---------------------
File Size:      4188160 bytes
Est. Time:      209 secs
MPEG Version:   1
MPEG Layer:     III
BitRate:        160 kBit/s
Sample Freq:    44100 kHz
Padding:        No
Mode:           Joint Stereo
```

```
Private:        No
Copyright:      No
Orginal:        No
Emphasis:       None
```

This needs to be transformed into html format and can be done using the following awk script:

```
function trim(input)
{
    result = ""
    n = split(input, words, " ")
    for (i = 1; i <= n; i++)
    {
       if (words[i] != " ")
       {
           if (length(result) > 0)
           {
                result = result " "
           }
           result = result words[i]
       }
    }
    return result;
}


{
    start = index($0, ":")
    if (start == -1)
    {
        next
    }
    NAME = trim(substr($0, 0, start - 1))
    CONTENT = trim(substr($0, start + 1, length($0)))
    if (NAME == "Song Title")
    {
         print "<title>" CONTENT "</title>"
    }
    else if (length(NAME))
    {
         print "<meta name=\"" NAME "\" content=\"" CON-
TENT "\">"
    }
    next
}
```

The complete filter command line looks like this:

```
Filter mp3=id3tool $1 | awk -
f mp32html.awk > $2 ; mp3header $1 | awk -f mp32html.awk » $2
```

The output of the line above command for this Mp3 file can be indexed by Alkaline:

```
<meta name="Filename" content="02 Les Enfoires.mp3">
<title>Quand on n’a que l’amour</title>
<meta name="Artist" content="Céline Dion & Mauranne">
<meta name="Album" content="La soirée des enfoirés 96">
<meta name="Note" content="Profits aux Restos du coeur">
<meta name="Year" content="1996">
<meta name="Genre" content="Chanson (0x66)">
<meta name="File Size" content="4188160 bytes">
<meta name="Est. Time" content="209 secs">
<meta name="MPEG Version" content="1">
<meta name="MPEG Layer" content="III">
<meta name="BitRate" content="160 kBit/s">
<meta name="Sample Freq" content="44100 kHz">
<meta name="Padding" content="No">
<meta name="Mode" content="Joint Stereo">
<meta name="Private" content="No">
<meta name="Copyright" content="No">
<meta name="Orginal" content="No">
<meta name="Emphasis" content="None">
```

Use the CustomMetas directive to expose required meta tags to the results page.

7.2.9. Other Sources

A site worth visiting for filters is http://www.w3.org/Tools/Filters.html. It has an extensive list of filters available.

Keypack (http://www.keypak.com/) and Blueberry Filtrex (http://www.blueberry.com/) claim to convert a huge amount of document formats.

7.3. Alkaline Robots, HTML and Meta Tags

7.3.1. Alkaline Robot Support

Alkaline fully supports robot directives described at the WebCrawler robots pages, http://info.webcrawler.com/mak/projects/robots/robots.html. Alkaline is a registered bot with a user-agent string: *AlkalineBOT/1.9*.

This includes full compliance with the /robots.txt directives, including the User-agent and Disallow restrictions.

Alkaline will not follow links if a <meta name="robots" content="nofollow"> tag is found. Alkaline will not index document contents if a <meta name="robots" content="noindex"> tag is found.

Alkaline robots support can be disabled for individual configurations by specifying *Robots=N* in the asearch.cnf file.

7.3.2. Alkaline Specific Meta Tags

Alkaline will look for specific meta tags in a document. Each meta tag is of format <meta name="alkaline" content="...">. The value of the meta tag can contain multiple elements separated by spaces and can be the following:

Table 7-3. Alkaline Specific Meta Tags

skip	skip indexing of the page, it will not be referenced
skipmeta	skip indexing of meta tags on the current page
skiplinks	do not gather links from the currently indexed page
skiptext	do not index free text on the current page

A <meta name="alkaline" content="skip"> tag will instruct Alkaline not just to avoid indexing the page, but also not to gather links from it. If you do not want the page to be indexed, but the links to be gathered, use <meta name="alkaline" content="skiptext skipmeta">.

If you with to exclude a pattern of pages from indexing but with links to be gathered, use the UrlIndex and/or the UrlSkip directives.

Example:

```
<meta name="alkaline" content="skipmeta skiplinks">
```

7.3.3. Alkaline Specific Html Tags

Alkaline will look for specific html tags in a document. These can include one-another. An Alkaline specific tag is always <*alkaline ...*> <*/alkaline*>.

Table 7-4. Alkaline Specific Html Tags

<alkaline skip>	skip the indexing of the section terminated by </alkaline>

<alkaline url="url">	add a link from the current page manually; useful for pages that generate, for example, JavaScript code that cannot be correctly interpreted by the parser; there's no need to terminate this tag with </alkaline>

Example:

```
<alkaline skip>John, please read this page!</alkaline>
<alkaline url="http://www.foo.com">
```

7.4. Online Administration and Statistics

7.4.1. Accessing the Online Administration

To access the online management, browse to the Alkaline server using the port on which Alkaline is running with no parameters. You will be redirected to the /admin path. For example, the Vestris Alkaline Management console can be found at http://search.vestris.com/admin/. You can test it with username *alkaline-manage* and the *manage* password.

The administrative section is password protected. A valid username and password can either be the Alkaline *root* or *alkaline-manage*. Alkaline users and passwords are described in *Chapter 3: Alkaline Server Configuration*.

It is also possible to login with the username *vestris* and the full certificate unlock key as the password which is available to the server administrator and Vestris Inc. only. All other locations, except server statistics, such as adding, removing and reindexing URLs require a real password and will not accept the certificate unlock key.

7.4.2. Server Parameters

Server statistics are available by selecting the *Server Parameters* tab in the menu.

7.4.2.1. Main Page

The *Main Page* contains links to the various sections of the Alkaline management section, web site and documentation.

7.4.2.2. Performance Counters

The *Performance Counters* runtime information includes the server version, system information, runtime dates and times, search thread count, search requests and the requests per minute rate. Note that this information may vary depending on the platform or version of Alkaline.

7.4.2.3. Global Configuration

The *Global Configuration* section includes the options from the global.cnf configuration file. The *Loaded : Yes* value shows whether a global.cnf configuration file was found and properly loaded.

7.4.2.4. Server Configurations

The *Server Configurations* menu contains the currently loaded configurations. Each page offers a basic search form, an option to reload the index for this particular section and the entire list of options set in the asearch.cnf file, including virtual configurations. In addition, this page contains the statistics of the search cache, including the latest search terms.

7.4.3. Server Operations

Such operations include restarting the server, adding or deleting urls.

Some of these options might not be available depending on the platform or version of Alkaline

7.4.3.1. Restart Server

Use this command to restart the index/search server. This will force the reload of the indexes and of all configurations - equivalent to a full interrupt/restart of Alkaline. The availability of this command may vary on different operating systems. Either the *root* or the *alkaline-restart* username with a valid password must be supplied for this operation.

7.4.3.2. Shutdown Server

Use this command to shutdown the index/search server. This will terminate Alkaline. The availability of this command may vary on different operating systems. Either the *root* or the *alkaline-restart* username with a valid password must be supplied for this operation.

7.4.3.3. Refresh Template

Alkaline stores search templates in memory for a better server availability. Select one of the currently cached templates from the list and use this command to reload it. Either the *root* or the *alkaline-manage* username with a valid password must be supplied for this operation.

7.4.3.4. Add / Reindex an Url

Add/reindex an url to the selected configuration. This operation adds a new url and immediately schedules it for reindexing (should be reindexed almost instantly). If the url is already present, it will simply be reindexed. The url will appear in the index only and will not depend on the configuration. Its links will not be followed by the indexing daemon. To keep track on urls that users add to the search engine, use the log files. Either the *root* or the *alkaline-add* password must be provided for this operation.

7.4.3.5. Delete an Url

Remove an url from the selected configuration.

If the url has been cached, it might take some time before it disappears from search results.

To keep track on URLs that users remove from the search engine, use the log files. If the url is linked from some other document, the indexing daemon might find it again and successfully reindex it. Either the *root* or the *alkaline-add* password must be provided for this operation.

7.5. Running Alkaline as a Windows NT/2000 Service

You may want to run Alkaline as a Windows NT service. This allows automatic Alkaline startup regardless whether a user interactively logs in or not, as well as a transparent control over the Alkaline's server start and termination. It also allows to run Alkaline under an account that has full rights to the relevant directories without having to grant any interactive user these rights.

7.5.1. Installation

Alkaline for Windows NT has service functions built in. The syntax is *asearch service [command] [[port] ...]*.

Before attempting to install Alkaline as a Windows NT service, make sure that you are able to run it from command line, search and index in background.

- To install Alkaline as a Windows NT service, run *asearch service install ...*. For example

  ```
  asearch service install 9999 as -no404
  ```

This will create a service named *AlkalineSE* and set it's startup to *Automatic*, so that the service is started whenever the computer reboots. Alkaline will run from the directory you have installed it from.

- To remove the Alkaline Windows NT service, run *asearch service remove*.
- To start the Alkaline Windows NT service from command line, run *asearch service start*.
- To stop the Alkaline Windows NT service from command line, run *asearch service stop*.

Alkaline needs to load its indexes at startup, so it might take some time to start. You can check the asearch.exe process with the Task Manager.

7.5.2. Options

When performing service operations, such as install, remove, stop and start, you can specify various options:

- *–servicename=string*: service name, default is AlkalineSE; this allows to install more than one service on the same server
- *–servicedisplayname=string*: service display name, default is *Alkaline Search Engine*; to specify strings with spaces, quote the entire option, for example *"–servicedisplayname=Alkaline Search for Foo.com"*

- *–serviceusername=domain\username*: username to run the service with, default is localsystem; note that the account you are using must have Logon as a Service rights
- *–servicepassword=password*: password to run the service with, default is none (localsystem password is controlled by the system)
- *–servicedescription=description*: description for this service instance

To change existing options without removing and reinstalling the service, run *regedt32.exe* or *regedit.exe* and open the *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AlkalineSE* registry key.

- During installation, Alkaline creates an *ImagePath* value which should never be changed. It’s value is *asearch service dispatch* and tells Alkaline to run as a Windows NT service process.
- The *Parameters\StartDirectory* key defines the initial directory to run Alkaline from. When the service is started, Alkaline will change the current directory to the one defined by this parameter.
- The *Parameters\StartArguments* key defines the port value and the aliases to load when the search engine is started. This is typically something like *9999 alias1 alias2*.
- The *Parameters\StartOptions* key defines all additional options to run Alkaline with. The values have the dashes stripped. This value might be empty.

The easiest way to change the Alkaline’s startup parameters is to re-run it with the *service install* options. The service creation will fail with an error message, but the options will be updated.

7.5.3. Troubleshooting

7.5.3.1. CreateService failed. Overlapped I/O operation is in

progress.

This error appears when attempting to install Alkaline. The service is already installed or has been removed and a Services Control Panel is still opened, which will mark the service for deletion but will remain pending till all instances of the service manager are closed. Close the registry editor and all Services Control Panels if any opened and retry.

7.5.3.2. Attempting to start AlkalineSE with no further error message.

This appears when attempting to start Alkaline as it is already running or when the service does not exist, no action has been performed.

7.5.3.3. ControlService failed. The service has not been started. The specified service does not exist as an installed service.

This error message appears when stopping the Alkaline service without starting it or when attempting to remove a service that does not exist.

7.5.3.4. Unable to start service, the parameter is incorrect.

One of the current directory, parameters or options contains an invalid value. Make sure that you have specified the correct parameters and that you can start Alkaline from the command line. Reinstall the service.

7.6. Alkaline Virtual Memory and Swap

Any operating system has a fixed amount of physical memory available. Usually, application need more than the physical memory installed on your system, for that purpose the operating system uses a swap mechanism: instead of storing data in physical memory, it uses a disk file.

On operating systems, such as Windows NT, Windows 2000 or UNIX, the memory is logically divided in pages. When the system needs a certain portion of memory which is currently in the swap (this is called a page fault) it will load all the corresponding pages into RAM. When a page is not accessed for a long time, it is saved back to disk and discarded.

If you look on the Windows NT Task Manager or the output from ps or top, Mem Usage is the working set size. It is the amount of physical memory which is directly (currently) allocated to the process. It can be accessed without causing a page fault. This includes pages shared with other processes. The Windows NT VM Size or the UNIX RSS/RES value is the total private virtual memory allocated to the process.

Alkaline has a built in swapping mechanism that uses memory mapped files to simulate virtual memory. Running Alkaline with *–enableswap* may save you up to 75% of physical memory but may as well significantly decrease indexing and searching performance. The *–enabelswap* command line option forces Alkaline to create a file, usually of a *blib-pid-index* format. On Windows NT, you will usually see one or more swap files of 32 MB each with a growing index number in the default temporary directory. On a UNIX system, only one single file is created and you can specify the filename. For example, use *–enableswap=/tmp/alkaline.swp*. This option is especially helpful because Alkaline does not get a chance to cleanup its swap file if the Alkaline process is killed on a UNIX system. The operating system handles automatic temporary file removal on Windows NT.

You can consider that Alkaline allocates the amount of memory shown under VM Size and RSS/RES. The excess of memory shown in the Mem Usage is all the swapped data that is currently resident because more physical memory is available or because that data is required by the program to run.

Try running Alkaline with *–enableswap* and notice the large differences in VM Size and RSS/RES values.

7.7. Indexing Guidelines

7.7.1. CGI-powered and Dynamic Sites

Alkaline has been intensively used with sites that are dynamically generated or contain cgi scripts. Several things should be done or taken in consideration when configuring Alkaline in order to avoid frequent questions or problems.

Usually, cgi requests are made on scripts that might have a different extension, for example somecgi.exe has a .exe extension or somecgi.pl, has a .pl one. These extensions must be added to the configuration with *ExtsAdd=exe* or *ExtsAdd=pl* or both, *ExtsAdd=pl,exe*.

By definition, a cgi request is identified by a trailing list of parameters after the ? character, for example */cgi-bin/foo.pl?name=value*. With a default configuration, Alkaline ignores such requests. You must add *Cgi=Y* to the configuration file.

Dynamically generate pages, except for Active Server Pages (.asp) often do not output a Content-Length field and never comply to *If-Modified-Since* headers. It might be judicious to add *Expire=Y* (or run Alkaline with *-expire*) in order to index all pages and avoid Alkaline spending time finding the last modified date for a document.

The cgi parameters are often case-insensitive, especially for .exe scripts under Windows NT. You might want to consider urls case-insensitive. Add *Insens=Y* to the configuration file.

A lot of cgi scripts generate dynamic data as a result of an html form post. Alkaline cannot simulate a post method neither fill a form for the user. You might want to add some links manually or using an <alkaline url=...> tag.

7.7.2. Indexing in Background

Background indexing is one of the most powerful Alkaline features. It allows to continuously index a site and make changes available to the search engine instantly. Some items should though be considered very seriously especially for heavy traffic sites and sites indexing large amounts of data.

Background indexing is disabled by setting *Reindex=N* in all configuration files or by running Alkaline with *–noreindex* option. The later is the preferred way as no background thread is created when *–noreindex* is specified on the command line.

A heavy traffic search site would have over a request every two seconds. Alkaline is known to handle 3-5 requests a second depending on the hardware configuration.

Everything in Alkaline is done to favor search speed and not background indexing. This includes regular checkups of search activity from the background thread in order to pause the later as soon as possible. Still, the background indexing thread and the search thread manipulate the same data and both will lock each other depending on the access needs.

Enabling background indexing implies at least 10-25% more memory usage and a much higher CPU activity, reaching 70% of CPU average usage compared to 3-5% without the background indexing thread.

Because of interlocked architecture of Alkaline, enabling background indexing means degrading search performance. Search performance will degrade by at least 25% during normal background reindexing. Search operations will degrade by up to 95% when writing indexes, thus background indexing is not advised on sites with over 50’000 documents.

If you enable background indexing on a huge site, be sure to use *SleepFile* and *SleepRoundtrip* parameters as they can increase performance of the search front-end dramatically.

Alkaline is capable of serving thousands of requests per minute, that is many requests a second. Since Alkaline is a full HTTP server and pools requests, it very unlikely that responsiveness to the clients would be an issue. Disabling the background indexing is the first thing to do when speed becomes a question or when you are having trouble keeping the engine stable.

7.7.3. Lotus Notes Domino

Alkaline fully supports indexing and searching of Lotus Notes Domino sites. Indexing Lotus Notes Domino generated sites requires several additional options to be added to the configuration files.

Domino requests might look like a simple demand to a .nsf file, adding *AddExts=nsf* to the configuration files will tell Alkaline to index documents with .nsf extension. All Domino requests look like cgi requests, so it is necessary to add *Cgi=Y*.

Moreover, Domino can generate multiple views for same content. Alkaline includes an expansion feature that will transform any page with collapsed elements into a request to it's expanded form. To enable this, add *Nsf=Y* to the configuration. The Nsf option will also enable lookup of full duplicates, that is pages that have different urls but the same content. Domino generates urls in all possible forms and shapes, often leading to the same content.

Domino is not case-sensitive for urls, but all UNIX servers are. Alkaline will treat /Foo and /foo as two different links. Thus *Insens=Y* must be added. Domino might generate empty links leading to pages with junk (which is probably a bug). Such a link looks like <a href=url> and is neither clickable, nor has any text. Alkaline will still follow such links unless you specify *EmptyLinks=N*.

Here's an example of a configuration file for the Geneva Hospital Domino web server:

```
Remote=N
UrlList=http://www.hug-ge.ch
Depth=-1
MaxFiles=-1
SleepRoundtrip=21600
CGI=Y
AddExts=nsf
Insens=Y
NSF=Y
NoEmptyLinks=Y
```

7.7.4. Mirroring Sites

It is possible to combine Alkaline filter features and the Alkaline spider to mirror an entire set of web sites.

Create the following *url2path* perl script and ensure that you can run it on your server. This script transforms an url into a fully qualitified path and issues the required commands to copy a file in the newly created directory hiararchy. This scrpt has been tested on a Linux server.

```
#!/usr/bin/perl

use Carp;
use URI;

@ARGV == 3 or Carp::croak ’usage: url2path [url] [source] [target]’;

my $url = URI->new($ARGV[0]);
my @path = $url->path_segments;
my $relative = $url->host;

for ($i = 0; $i < $#path; $i++)
{
      my $segment = $path[$i];
      $relative = $relative.$segment.’/’;
}

my $prefix = $ARGV[2];
my $fullpath = $prefix.$relative;
my $fullfile = $fullpath.$path[$#path];

if (length($path[$#path]) == 0)
{
     $fullfile = $fullfile."index.html";
}

print "url2path: saving ".$fullfile."\n";
```

```
system("mkdir -p ".$fullpath);
system("cp ".$ARGV[1]." ".$fullfile);
```

Use a regular asearch.cnf to invoke the script for each newly downloaded document using the *Filter* directive, similar to this one:

```
UrlList=http://www.foo.com/
SkipText=Y
SkipMeta=Y
Filter=/usr/bin/url2path.pl $3 $1 /home/mirrors/ ; mv $1 $2
ExtsAdd=jpg,gif
```

7.7.5. Indexing Local File System

Alkaline supports indexing of local file system files since version 1.7. To index a particular directory and all its subdirectories, use the *file://* format for your urls. For example:

```
# UNIX asearch.cnf
UrlList=file:///home/user/
```

```
# Windows asearch.cnf
UrlList=file://d:\documents\web\
```

Alkaline will treat directory listings as a document and index it according to normal rules. All configuration options valid for http:// urls fully apply with file:// urls, this includes extensions to be indexed and links to follow.

Search results returned are of the file:// url format as well. To index a web site that is stored locally, use the ReplaceLocal directive and an http:// format url. To index web content that is not linked to each-other, use the file:// url format and the Replace directive to render search results.

7.8. Working With Us

7.8.1. Reporting Problems

During the product release cycle, the Alkaline development and test teams work closely together to ensure the highest quality of each new version. Although the hundred percent of known issues are addressed in the development timeframe, Vestris Inc. acknowledges the possibility of code defects and/or interoperability issues in production builds.

When you experience problems with our software, please do make absolutely sure that you are looking at a consistent, unexpected defect. Do read the documentation and check that the answer to your problem is not available in the product's frequently asked questions at http://alkaline.vestris.com/docs/alkaline-faq/index.html. Check the latest Alkaline *What's New* at http://alkaline.vestris.com/whatsnew.html for the problem you are experiencing. Download and try the latest beta from http://alkaline.vestris.com/files/asearch-distrib/NextReleaseBeta/, when available.

If you have a question, issue or believe that you have found an Alkaline bug, please do report it on our website at http://www.vestris.com/vestris/bugreport.html. We strongly encourage you to explain your problem in the shortest and most concise way possible. The defects that can be reproduced consistently are those fixed fastest. After we receive a problem, we will contact you with more questions, a fixed version or an alternative solution, in the best timeframe possible.

7.8.2. Built-in Tracing

A member of our staff may ask you to enable the so-called tracing mechanism, built-in Alkaline. Tracing is a lightweight replacement for a debugger, that allows developers to collect internal output from Alkaline without ever connecting to the customer's machine or requiring any special access. Shortly, when tracing is enabled, a huge amount of internal information is output to the console of a running Alkaline. This output can dramatically help diagnose problems.

For example, if search consistently produces a fatal error that forces Alkaline to terminate, enabling tracing for the search operations will provide output for the granular progresses of such, including retrieving data sets, sorting results and rendering search pages. A developer can then isolate and fix the problem.

A trace belongs to a particular tag and is output at a particular level. There're several levels of tracing, such as crash, error, warning, information and verbose. There're many tags, such as indexing, search and http layer. To see all available tags and flags in your build of Alkaline, run

```
$ ./asearch tracing
[tracing tags (-tracetags=list)]
1: reserved tag
2: generic library functions
3: general system functions
4: memory allocation system requests
5: socket system level requests
6: http requests and responses
7: locks, semaphores and mutexes
8: dns lookups, name resolution
9: tcp server
10: reserved tag
11: reserved tag
12: reserved tag
13: generic engine tag
14: merge
15: excludewords
16: search
17: remove
18: index
19: client and admin
20: reserved tag
[tracing levels (-tracelevels=max)]
1: crash
2: error
3: warning
4: information
5: verbose
```

To enable tracing, Alkaline has two command line parameters: *–tracetags=tag1[,tag2,...]* and *–tracelevels=max*. For example, the search tag is 16 (from the output above). The most verbose trace level is 5. To see all internal search operations, the following command line should be used:

```
./asearch ... -tracetags=16 -tracelevels=5
```

To enable tracing for both HTTP and DNS requests, the following command line should be used:

```
./asearch ... -tracetags=6,8 -tracelevels=5
```

Your console will be filled with messages about all outgoing and incoming HTTP requests along with the asynchronous domain name service queries.

The tag numbers may change from one build to another as the development team adds new tags or more tracing information. The *tracing* command-line argument always provides the right tag numbers.

To collect tracing information, simply redirect the output to a file. For example

```
./asearch ... -tracetags=4,12,14 -tracelevels=5 > trace-Oct5-
2001.txt
```

7.8.3. Providing Vestris Inc. with Server Access

Our customers often provide us with access to hardware we cannot afford, various operating systems and other fancy machines for Alkaline builds. In some situations the developer cannot locate, diagnose or fix the problem the customer is seeing on his server. When either of this happens, a member of the Vestris Inc. staff may request a telnet (terminal) access to your server. We sincerely appreciate this kind of help from our partners and customers as it allows us to both continue releasing free versions of Alkaline for non-commercial organizations and fix platform-related problems quickly.

We usually request an access for a plain user account *dblock* (for historical reasons). You have the choice for telnet or ssh. If you wish to setup server ip exclusion, please use the address of *ns.vestris.com*.

We will never request *root* access to your server. We will never give, distribute or disclose customer's or access information. We always use strong passwords and never store them on paper or disk.

If you are giving us access for debugging, please check whether you have a working *gdb* debugger. You can download GNU software from http://www.gnu.org Simply type

```
$ gdb
GNU gdb 4.18
Copyright 1998 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public Li-
cense, and you are
welcome to change it and/or distribute copies of it under cer-
tain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show war-
ranty" for details.
This GDB was configured as "i386-redhat-linux".
(gdb)
```

If you are giving us access for building Alkaline, please ensure that you have a working *cvs* from http://www.cvshome.org.

```
$ cvs -version
Concurrent Versions System (CVS) 1.10.6 (client/server)
```

Check that you have *gcc 2.8.0* or above, or *egcs*.

```
$ gcc -version
egcs-2.91.66
```

Check that you have GNU make. You may have a different *make* and *gmake*, we just need one GNU make.

```
$ make -version
make: Warning: Ignoring DistributedMake -v option
```

```
$ gmake -version
GNU Make version 3.77, by Richard Stallman and Roland McGrath.
```

As we build Alkaline on your server, a copy of the entire source tree will be placed on it. We realize that having an administrative access to the account and/or physically hosting the machine gives you full access to all the files. As Alkaline is not an open-source initiative, it is mutually understood that you must not distribute, alter, publish or copy this source.

Chapter 8. Alkaline Tools

8.1. Introduction

Alkaline tools is distributed separately and is free. An OS-specific package can be found in the same directory as the binary distribution of Alkaline for your platform at http://alkaline.vestris.com/files/asearch-distrib/.

8.2. NdxScan - most frequent words dump

8.2.1. Introduction

NdxScan parses and analyses siteidx?.ndx files. It allows to create exclusion dictionaries based on existing indexes and gather statistics on most frequently indexed word forms.

8.2.2. Usage

```
Alkaline NDXScan 1.1 (c) Vestris Inc. 1994-
2000 All Rights Reserved
parses the NDX file and sorts words by frequency, most fre-
quent first
usage: ndxscan [-s -tX] siteidx?.ndx
        -s: silent, no headers
       -
tX: where X is the occurrences threshold (positive number)
```

8.2.3. Example

```
bash$ ndxscan as/siteidx3.ndx -t83
Alkaline NDXScan 1.1 (c) Vestris Inc. 1994-
2000 All Rights Reserved
[**********][sorting][done]
86      to
86      All
86      com
86      Home
86      Search
86      Support
83      rights
83      reserved
83      CONTENT-TYPE:html
83      CONTENT-TYPE:text
83      CONTENT-TYPE:charset
```

8.3. UrtList - indexed/found urls dump

8.3.1. Introduction

UrtList parses a siteidx?.urt file and uses the siteidx?.ndx file depending on the options selected. It allows to output all urls available in the database, sort them or output only indexed urls.

8.3.2. Usage

```
Alkaline URTList 1.0 (c) Vestris Inc. 1994-
2000 All Rights Reserved
parses the URT file and outputs all urls
```

```
usage: urtlist [-iso] siteidx?.urt
       -s: silent, no headers
       -i: show indexed urls only
           (expects siteidx?.ndx in the same path)
       -o: sort
```

8.3.3. Example

```
bash$ urtlist as/siteidx1.urt -io
Alkaline URTList 1.0 (c) Vestris Inc. 1994-
2000 All Rights Reserved
[**********][956 lines]
[**********][90/955 indexed urls]
[sorting]
http://www.foo.com/
http://www.foo.com/default.htm
http://www.foo.com/business/
http://www.foo.com/college/
http://www.foo.com/bar/
http://www.foo.com/bar/index.html
```

8.4. LnxDump - cross-reference url dump

8.4.1. Introduction

LnxDump uses the siteidx?.urt and siteidx?.lnx files. It allows to output all urls linked from a document and all documents that link to an url.

8.4.2. Usage

```
Alkaline LNXDump 1.0 (c) Vestris Inc. 1994-
2000 All Rights Reserved
parses the URT and the LNX files and outputs links
usage: lnxdump <options> <index-location> <url>
        -s:     silent, no headers
        -f:     forward, all pages linked from url (default)
        -r:     reverse, all pages linking to url
```

8.4.3. Example (Forward)

All pages that http://alkaline.vestris.com/index.html links to (all direct links on the page):

```
bash$ lnxdump vestris http://alkaline.vestris.com/index.html
http://alkaline.vestris.com/
http://alkaline.vestris.com/sites.html
http://www.vestris.com/
http://www.vestris.com/tech/split.html
...
```

8.4.4. Example (Reverse)

All pages that link directly to http://alkaline.vestris.com/:

```
bash$ lnxdump -r vestris http://alkaline.vestris.com/
http://www.vestris.com/index-full.html
http://alkaline.vestris.com/sites.html
http://www.vestris.com/software/software.html
http://www.vestris.com/sti/company.html
...
```

8.5. MrtgStats - mrtg-compatible statistics

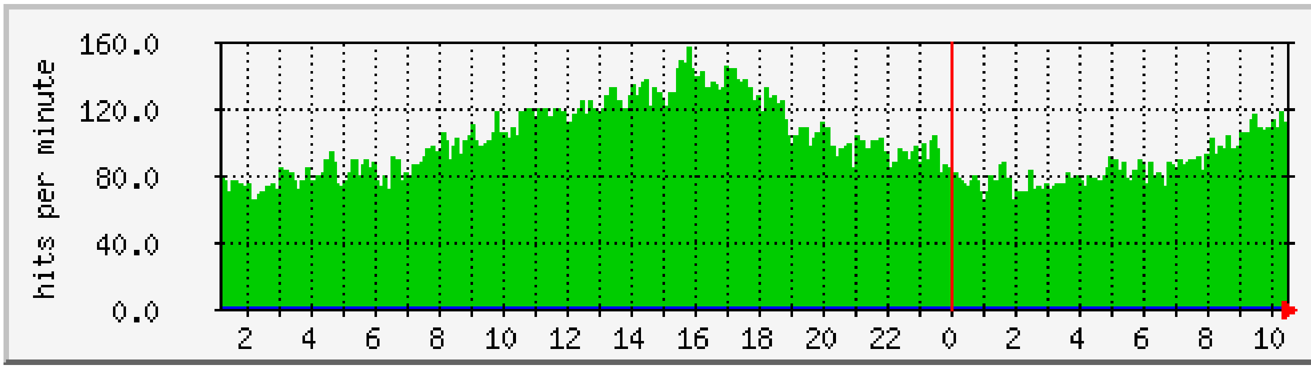
8.5.1. Introduction

The Multi Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network-links. It is widely used around the world to generate html pages containing gif images which provide a live visual representation of this traffic. MRTG is mainly a Swiss open-source product by Tobias Oetiker <oetiker@ee.ethz.ch> and Dave Rand <dlr@bungi.com> from the University of Zurich. Check http://www.mrtg.org/ for how to download and install MRTG.

MrtgStats is a small module which can query a running Alkaline and output MRTG-compliant statistics, which can be graphed using MRTG. You must be running Alkaline 1.3.0904.0 or later.

MRTG produces fully configurable and customizable graphics, such as shown in Figure 8-1.

Figure 8-1. Live Alkaline MRTG Statistics



8.5.2. Installation

Download and install MRTG from

http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html.

The MrtgStats syntax is the following:

```
mrtgstats server:port username password xml-field
```

The server runs Alkaline on the specified port. The username is the one to access the admin section (root or alkaline-manage), so is the password. The xml-field defines the source data that is being monitored.

You can query the following xml data:

Table 8-1. Performance Counters

/alkaline/server/pool/uptime	server uptime in seconds
/alkaline/server/pool/threads	current running threads
/alkaline/search/requests	total search queries
/alkaline/search/rpm	requests per minute
/alkaline/system/cpu/system	kernel CPU time (not available on all platforms)
/alkaline/system/cpu/user	user CPU time (not available on all platforms)
/alkaline/system/cpu/time	shared kernel/user CPU time (not available on all platforms)

Make sure that you can access the performance counters in the admin section with this username and password pair.

Create a search.cfg configuration file in the same directory as other .cfg MRTG configuration files, which looks like this:

```
WorkDir: /home/webroot/mrtg

Title[fooaccess]: stats for www.foo.com:9420
# place on a single line
Target[fooaccess]: ‘mrtgstats
```

```
 www.foo.com:9420 alkaline-
manage pass /alkaline/search/requests‘
MaxBytes[fooaccess]: 100000
PageTop[fooaccess]: search count on www.foo.com
Options[fooaccess]: growright perminute
ShortLegend[fooaccess]:
YLegend[fooaccess]: hits per minute
LegendO[fooaccess]:
LegendI[fooaccess]:  search count:
Legend1[fooaccess]: search count
Legend2[fooaccess]:

Title[foothreads]: stats for www.foo.com:9420
# place on a single line
Target[foothreads]: ‘mrtgstats
 www.foo.com:9420 alkaline-
manage pass /alkaline/server/pool/threads‘
MaxBytes[foothreads]: 100000
PageTop[foothreads]: thread count on www.foo.com
Options[foothreads]: growright gauge
ShortLegend[foothreads]:
YLegend[foothreads]:
LegendO[foothreads]:
LegendI[foothreads]:  thread count:
Legend1[foothreads]: thread count
Legend2[foothreads]:
```

This is just an example. You should replace the paths and the parameters to mrtgstats by your own values.

Setup a crontab entry or a schedule entry under Windows NT such as:

```
# place on a single line
0,5,10,15,20,25,30,35,40,45,50,55 * * * *
mrtg /home/webroot/mrtg/search.cfg > /dev/null 2>&1
```

MRTG will run mrtgstats every five minutes and create searchhits.html in the same directory as the rest of MRTG files with graphical output of the search engine statistics. These are the rpm (requests per minute) and the threads count, which closely matches your server's average load.

8.6. Perl Modules - open API

8.6.1. Introduction

The Perl framework of controlling Alkaline is an extensible set of Perl source code modules. You can easily extend the existing functionality by implementing your own functions.

8.6.2. Alkaline::Server

Alkaline::Server implements server control functions. To install it, run

```
perl Makefile.PL
make
make install
```

8.7. AlkalineStop.pl - stop the running daemon

8.7.1. Introduction

AlkalineStop.pl uses Alkaline::Server to stop a running daemon.

8.7.2. Usage

```
perl AlkalineStop.pl [server:port] [user] [password]
```

8.8. AlkalineReloadIndex.pl - reload an index

8.8.1. Introduction

AlkalineReloadIndex.pl uses Alkaline::Server to reload one or all indexes of a running daemon.

8.8.2. Usage

```
perl AlkalineReloadIndex.pl [server:port] [user] [pass-
word] <[configuration]>
```

8.8.3. Notes

The configuration is optional. If omitted, all indexes will be reloaded. This script is fully supported since Alkaline 1.41.0411.0.

8.9. AlkalineAddUrl.pl - add an url to an

existing index

8.9.1. Introduction

AlkalineAddUrl.pl uses Alkaline::Server to add a new url to an existing index of a running search engine server.

8.9.2. Usage

```
perl AlkalineAddUrl.pl [server:port] [user] [password] [con-
fig] [url]
```

8.9.3. Notes

This script is fully supported in Alkaline 1.7.1213.0.

8.10. AlkalineDeleteUrl.pl - remove an url from an existing index

8.10.1. Introduction

AlkalineDeleteUrl.pl uses Alkaline::Server to remove an url from an existing index of a running search engine server.

8.10.2. Usage

```
perl AlkalineDeleteUrl.pl [server:port] [user] [password] [con-
fig] [url]
```

8.10.3. Notes

This script is fully supported in Alkaline 1.7.1213.0.

8.11. AlkalineGetStats.pl - get statistics in xml format

8.11.1. Introduction

AlkalineGetStats.pl uses Alkaline::Server to retrieve the server statistics in xml format.

8.11.2. Usage

```
perl AlkalineGetStats.pl [server:port] [user] [password]
```

8.11.3. Notes

This script is fully supported in Alkaline 1.7.1213.0.

8.12. AlkalineRefreshTemplate.pl - refresh a

search results template

8.12.1. Introduction

AlkalineRefreshTemplate.pl uses Alkaline::Server to reload a search template on a running server.

8.12.2. Usage

```
perl AlkalineRefreshTemplate.pl [server:port] [user] [password] [template]
```

8.12.3. Notes

This script is fully supported in Alkaline 1.7.1213.0.

8.13. AlkalineRestart.pl - restart the server

8.13.1. Introduction

AlkalineRestart.pl uses Alkaline::Server to restart the search engine server.

8.13.2. Usage

```
perl AlkalineRestart.pl [server:port] [user] [password]
```

8.13.3. Notes

This script is fully supported in Alkaline 1.7.1213.0. The restart feature is supported on selected UNIX platforms only.

8.14. AlkalineUnlock.pl - submit a certificate

8.14.1. Introduction

AlkalineUnlock.pl uses Alkaline::Server to submit a certificate to a running server.

8.14.2. Usage

```
perl AlkalineUnlock.pl [server:port] [user] [password] [un-
lock key]
```

8.14.3. Notes

This script is fully supported in Alkaline 1.7.1213.0.

Chapter 9. Alkaline Licensing and Terms of Use

9.1. Introduction

This section details the Alkaline server software licensing.

9.2. Server License Agreement

By downloading this software, you agree to the Alkaline Server License Agreement and terms of use described in this section.

9.3. Terms of Use

9.3.1. General

As part of our service, Vestris Inc. agrees to provide freely downloadable evaluation versions of this software. Vestris Inc. agrees to provide this service free of charge. Upon notice published by Vestris Inc., Vestris Inc. may amend or modify these Terms of Service at any time. You agree to use the service in accordance with the Terms of Service.

9.3.2. Copyright Notices

The product is owned by Vestris Inc. All rights in the product including License Agreement Number, copyrights, licensing rights, patents, trademarks, trade secrets, design rights, engineering rights, moral rights, and any other intellectual property rights belong to Vestris. These rights are not transferred as part of this agreement.

No part of the product including the License Agreement may be reproduced, published, transmitted electronically, mechanically or otherwise, transcribed, stored in a retrieval system or translated into any language in any form, by any means, for any purpose other than the purchaser's personal use, without the express written permission of Vestris.

9.3.3. End User License Agreement

This license agreement is a legal agreement between you the *end user* (either as an individual or an entity) and Vestris Inc. (a BVI company) herein after referred to as *Vestris*. The software and any of its supporting documentation and associated hardware are herein after referred to as the *product*.

By installing or using the product, you indicate your complete and unconditional acceptance of these terms and conditions. If you do not agree to the terms and conditions of this agreement promptly remove the product form your computer and destroy associated documentation.

Should there be any conflict between the terms and conditions of this agreement and the terms and conditions of any other agreement between you and Vestris or their servants or agents in relation to the product the terms and conditions of this agreement shall apply.

If any provision of this agreement is found to be unlawful or void then that provision shall be severed from this agreement and will not affect the validity of the remaining provisions.

Information contained in the product is subject to change without notice and does not represent a commitment or contracted obligation on the part of Vestris.

9.3.4. Evaluation License Agreement

This Evaluation License Agreement grants you a non-exclusive License to use the product for an evaluation period of fifteen (15) days from the date you install the product. On the sixteenth (16) day after you installed the product you must either register the product by means of purchasing a commercial License Agreement from Vestris or destroy all copies of the product in your possession, and any related documentation.

You may make a single copy of the *software only* and no other part of the product in a machine readable form for backup use only.

This License Agreement (including the product) is not transferable. You may not lend, loan, lease, rent, sell or distribute the product (or copies) in any form.

9.3.5. Refunds Policy

Each purchase is final. As Vestris Inc. offers users partial or full evaluation versions of the software, our policy is not to refund. Exceptions to this policy include illegally purchased products, such as using a stolen credit card.

If you are unsure whether you are entitled to a refund, please email admin@vestris.com. For a refund that is disputed with Vestris Inc., please send all the purchase details to Kagi at refund@kagi.com and Kagi will handle your request.

For stolen credit cards and other such claims, please file chargeback with your financial institution.

9.3.6. Limitations of Use

You are expressly forbidden from making alterations or modifications to, merge, adapt, de-compile, disassemble, reverse engineer, or attempt to discover the source code without the expressed written permission from Vestris. You are expressly forbidden from using any part of the product in life saving or life critical applications without the expressed written permission from Vestris.

9.3.7. Limitations of Liability

Under no circumstances, including, but not limited to, negligence, shall Vestris Inc., its subsidiary and parent companies or affiliates be liable for any direct, indirect, incidental, special or consequential damages that result from the use of, or the inability to use, Vestris Inc. services. You specifically acknowledge and agree that Vestris Inc. is not liable for any defamatory, offensive or illegal conduct of any user of the free forums created using the Vestris Inc. service. If you are dissatisfied with any remotely hosted forum material, or with any of Vestris Inc. terms and conditions, your sole and exclusive remedy is to discontinue using Vestris Inc. services. In addition, You release Vestris Inc. and its affiliates from any damages that you incur, and agree not to assert any claims against them, arising from your use of it's products or services.

9.3.8. Privacy

Unless the user informs us otherwise, Vestris Inc. reserves the right to use and to disclose to third party vendors user information (e.g. user name and e-mail address) for purposes such as providing users with information about products and services.

9.3.9. Arbitration

Vestris Inc. is a registered British Virgin Islands company at Trident Chambers, P.O. Box 146, Wickhams Cay, Road Town, Tortola, British Virgin Islands. If arbitration should occur, the court can only be governed by laws of London, Great Britain.

9.3.10. Disclaimer

Vestris Inc. has provided links and pointers to Internet sites maintained by third parties. Neither Vestris Inc., its parent or subsidiary companies nor their affiliates operate or control in any respect any information, products or services on these third-party sites. The materials in this site and the third-party sites are provided "as is" and without warranties of any kind either express or implied. To the fullest extent permissible pursuant to applicable law, disclaims all warranties, express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. Vestris Inc. does not warrant that the functions contained in the materials will be uninterrupted or error-free, that defects will be corrected, or that this site, including bulletin boards, or the server that makes it available, are free of viruses or other harmful components. Vestris Inc. does not warrant or make any representations regarding the use or the results of the use of the materials in this site or in third-party sites in terms of their correctness, accuracy, timeliness, reliability or otherwise. You (and not Vestris Inc.) assume the entire cost of all necessary maintenance, repair or correction.

9.4. Definition of Non-Commercial Companies

This policy, mainly extracted from law of Great Britain and Canada, addresses the conditions that an entity must meet to qualify as a non-profit organization for purposes of the non-commercial licensing. When determining whether an entity is a *non-profit organization*, the entity must meet all of the following conditions:

- It was organized solely for non-profit purposes.
- It is in fact operated solely for non-profit purposes.
- It does not distribute or otherwise make available for the personal benefit of any member any of its income.

In case a group, company or individual does not meet all of the described conditions, it is automatically considered as a profit company, group or individual.

9.4.1. It was organized solely for non-profit purposes.

To be a non-profit organization, an entity must be organized solely for a purpose other than profit. To establish the purpose for which an entity was organized, Vestris Inc. will normally look to the instruments by which it was created. These instruments may include the organizational web site, letters patent, articles of incorporation, orders-in-council, legislation, memoranda of agreement, by-laws, articles and so on.

To qualify as a non-profit organization, ideally the governing documents should contain a statement that the entity is organized solely for non-profit purposes. However, in some situations they may not. In those situations Vestris Inc. will examine the purposes for which the entity was organized to determine whether the entity was organized solely for non-profit purposes. Entities which are organized solely for a non-commercial public purpose will be considered to be organized for non-profit purposes. This public purpose may include: social welfare, civic improvement, pleasure, recreation, relief of poverty, advancement of education or religion or other similar purpose. In general terms, social welfare means that which provides assistance for disadvantaged groups or for the common good and general welfare of the people of the community. Civic improvement includes the enhancement in value or quality of community or civic life. An example would be an association that works for the advancement of a community by encouraging the establishment of new industries, parks, museums, etc. Pleasure or recreation means that which provides a state of gratification or a means of refreshment or diversion. Examples include social clubs, golf clubs, curling clubs, badminton clubs and so on that are organized and operated to provide recreational facilities for the enjoyment of members and their families.

An entity may be considered to be organized solely for non-profit purposes if its aims and activities are directed toward the general improvement of conditions within one or more areas of business. An example of this would be where an entity was organized to advance the educational standards within a particular industry or profession, to publicize, improve and promote the entity’s objectives in a general way and to encourage the exchange of relevant technical information. If the activities of such an entity were consistent with these aims, then it would qualify as a non-profit organization provided that all other conditions with respect to non-profit organizations were complied with. However, the entity will probably not qualify as a non-profit organization if it is primarily involved, for example, in an activity that is directly connected with the sales of members’ goods or services and for such services receives a fee or commission computed in relation to sales promoted. Such an entity is normally considered to be an extension of the members’ sales organizations and will be considered to be carrying on a normal commercial operation. If the fees and commissions charged are well beyond the needs of the entity and these earnings are accumulated and invested as described below by the entity, this would be another reason why the entity would not qualify as a non-profit organization.

In some cases an entity may be organized under legislation for corporations without share capital. Corporations without share capital are generally regarded as non-profit corporations. Such corporations may be incorporated federally, or provincially. Legislation for corporations without share capital usually provides that the corporations are to be carried on without the purpose of gain for their members and any profit to such a corporation is to be used in promoting its objects. Vestris Inc. may use the information that an entity is organized under the applicable provisions of such legislation as evidence to establish that it was organized for non-profit purposes.

On the other hand, an entity may be organized under legislation for corporations with share capital. If a corporation is organized under such legislation, without any statement in the governing documents that it is organized for non-profit purposes, this may be conclusive evidence that it was organized for profit purposes.

9.4.2. Operated solely for non-profit purposes.

To be a non-profit organization, the entity must be one that is operated solely for a purpose other than profit. The determination of whether an entity is operated solely for non-profit purposes must be based on the facts of each case. Such a determination cannot be made in advance. Past activities will be reviewed. The length of time for pertinent past activities will depend on the particular situation.

Vestris Inc. is of the view that an entity is not operated solely for non-profit purposes when its principal activity is the carrying on of a commercial activity. Some characteristics of an activity that might be indicative that it is not operated in a non-profit manner are:

- It is a trade or business that is operated in a normal commercial manner.
- Its goods or services are not restricted to members and their guests.
- It is operated on a profit basis rather than a cost-recovery basis.

- It is operated in competition with taxable entities carrying on the same trade or business.

An entity may carry on an income-generating activity and still qualify as a non-profit corporation. To qualify, the income-generating activity must be carried on, and the resulting income must be used by the entity to achieve its declared non-profit objectives.

In certain cases, an entity may earn income in excess of its expenditures and still qualify as a non-profit organization. The excess may result from the activity for which it was organized or from some other activity. However, if a material part of the excess is accumulated each year and the balance of accumulated excess at any time is greater than the entity's reasonable needs to carry on its non-profit activities, Vestris Inc. will consider profit to be one of the purposes for which the entity is operated. This will be particularly so where assets representing the accumulated excess are used for purposes unrelated to its objects such as:

- Long-term investments to produce property income.
- Enlarging or expanding facilities used for normal commercial operations.
- Loans to members.

This may also be the case where the accumulated excess is invested in a term deposit or guaranteed investment certificate that is regularly renewed from year to year, whether or not the principal is adjusted from time to time.

The amount of accumulated excess considered reasonable in relation to the needs of an entity to carry on its non-profit activities is dependent on such things as the amount and pattern of receipts from various sources such as membership fees, training course fees, exam fees and so on. It is conceivable that there would be situations where an accumulation equal to one year's reasonably anticipated expenditures on its non-profit activities may not be considered excessive while in another situation an accumulation equal to two months' reasonably anticipated expenditures would be considered more than adequate. For example, a year-end accumulation equal to the following year's anticipated expenditures would probably be considered reasonable where an entity carries out its *annual fund drive* in the last month of its fiscal period in anticipation of its non-profit activities planned for the following year. However, where another entity raises its funds on a regular basis throughout the year, it may be difficult to justify a year-end accumulation in excess of an amount equal to its expenditures for one or two months. It is noted that where the present balance of accumulated excess is excessive or an annual excess is regularly accumulated it may indicate that the entity's aims are two-fold: to earn profits and to carry out its non-profit purposes. In such a case, the *operated solely* requirement would not be met.

As discussed above, accumulating surplus funds in excess of its current needs may affect the entity's status as a non-profit organization. However, in certain cases, when an entity requires a time period in excess of the current and prior year to accumulate the funds needed to acquire a capital property that will be used to achieve its declared non-profit activities, the entity may still qualify as a non-profit organization. For example, this could be the case if an entity annually sets aside funds for a special project such as the construction of a new building to replace an existing building when it deteriorates or no longer meets the entity's needs. In such cases, any funds accumulated for such a project should be clearly identified and all transactions concerning the project should be clearly set out in the entity's accounting records. Provided the funds accumulated are used for that project, an entity's non-profit status should not be affected.

9.4.3. No personal benefit for any member.

To be a non-profit organization, no part of the income of an entity, whether current or accumulated, may be payable to, or otherwise made available for the personal benefit of, any member of that entity. An entity may fail to comply with this requirement in a variety of ways. For example, an entity would not qualify as a non-profit organization if:

- It distributed income during the year, either directly or indirectly, to or for the personal benefit of any member.
- It has the power at any time to declare and pay dividends out of income or patronage dividends out of surplus.

- It, in the case of a winding-up, dissolution or amalgamation, has the power to distribute income to a member.

The presence of any of the circumstances described above would be conclusive evidence, subject to the comments below on what constitutes a personal benefit, that income was payable to, or otherwise made available for the personal benefit of, a member and that the entity did not qualify as a non-profit organization.

As indicated above, an entity will not qualify as a non-profit organization if it distributes income during the year to, or for the personal benefit of, any of its members. This may occur when the facilities of an otherwise non-profit organization, such as a non-profit golf or ski club, are used by non-members (other than guests of a member) and the income resulting from the fees charged to non-members is used to:

- Subsidize the fees or assessments charged to the members for the use of the facilities so that such amounts are either well below cost or nil.
- To acquire and maintain facilities or other property which the members use for no charge or for a fee well below cost.

In these circumstances, income of the entity is considered to be payable to, or otherwise available for the personal benefit of, its members and the entity would not qualify as a non-profit organization. However, if only members and guests of a member can use the entity's facilities, an entity will generally qualify as a non-profit organization provided the income from the guest fees or fund-raising activities is used to achieve the entity's declared non-profit objectives.

It is the Vestris Inc. view that certain types of payments made directly to members, or indirectly for their benefit, will not, in and by themselves, disqualify an entity from being a non-profit organization. Such payments include salaries, wages, fees or honorariums for services rendered to the entity, provided that amounts paid are reasonable and no more than those paid in arm's length situations for similar services. It also applies to payments made to employees or members of the entity to assist them in covering their expenses to attend various conventions and meetings as delegates on behalf of the entity, provided attendance at such conventions and meetings is to further the aims and objectives of the entity. In addition, Vestris Inc. considers the campaign expenditures of a political party, but not the payments to a candidate other than reimbursement of reasonable expenses, which will often result in an indirect benefit for a candidate, are not the type of personal benefit contemplated that would cause the party to be denied non-profit status under that provision.

9.5. Software and Service Pricing

A single server is a physical machine and is not necessarily what you index (remote servers, virtual servers).

- An individual or a non profit organization can download, install and use Alkaline for free. Alkaline server certificates are delivered on demand. Definition of a non-commercial company is available in Section 9.4.

  For example, *FooBar*, a non-commercial organization runs two Alkaline servers available at http://www.foo.org and http://www.bar.org at no cost.

- Commercial companies and sites must buy Alkaline. The price is of 350.- US$ for the first server license and 150.- US$ for each additional one. Each separate server requires a license.

  For example, *FooBar*, a commercial company, runs an Alkaline server for the external http://www.foo.com site and another two servers for the internal http://foo and http://bar sites at a total cost of $350.00 + 2 * $150.00 = $650.00.

- Commercial companies reselling and/or offering commercial clients the search engine capability should purchase a full Alkaline license for the first server, an additional server license for each other server, plus an additional server license for each separate client.

  For example, *Foo*, a web hosting company, runs two Alkaline servers for their three customers, *Bar*, *Tender* and *Loin* at a cost of $350.00 + $150.00 = $500.00 for the two servers hosted by *Foo*, plus 3 * $150.00 = $450.00 for their three customers for a total cost of $950.00.

- Commercial companies offering their non-commercial clients the search engine capability should purchase at least a full server license.

  For example, *Foo*, a web hosting company, runs two Alkaline servers for their three non-commercial customers, *Bar*, *Tender* and *Loin* at a cost of $350.00 + $150.00 = $500.00 for the two servers hosted by *Foo* and no additional cost for the three customers.

9.6. Reseller, Source-Code and OEM Licensing

- If you wish to discuss confidential information with our company, we offer you the possibility to sign a Non-Circumvention, Non-Disclosure and Confidentiality Agreement. We will send you the agreement signed by an authorized person in return.

- If you wish to purchase 10+ Alkaline licenses, please fill and sign the Reseller License Agreement. This agreement offers discounts of up to 70% for bulk licensing.

- If you wish to purchase the source code of Alkaline, please read the Source Code License Agreement. If you agree on general terms, please send us a detailed description of the purposes for acquiring the source code and we will discuss a pricing schedule.

- Please email admin@vestris.com for all other questions and licensing issues.

9.7. Paying for a License

9.7.1. Online Purchase

Online purchase with a credit card is the preferred method of payment.

- Payment is secure over SSL and fast.
- A trusted organization (Kagi), not Vestris Inc., will debit your account.
- Kagi has been serving customers, including Vestris Inc. clients, for many years now. Vestris Inc. uses Kagi services since 1994.
- Vestris Inc. will never see and has no means of obtaining your credit card information.
- You will receive registration information in 24-48 hours depending on processing time from Vestris Inc. and an immediate confirmation from Kagi.
- Your order will be confirmed only after your account is debited in order to avoid extensive internet fraud.
- There is no paperwork, but you can request a paper receipt for your transaction.

To pay for a license, use any of the links below:

- English
- German
- French
- Italian
- Dutch
- Swedish
- Japanese

For detailed information about how to fill in Kagi forms for your online purchase, please refer to http://www.kagi.com/FAQs/StepKoopPay.html.

9.7.2. Email Purchase

When paying via email you must use a credit card. You can download the Register program from http://www.vestris.com/files/Register.zip, that will guide you through the registration process. Detailed instructions for email payments can be found at http://www.kagi.com/FAQs/StepEmailPay.html.

9.7.3. Fax Purchase

When paying via email you must use a credit card. You can download the Register program from http://www.vestris.com/files/Register.zip, that will guide you through the registration process. Detailed instructions for email payments can be found at http://www.kagi.com/FAQs/StepFaxPay.html.

9.7.4. Postal Purchase

All forms of payment are accepted via postal mail. Detailed instructions can be found at http://www.kagi.com/FAQs/StepPostalPay.html.

9.7.5. Purchase Orders

Vestris Inc. does not accept purchase orders starting January 1st 2000, unless they exceed US$1000.00. If you have such an order, please send full billing and contact information to admin@vestris.com.

You have several other options:

- You can use an organizational credit card to purchase the products.

- You can use your personal credit card, pay for a postcard receipt and then when you receive the receipt, submit that proof of purchase to your organization for a refund.
- You can create an invoice to submit to your purchasing department by using the Register program. Please read http://www.kagi.com/FAQs/POs.html for details.

9.7.6. Questions and Issues

For payment processing issues with Kagi, please email admin@kagi.com. For other payment and licensing questions or issues, please contact admin@vestris.com.

9.8. Software Certification and Registration

9.8.1. Certification Mechanism

This software uses a certification mechanism to unlock it's binaries. If you install the software on your server, you should read this section.

First, download and install Alkaline. If you are a commercial company, pay for a license.

You should be able to perform search operations and navigate to the admin section. If you are unable to navigate to the admin section, or if you do not know how to do that, please refer to the Alkaline FAQ.

From the admin links, choose the Certification link. A form is shown. If the form is not shown and a message, such as "a valid certificate is installed for this server", then you have already performed the certification operations.

Fill in and post the form. You must provide valid information, especially your email address. The form contains the certificate request field for your server. If you are on a private network and cannot post the form to the internet, copy the certificate and fill a similar form at http://www.vestris.com/certify/certify.html. You should immediately receive a success message and a confirmation email within a few minutes.

You may receive an email stating that the request was invalid or that the certification has been refused. Read that email carefully and try to correct the problem. If you are still having certification problems, please email admin@vestris.com.

Upon successful completion of your request, you will receive the certificate in an email. Copy-paste the certificate to the same certification form under the Unlock Key section. Submit the certificate. This should succeed and the nags will be removed. Check the other frequently asked questions about certification if the server has returned a particular error.

I. The asearch.cnf Configuration Reference

The asearch.cnf Configuration Reference

UrlList

Name

`UrlList` — specify the root URLs to index

Synopsis

UrlList = *`url1`* `[`*`,url2`*`][`*`,url3 ...`*`]`

Description

This parameter defines the root URLs to retrieve. Alkaline will start the indexing process with each of these URLs in a sequential order. An url can be of both *http://server:port/path/file?arg* and *file:///path/name* formats. The file:// format was added in version 1.7.

Unless the **Robots** directive is set to No, Alkaline will attempt to retrieve the *http://server-name/robots.txt* file first first. Contents of the robots.txt file are stored for the entire session. No robots.txt file is retrieved for local file:// urls.

After retrieving the first page defined by the UrlList directive, Alkaline will extract all links from it and schedule them for indexing in the order they were found. A thread from a thread pool will pickup each of these urls, match them to the current asearch.cnf rules and if required, restart the indexing process for the particular url. Local directory listings retrieved following a file:// url will be treated as html content with links to each document and subdirectory.

After emptying the queue of urls, retrieved using the initial entry of the UrlList directive, Alkaline will continue with the next url in the list, if any. Otherwise, if more asearch.cnf files are specified, the spider will pursue with the next asearch.cnf configuration. Once the entire list of configurations is processed, it will mark an inactivity period of SleepRoundtrip and restart with the first configuration file.

Example

```
UrlList=http://www.foo.com,http://foo.bar.ch
```

UrlExclude

Name

`UrlExclude` — exclude urls from indexing

Synopsis

UrlExclude = *`comma-separated list of urls`*

Description

Any url found by the spider starting with one of the urls defined by this directive will not be scheduled for indexing. Alkaline will do a case-sensitive match.

With the global RegExp option enabled or a RegExp prefix, the parameter must be a list of regular expressions. This option has the *RegExp UrlExclude* extension since version 1.6.

Note that this directive has no effect on already indexed urls. It will be used only when a new url is found by the spider.

Example

```
UrlExclude=http://www.foo.com/private/,http://foo.bar.com/stats.html
```

UrlListFile

Name

`UrlListFile` — add a list of urls from a file to UrlList

Synopsis

UrlListFile = *`file1`* [,*`file2`*][,*`file3 ...`*]

Description

Whenever you are indexing a large set of individual urls or have more than a few items to add to the UrlList directive, it is more convenient to use UrlListFile rather than UrlList. The argument must be a list of files with absolute paths or relative to the directory that Alkaline is run from. Each file contains a list of urls, each url must be on a single line.

Alkaline will reload the lists of urls at each full indexing roundtrip. You can create your own tools that populate a file pointed by this directive while Alkaline is indexing; which is simpler than writing a full asearch.cnf file. Each url is treated as if it was written part of the list defined by the UrlList directive.

To index a large set of individual urls, you can use the UrlListFile directive along with a Depth setting of 0. Thus, you can create a small tool that allows users to add their sites to the Alkaline index from a web page - it will simply append the requested url to a file pointed by UrlListFile.

Example

```
UrlListFile=/usr/local/alkaline/urls.list
```

Where the urls.list file contains:

```
http://www.foo.com/
http://foo.bar.com/stats/
```

UrlExcludeFile

Name

`UrlExcludeFile` — add an url from a file to UrlExclude

Synopsis

UrlExcludeFile = *`file1`* [,*`file2`*][,*`file3 ...`*]

Description

Whenever you want to exclude a large set of urls, it is more convenient to use UrlExcludeFile rather than UrlExclude. The argument must be a list of files with absolute paths or relative to the directory that Alkaline is run from. Each file contains a list of urls, each url must be on a single line.

Alkaline will reload the lists of urls at each full indexing roundtrip. Each url is treated as if it was written part of the list of the UrlExclude directive.

Example

```
UrlExcludeFile=/usr/local/alkaline/exclude.list
```

Remote

Name

`Remote` — index or exclude urls from a remote site

Synopsis

Remote = *`Y / N`*

Description

When N, the spider will not schedule urls, that do not have the exact same server name as the root url defined by UrlList, for indexing.

Both server names www.foo.com and foo.bar.com are considered remote from www.bar.com.

Default

```
Remote=N
```

UrlInclude

Name

`UrlInclude` — define a global url include scope

Synopsis

UrlInclude = *url1* [,*url2*][,*url3 ...*]

Description

Instruct Alkaline to spider a particular scope of urls. This directive potentially allows to spider, for example all .bar.com or all .com domains.

With the global RegExp option enabled or a RegExp prefix, the parameter must be a list of regular expressions. This option has the *RegExp UrlInclude* extension since version 1.6.

Example

To index all .bar.com domains, such as foo.bar.com and www.bar.com:

```
UrlList=http://www.bar.com/
UrlInclude=.bar.com
Remote=Y
```

Since Alkaline is a spider and follows links, this directive alone is not enough to spider all the .bar.com domains. A physical link from a page in www.bar.com must exist in any case to some other page in foo.bar.com.

UrlIncludeFile

Name

`UrlIncludeFile` — define a global url include scope list

Synopsis

UrlIncludeFile = `file1 [,file2][,file3 ...]`

Description

Whenever you want to define a large set of scope urls, it is more convenient to use UrlIncludeFile rather than UrlInclude. The argument must be a list of files with absolute paths or relative to the directory that Alkaline is run from. Each file contains a list of urls, each url must be on a single line.

Alkaline will reload the lists of urls at each full indexing roundtrip. Each url is treated as if it was written part of the list of the UrlInclude directive.

Example

```
UrlIncludeFile=/usr/local/alkaline/inc.list
```

UrlIndex

Name

`UrlIndex` — follow links and index

Synopsis

UrlIndex = `url1 [,url2][,url3 ...]`

Description

Instruct Alkaline to index only a particular set of urls. When specified, all urls *not* matching the UrlIndex criteria, will be retrieved, parsed, but not indexed. This means their content and meta tags will not be used and matching pages will not appear in search results, but the links will still be followed. Unlike the UrlInclude directive, this option is processed after a page is retrieved.

With the global RegExp option enabled or a RegExp prefix, the parameter must be a list of regular expressions. This option has the *RegExp UrlIndex* extension since version 1.6.

This option was added in version 1.3 (09-Jul-2000).

Example

To index only http://www.foo.com/bar/ files, but follow all links:

```
UrlList=http://www.foo.com/
UrlIndex=http://www.foo.com/bar/
```

UrlIndexFile

Name

`UrlIndexFile` — follow links and index

Synopsis

UrlIndexFile = *`file1`* [,*`file2`*][,*`file3 ...`*]

Description

Whenever you want to define a large set of scope urls, it is more convenient to use UrlIndexFile rather than UrlIndex. The argument must be a list of files with absolute paths or relative to the directory that Alkaline is run from. Each file contains a list of urls, each url must be on a single line.

Alkaline will reload the lists of urls at each full indexing roundtrip. Each url is treated as if it was written part of the list of the UrlIndex directive.

Example

```
UrlIndexFile=/usr/local/alkaline/urlindex.list
```

UrlSkip

Name

`UrlSkip` — follow links, but do not index

Synopsis

UrlSkip = *`url1`* `[,`*`url2`*`][,`*`url3`* `...]`

Description

Instruct Alkaline not to index a particular set of urls. When specified, all urls matching the UrlSkip criteria, will still be spidered but not indexed. This means their content and meta tags will not be indexed, these pages will not appear in search results, but the links that they contain will still be followed. Unlike the UrlExclude directive, this option is processed after a page is retrieved.

With the global RegExp option enabled or a RegExp prefix, the parameter must be a list of regular expressions. This option has the *RegExp UrlSkip* extension since version 1.6.

This option was added in version 1.3 (09-Jul-2000).

Example

To avoid indexing http://www.foo.com/bar/ files, but follow all links from these pages:

```
UrlList=http://www.foo.com/
UrlSkip=http://www.foo.com/bar/
```

UrlSkipFile

Name

`UrlSkipFile` — follow links, but do not index

Synopsis

UrlSkipFile = *`file1`* [,*`file2`*][,*`file3 ...`*]

Description

Whenever you want to define a large set of scope urls, it is more convenient to use UrlSkipFile rather than UrlSkip. The argument must be a list of files with absolute paths or relative to the directory that Alkaline is run from. Each file contains a list of urls, each url must be on a single line.

Alkaline will reload the lists of urls at each full indexing roundtrip. Each url is treated as if it was written part of the list of the UrlSkip directive.

Example

```
UrlSkipFile=/usr/local/alkaline/urlskip.list
```

Depth

Name

`Depth` — define the relative url depth of indexing

Synopsis

Depth = *-1 / positive number*

Description

Defines the maximum depth of urls to follow. Alkaline will first index pages of Depth 1, 2, etc. A negative value of -1 instructs the spider to ignore this setting.

With Remote=Y and Depth=-1, Alkaline will attempt to index all encountered links (the entire web).

To index a large set of individual urls, you can use the UrlListFile directive along with a Depth setting of 0. This will have the same effect as the SkipLinks=Y directive.

Default

```
Depth=-1
```

SiteDepth

Name

`SiteDepth` — define the maximum paths depth of urls to follow

Synopsis

SiteDepth = *`-1 / 0 / positive number`*

Description

Index urls that are at maximum **SiteDepth** path segments deep, relative to the server name. For example http://www.foo.com/ is has a **SiteDepth** of 0, http://www.foo.com/bar/ has a **SiteDepth** of 1, etc.

To index urls at the root only, specify **SiteDepth=0**.

Default

```
SiteDepth=-1
```

RemoteDepth

Name

`RemoteDepth` — define the maximum remote depth of urls to follow

Synopsis

RemoteDepth = *`-1 / number`*

Description

Index urls that are at most **RemoteDepth** hops away. Unlike the **Depth** setting, **RemoteDepth** counts real distance from the first server and correctly handles loop paths.

Specifying **RemoteDepth=0** is equivalent to **Remote=N**, although **Remote=N** is caught much earlier in the processing stage.

This option was added in version 1.6.0824.0.

Default

```
RemoteDepth=-1
```

MaxFiles

Name

`MaxFiles` — the maximum number of files to index

Synopsis

MaxFiles = *`-1 / positive number`*

Description

Index at most **MaxFiles** urls. After this number is reached, Alkaline will write the index, clear all scheduled urls and quit indexing. If this setting is left at the default value of -1, Alkaline will stop indexing after the scheduled urls list is empty (all urls have been indexed under the rules of the asearch.cnf file).

Default

```
MaxFiles=-1
```

MaxLinks

Name

`MaxLinks` — the maximum number of links to follow

Synopsis

MaxLinks = *`-1 / positive number`*

Description

Index at most **MaxLinks** urls from the topmost url. A topmost url is typically an entry in UrlList.

After this number is reached, Alkaline will write the index, clear all scheduled urls and quit indexing the current topmost url. If this setting is left at the default value of -1, Alkaline will stop indexing after the scheduled urls list is empty (all urls have been indexed under the rules of the asearch.cnf file).

This option was added in version 1.3 (03-Aug-2000).

Default

```
MaxFiles=-1
```

Upper

Name

`Upper` — follow or ignore parent server paths

Synopsis

Upper = *`Y / N`*

Description

An **Upper** url is a partial leftmost match of the current url. For example http://www.foo.com/bar/ is an **Upper** url for http://www.foo.com/bar/drinks/.

Default

`Upper=Y`

Reindex

Name

`Reindex` — index in background

Synopsis

Reindex = *`Y / N`*

Description

Instructs Alkaline to index the current configuration during the background indexing process.

Notes

The *NoReindex* option with exact opposite effect was deprecated in version 1.6.0824.0 and replaced by *Reindex*. The deprecated option is still supported by the engine.

Default

```
Reindex=Y
```

Exts

Name

`Exts` — valid extensions for files to spider

Synopsis

Exts = `*ext1*` [,*ext2*][,*ext3* ...]

Description

Defines a list of valid extensions for files to retrieve and index. Extensions must not have the leading dot. Urls without an extensions, such as http://www.foo.com/bar/ will be retrieved anyway.

Default

```
Exts=html,htm,txt,shtml
```

ExtsAdd / AddExts

Name

`ExtsAdd / AddExts` — add extensions to the Exts directive

Synopsis

ExtsAdd = *`ext1`* [,*`ext2`*][,*`ext3 ...`*]

AddExts = *`ext1`* [,*`ext2`*][,*`ext3 ...`*]

Description

List of extensions to add to the Exts directive. For example, to spider .pdf files, you will need to adjust the extensions to contain .pdf.

Example

```
ExtsAdd=pdf,php,asp,exe
```

Robots

Name

`Robots` — respect server robots directives

Synopsis

Robots = *Y / N*

Description

Respect server robots and META robots directives.

> Ignoring server robots settings is very impolite. Besides, there might be other reasons why certain paths are excluded. They might lead to a large amount of dynamically generated pages with few differences or worse, to an infinite amount of generated dynamic pages that will leave Alkaline indexing forever a dataset of a constantly exploding size.

Notes

The *NoRobots* option with exact opposite effect was deprecated in version 1.6.0824.0 and replaced by *Robots*. The deprecated option is still supported by the engine.

Default

```
Robots=Y
```

SkipText

Name

`SkipText` — do not index plain text

Synopsis

SkipText = `Y / N`

Description

Skip all plain text except META tags when indexing a page.

Default

`SkipText=N`

SkipMeta

Name

`SkipMeta` — do not index meta tags

Synopsis

SkipMeta = `Y / N`

Description

Skip all meta tags when indexing a page.

Default

`SkipMeta=N`

SkipLinks

Name

`SkipLinks` — do not follow links

Synopsis

SkipLinks = *`Y / N`*

Description

Do not follow links when indexing a page. This option has the same effect as Depth=0 and is commonly used in conjunction with the UrlListFile option.

Default

`SkipLinks=N`

EmptyLinks

Name

`EmptyLinks` — process and queue invisible links

Synopsis

EmptyLinks = *`Y / N`*

Description

Process and queue empty links, such as <a href=url></a>. Such links are invisible and cannot be clicked on from a browser, but they are useful if you need to link a page for a search engine exclusively.

Disable this option for Lotus Notes Domino generated sites, which randomly makes empty links leading to nowhere.

If you want to add links for Alkaline specifically, use the <alkaline url=... > tag.

Notes

The *NoEmptyLinks* option with exact opposite effect was deprecated in version 1.6.0824.0 and replaced by *EmptyLinks*. The deprecated option is still supported by the engine.

Default

```
EmptyLinks=Y
```

Index.html

Name

`Index.html` — default for urls without a filename

Synopsis

Index.html = *`file name`*

Description

An url such as http://www.foo.com/ does not have a file name. The server might resolve it to http://www.foo.com/index.html, index.asp or any other name. To avoid duplicates, Alkaline uses the MD5 hash algorithm for page contents, but it will not be effective if the default page is dynamic. In such cases you may want to force the default name in order to avoid duplicate results.

Example

```
Index.html=default.asp
```

HeaderLength

Name

`HeaderLength` — maximum $header length

Synopsis

HeaderLength = *`positive or zero number`*

Description

Defines maximum amount of characters to store as the free text shown with the $header directive in search results, unless a meta description tag is found on the page. In the later case, Alkaline will ignore this option.

Default

```
HeaderLength=256
```

FreeCharset

Name

`FreeCharset` — disable character decoding

Synopsis

FreeCharset = *`Y / N`*

Description

Do not decode accentuated characters: if you search pages other than English, French or German (for example Russian or Georgian) you must set this option. It will avoid character transformation of &acute;-like symbols into non-accentuated ones during both search and index phases. If you use this option, also set <!–SET FREECHARSET–1–> in the search.html templates.

You must set this option in all asearch.cnf configurations (in general, the first configuration defines the behavior of searching and each configuration defines its behavior of indexing).

Default

```
FreeCharset=N
```

Redirect

Name

`Redirect` — define equivalent or redirected urls

Synopsis

Redirect *`source-url = target-url`*

Description

Define equivalent or redirected urls. Alkaline will consider two sites like http://www.foo.com and http://foo.com to have different root urls. This directive solves the problem and avoids Alkaline not to index the same files twice. You can also use this option to redirect an IP address to it's textual url equivalent.

Example

```
Redirect http://foo.com=http://www.foo.com
```

UrlReplace

Name

`UrlReplace` — indexed urls text replacements

Synopsis

UrlReplace *`source = target`*

Description

Define a text replacement for all indexed urls. The replacement is performed before any topmost url is retrieved or before any new url is scheduled.

With the global RegExp option enabled or a RegExp prefix, the source and target parameters must be regular expressions. This option has the *RegExp UrlReplace* extension since version 1.6.

Running Alkaline with the *-exv* parameter will output the details of replacements for this function, including regular expressions syntax errors. A warning will be issued when an url is truncated to an empty string as a result of the regexp replacement. To exclude urls from indexing, you should use the UrlExclude directive.

This option was added in version 1.3 (12-Jul-2000).

Example

```
UrlReplace foo=bar
```

```
RegExp UrlReplace (.*)(www)(.*)=\1ns\3
RegExp UrlReplace (.*)\$password\=[0-
9]*(.*)=\1$password=default\2
```

Replace

Name

`Replace` — absolute url string replacement

Synopsis

Replace *`source-url = target-url`*

Description

Define string replacement for URLs in the search results. This can be useful when indexing local intranet domains that are behind firewalls.

With the global RegExp option enabled or a RegExp prefix, the source and target parameters must be regular expressions. This option has the *RegExp Replace* extension since version 1.8.

Example

The fully qualified host name of a server is www.foo.com. This is behind a firewall, so it is not known to the DNS. What is being returned from the search engine is http://www.foo.com/index.html. To return http://195.141.15.96/index.html, the following would have to be added:

```
Replace www.foo.com=195.141.15.96
RegExp Replace (.*)/folder/(.*)=\1/folder/item.php?id\=\2
```

Cgi

Name

`Cgi` — schedule cgi urls for indexing

Synopsis

Cgi = *`Y / N`*

Description

A cgi url contains a valid query string after the ? character. Such urls will scheduled for indexing only when this option is set.

Default

```
Cgi=N
```

Nsf

Name

`Nsf` — Lotus Notes Domino support

Synopsis

Nsf = *`Y / N`*

Description

Replace urls to Lotus Notes Domino queries such as *.nsf*?OpenDocument* by *.nsf*?OpenDocument&ExpandView;. This option should be enabled for Lotus Notes Domino servers in order to avoid duplicate search results pointing to a different view of the same document.

Default

```
Nsf=N
```

Insens

Name

`Insens` — case-insensitive url parsing and MD5

Synopsis

Insens = *`Y / N`*

Description

Treat urls and documents as case-insensitive; this includes calculating the MD5 signature to avoid duplicates and case-insensitive comparisons of urls. This option should be enable for Lotus Notes Domino and might be useful for Windows NT servers.

Default

```
Insens=N
```

LowerCase

Name

`LowerCase` — convert to lowercase

Synopsis

LowerCase = *`Y / N`*

Description

Convert all individual word forms to lowercase before indexing them. This should reduce the index size, but makes case-sensitive search impossible.

This option was added in version 1.3 (11-Aug-2000).

Default

`LowerCase=N`

UpperCase

Name

`UpperCase` — convert to uppercase

Synopsis

UpperCase = *`Y / N`*

Description

Convert all individual word forms to uppercase before indexing them. This should reduce the index size, but makes case-sensitive search impossible.

This option was added in version 1.32.1028.0.

Default

`UpperCase=N`

ExcludeWords

Name

`ExcludeWords` — exclude a list of words from indexing

Synopsis

ExcludeWords *`[type]`* `=` *`file1`* `[,`*`file2`*`][,`*`file3 ...`*`]`

Description

Each file must contain a list of words to exclude. Each word should be on a separate line. With the global RegExp option enabled or with the RegExp prefix, each word must be a regular expression. The *type* must be either a mime type, a valid extension or can be omitted. This option has the *RegExp ExcludeWords* extension since version 1.6.

> If you use an ExcludeWords directive with RegExp=Y, the default English dictionary supplied in the distribution cannot be used. It contains a `.*` expression which will exclude all possible word combinations.

It is possible to include multiple ExcludeWords directives in configuration files, but each type must have at most one directive. Dictionaries are loaded when they are required only.

> An NdxScan tool that outputs most frequent words in an Alkaline database is available as part of the AlkalineTools distribution.

Example

```
ExcludeWords=/usr/local/alkaline/words.list
ExcludeWords adobe/pdf=/usr/local/alkaline/adobe.list
```

The words.list file contains:

```
vestris
Vestris
Alkaline
```

IndexWords

Name

`IndexWords` — include only a list of words from a dictionary

Synopsis

IndexWords *`[type] = file1`* `[`*`,file2`*`][`*`,file3 ...`*`]`

Description

Exact opposite behavior of the ExcludeWords directive. Only words found in a dictionary for the given type will be included to the index. With the global RegExp option enabled or with the RegExp prefix, each word must be a regular expression. This option has the *RegExp IndexWords* extension since version 1.6.

Example

```
IndexWords=/usr/local/alkaline/words.list
```

IncludePages

Name

`IncludePages` — include pages containing words

Synopsis

IncludePages `[type] = file1 [,file2][,file3 ...]`

Description

Each file must contain a list of words. Each word should be on a separate line. With the global RegExp option enabled or with the RegExp prefix, each word must be a regular expression. This option has the *RegExp IncludePages* extension since version 1.6. The *type* must be either a mime type, a valid extension or can be omitted.

It is possible to include multiple IncludePages directives in configuration files, but each type must have at most one directive. Dictionaries are loaded when they are required only.

This option should be used in conjunction with IncludePagesAll directive.

ExcludeWords is first used on all indexed data before the IncludePages or the ExcludePages directives are processed.

Example

```
IncludePages=/usr/local/alkaline/words.list
IncludePages adobe/pdf=/usr/local/alkaline/adobe.list
```

The words.list file contains:

```
vestris
Vestris
Alkaline
```

IncludePagesAll

Name

`IncludePagesAll` — define the behavior of IncludePages

Synopsis

IncludePagesAll = `*Y / N*`

Description

When set, a page will have to match all words in all applicable dictionaries defined by IncludePages in order to be indexed. Otherwise, any word in any of the applicable dictionaries defined by IncludePages is sufficient.

Default

```
IncludePagesAll=N
```

ExcludePages

Name

`ExcludePages` — stop words

Synopsis

ExcludePages *[type] = file1* [,*file2*][,*file3 ...*]

Description

Each file must contain a list of words. Each word should be on a separate line. With the global RegExp option enabled or with the RegExp prefix, each word must be a regular expression.This option has the *RegExp ExcludePages* extension since version 1.6. The *type* must be either a mime type, a valid extension or can be omitted.

It is possible to include multiple ExcludePages directives in configuration files, but each type must have at most one directive. Dictionaries are loaded when they are required only.

This option should be used in conjunction with ExcludePagesAll directive.

ExcludeWords is first used on all indexed data before the IncludePages or the ExcludePages directives are processed.

Example

```
ExcludePages=/usr/local/alkaline/words.list
ExcludePages adobe/pdf=/usr/local/alkaline/adobe.list
```

The words.list file contains:

```
vestris
Vestris
Alkaline
```

ExcludePagesAll

Name

`ExcludePagesAll` — define the behavior of ExcludePages

Synopsis

ExcludePagesAll = *`Y / N`*

Description

When set, a page will have to match all words in all applicable dictionaries defined by ExcludePages in order not to be indexed. Otherwise, any word in any of the applicable dictionaries defined by IncludePages is sufficient.

Default

```
ExcludePagesAll=N
```

Expire

Name

`Expire` — treat all documents as out of date

Synopsis

Expire = *`Y / N`*

Description

Disable the lookup and the setting of the if-Modified-Since option when reindexing a site. Enable this option when, for example, indexing exclusively dynamic pages.

This option can be forced from the command line with *–expire*.

Default

```
Expire=N
```

NewOnly

Name

`NewOnly` — index only new documents

Synopsis

NewOnly = `Y / N`

Description

Index only documents that are absent from the index. Any document that has been previously indexed will not be checked for changes. This allows to restart indexing from the abandoned point and to index only newly added documents to the files pointed by the UrlListFile directive.

Enabling this option is not suitable for background indexing or reindexing of a previously built database from command line. It is intended for sites maintaining full lists of urls collected from users or generated dynamically.

📌 This option can be forced from the command line with *–newonly*.

Default

```
NewOnly=N
```

SkipParseLinks

Name

`SkipParseLinks` — index links in skip sections

Synopsis

SkipParseLinks = *`Y / N`*

Description

Index urls from a href, base href, frame, etc. sections even inside an <alkaline skip> block.

Default

```
SkipParseLinks=N
```

MetaDescription

Name

`MetaDescription` — store meta descriptions

Synopsis

MetaDescription = *`Y / N`*

Description

Store the meta description tag contents rather than a portion of plain text, shown as the document header in the search results,

Notes

The *NoMetaDescriotion* option with exact opposite effect was deprecated in version 1.6.0824.0 and replaced by *MetaDescription*. The deprecated option is still supported by the engine.

Default

```
MetaDescription=Y
```

TextDescription

Name

`TextDescription` — store plain text descriptions

Synopsis

TextDescription = *`Y / N`*

Description

Store descriptions from plain text, shown as the document header in the search results. If this option is set to N, a meta description is available and MetaDescription=Y, only meta descriptions will be shown.

Notes

The *NoTextDescription* option with exact opposite effect was deprecated in version

1.6.0824.0 and replaced by *TextDescription*. The deprecated option is still supported by the engine.

Default

```
TextDescription=Y
```

Md5

Name

`Md5` — enable the MD5 document matching

Synopsis

Md5 = `Y / N`

Description

MD5 Message-Digest Algorithm is described in RFC 1321 available at http://www.faqs.org/rfcs/rfc1321.html. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit *fingerprint* or *message digest* of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest.

The MD5 digest mechanism within Alkaline enables removal of duplicate pages pointed by two different urls. This is typical for documents that can be both accessed with *http://www.server.com/* and *http://www.server.com/index.html*. Alkaline believes that this is a different document and catches the fact that they are identical by comparing the MD5 digests.

Setting this option to N when indexing single urls from different sites or fully dynamic pages may speed-up the indexing process.

Notes

The *NoMd5* option with exact opposite effect was deprecated in version 1.6.0824.0 and replaced by *Md5*. The deprecated option is still supported by the engine.

Default

`Md5=Y`

Cookie

Name

`Cookie` — set a global cookie to be set to servers

Synopsis

Cookie *`name = value`* `[`*`; domain = domurl`*`][`*`; path = uri-path`*`][`*`; expires = utc date`*`]`

Description

Set a random cookie that will be set as part of the *Cookies:* header to matching servers.

Each *Cookie* entry defines a single cookie that can contain optional *domain*, *path* and *expires* attributes.

Unlike with regular browsers, Alkaline does not enforce any limitations on the amount of cookies or the length of cookie values. This option also allows to specify a cookie sent to all servers (a cookie without a domain or a path attribute).

For more information about cookies, please refer to the original Netscape specification at http://www.netscape.com/newsref/std/cookie_spec.html.

Example

```
# (write on one line)
Cookie PARTNUMBER=00000_55555;
 domain=mydomain.com;
 path=/;
 expires=Wednesday, 09-Nov-2000 23:12:40 GMT
Cookie SECTION=third%20floor
```

Cookies

Name

`Cookies` — retrieve, store and send cookies

Synopsis

Cookies = *`Y / N`*

Description

Enables receiving, storing and sending cookies.

Specifying the Cookie option along with *Cookies=N* will have no effect as no cookies will be sent or received.

For more information about cookies, please refer to the original Netscape specification at http://www.netscape.com/newsref/std/cookie_spec.html.

Notes

The *NoCookies* option with exact opposite effect was deprecated in version 1.6.0824.0 and replaced by *Cookies*. The deprecated option is still supported by the engine.

Default

```
Cookies=Y
```

RequestHeader

Name

`RequestHeader` — define an http request header

Synopsis

RequestHeader *`name`* `=` *`value`*

Description

Set a random request header that is sent to http servers at every request. You can specify multiple request headers and redefine the User-Agent header that Alkaline sets to *User-Agent: AlkalineBOT/Major.Minor (Major.Minor.Build.Dot Alpha/Beta/RTM)*.

Notes

This feature was added in build 1.4.0215.0.

Example

```
RequestHeader User-Agent=AlkalineBOT/1.9 (vestris.com)
RequestHeader Content-Language=mi,en
```

Filter

Name

`Filter` — define a document type filter

Synopsis

Filter = *`[type] = command line expression`*

Description

The *type* must be either a mime type, a valid extension or can be omitted. The command line expression defines the sequence of operations to invoke for the particular type.

Please refer to the filters documentation section for more details.

Example

```
Filter adobe/pdf=/usr/bin/pdf2text -q $1 $2
```

Object

Name

`Object` — define an embedded objects filter

Synopsis

Object *`clsid:guid = command line expression`*

Description

The *guid* must be a valid class-id guid. The command line expression defines the sequence of operations to invoke for the particular object type.

Please refer to the embedded objects documentation section for more details.

Example

```
# Shockwave Flash
# (write on one line)
Object clsid:D27CDB6E-AE6D-11cf-96B8-444553540000=
 /usr/local/bin/swf-filter $sourcefile
 -menu="$menu" > $targetfile
```

ObjectDocument

Name

`ObjectDocument` — define the default param for Object

Synopsis

ObjectDocument *`clsid:guid = param value`*

Description

The *guid* must be a valid class-id guid. The param value must be a valid one for the particular object type.

Please refer to the embedded objects documentation section for more details.

Example

```
# Shockwave Flash
ObjectDocument clsid:D27CDB6E-AE6D-11cf-96B8-444553540000=movie
```

MaxSize

Name

`MaxSize` — maximum document size

Synopsis

MaxSize = *`-1 / positive number`* [*`K,KB,M,MB`*]

Description

Defines the largest document to retrieve in bytes unless otherwise specified. A negative value removes any limit.

Default

```
MaxSize=1MB
```

Auth

Name

`Auth` — supply credentials for authentication

Synopsis

Auth *[domain\]username = password*

Description

When a server responds by *401/Unauthorized* and challenge Alkaline for BASIC authentication with a *WWW-Authenticate: BASIC realm=...* header, Alkaline will attempt to use the credentials defined by this option. The domain value is ignored by this protocol.

In Alkaline for Windows NT, when a server responds by *401/Unauthorized* and challenge Alkaline for Windows NT Challenge/Response NTLM authentication with a *WWW-Authenitcation: NTLM* header, Alkaline will attempt to negotiate a security context using credentials defined by this option. If the server rejects all credentials, negotiating with the current user context will also be attempted. Hence, if you are indexing an intranet as an authorized user, you do not need to specify any Auth options.

It is possible to specify as many Auth directives as required.

To set up such a protected zone Apache uses for example .htaccess and .htpasswd files. On Windows and IIS, the virtual directory security settings can be BASIC or Windows Authentication. Your browser usually pops a login window when you navigate to such a site.

NTLM authentication is not supported on UNIX. Digest, Negotiate or Kerberos authentications are not supported in all versions.

Support for NTLM authentication was added in version 1.5.

Example

```
Auth Foo=foopassword
Auth Bar=barpassword
# Windows NT authentication
```

```
Auth ntdom\asearch=password
```

Proxy

Name

`Proxy` — define an HTTP proxy to use for document retrieval

Synopsis

Proxy = *`server:port`*

Description

Use an HTTP/1.0 compliant proxy server.

Example

```
Proxy=proxy.foo.com:80
```

Timeout

Name

`Timeout` — network timeout period

Synopsis

Timeout = *`positive number`*

Description

Number of seconds to timeout on document retrieval operations while indexing. This includes all attempts to connect, read or write to the network.

Default

`Timeout=30`

DnsTimeout

Name

`DnsTimeout` — dns lookup timeout period

Synopsis

DnsTimeout = *`positive number`*

Description

Number of seconds to timeout on domain name resolution while indexing.

The Domain Name System (DNS) is described in RFC 1034 and RFC 1035, available at http://www.faqs.org/rfcs/rfc1034.html and http://www.faqs.org/rfcs/rfc1035.html.

The Domain Name System (DNS) is the method by which Internet addresses in mnemonic form such as ns.vestris.com are converted into the equivalent numeric IP address such as 123.220.4.1. To the user and application process this translation is a service provided either by the local host or from a remote host via the Internet. The DNS server (or resolver) may communicate with other Internet DNS servers if it cannot translate the address itself.

This option was added in version 1.6.0915.0.

Default

```
DnsTimeout=30
```

Retry

Name

`Retry` — retry count for a timed-out connection

Synopsis

Retry = *`0 / positive number`*

Description

Number of times to retry when retrieving a remote page and receiving a time-out or a network error. It often happens that an existing page cannot be retrieved because of network interruptions or latencies.

Default

```
Retry=3
```

SleepFile

Name

`SleepFile` — lazy mode delay between files

Synopsis

SleepFile = *0 / positive number*

Description

Delay in seconds for the indexing thread pool to sleep after a file has been indexed in lazy mode. Alkaline goes into a so-called lazy mode when responding to search requests.

When you specify this option with a -mi parameter bigger than 1 (which is the default), you might see no effect of this option because it defines the sleep interval of each indexing thread rather than across all indexing threads. Use -mi=1 to obtain the correct effect of sleep interval.

Default

```
SleepFile=1
```

SleepRoundtrip

Name

`SleepRoundtrip` — lazy mode delay between roundtrips

Synopsis

SleepRoundtrip = *`0 / positive number`*

Description

Delay in seconds for the indexing thread pool to sleep after a whole group of urls has been processed in lazy mode.

If you search multiple sites with multiple asearch.cnf configurations, you may want to make a longer delay after the last asearch.cnf is processed.

Default

```
SleepRoundtrip=60
```

LogFile

Name

`LogFile` — write a log file

Synopsis

LogFile = *file*

Description

Alkaline will append the date, time, remote ip, the search group and the search string to this file when a user performs a search operation. Multiple configurations can have the same LogFile.

The log file is not emptied at Alkaline startup - it is your responsibility to rotate the logs.

Example

```
LogFile=/usr/local/alkaline/log/search.log
```

ExactSize

Name

`ExactSize` — exact search word length

Synopsis

ExactSize = *0 / positive number*

Description

Maximum length of a word to search exact only. Exact searches are dramatically faster. It a CPU time loss to search for all words containing the character ’a’.

Example

With ExactSize=3, searching *in the tree* is equivalent to *"in" "the" tree*.

Default

```
ExactSize=1
```

WriteIndex

Name

`WriteIndex` — database write interval

Synopsis

WriteIndex = *`-1 / positive number`*

Description

Alkaline writes indexes after finding the amount of changed files equal to this setting or at the end of each roundtrip.

Default

```
WriteIndex=100
```

Example

Write indexes at the end of the roundtrip only:

```
WriteIndex=-1
```

RegExp

Name

`RegExp` — enable regular expressions

Synopsis

RegExp = *`Y / N`*

Description

Globally enable regular expressions rather than partial matches for options that support it. Currently these include UrlReplace, UrlExclude, UrlInclude, UrlSkip, ExcludeWords, IndexWords, IncludePages and ExcludePages. Contents of UrlExcludeFile, UrlIncludeFile and UrlSkipFile, populating UrlExclude, UrlInclude and UrlSkip respectively, also comply to this directive.

Regular expressions can contain the following special control characters:

Table 1. RegExp Control Characters

^	Beginning of the string. The expression ^*A* will match an *A* only at the beginning of the string.
^	The caret (^) immediately following the left-bracket ([) has a different meaning. It is used to exclude the remaining characters within brackets from matching the target string. The expression *[^0-9]* indicates that the target character should not be a digit.
$	The dollar sign (*$*) will match the end of the string. The expression *abc$* will match the sub-string *abc* only if it is at the end of the string.
|	The alternation character (|) allows either expression on its side to match the target string. The expression *a|b* will match *a* as well as *b*.
.	The dot (.) will match any character.
*	The asterisk (*) indicates that the character to the left of the asterisk in the expression should match 0 or more times.
+	The plus (+) is similar to asterisk but there should be at least one match of the character to the left of the + sign in the expression.
?	The question mark (*?*) matches the character to its left 0 or 1 times.
()	The parenthesis affects the order of pattern evaluation.

[ ]	Brackets (*[* and *]*) enclosing a set of characters indicates that any of the enclosed characters may match the target character.

The parenthesis, besides affecting the evaluation order of the regular expression, also serves as tagged expression which is something like a temporary memory. This memory can then be used when we want to replace the source expression with a replace expression. The replace expression can specify an & character which means that the & represents the sub-string that was found. So, if the sub-string that matched the regular expression is *abcd*, then a replace expression of *xyz&xyz* will change it to *xyzabcdxyz*. The replace expression can also be expressed as *xyz\0xyz*. The *\0* indicates a tagged expression representing the entire sub-string that was matched. Similarly you can have other tagged expression represented by *\1*, *\2* etc. Note that although the tagged expression 0 is always defined, the tagged expression 1, 2, etc. are only defined if the regular expression used in the search had enough sets of parenthesis. Here are few examples:

Table 2. RegExp Examples

String	Search	Replace	Result
Mr.	(Mr)(\.)	\1s\2	Mrs.
abc	(a)b(c)	&-\1-\2	abc-a-c
bcd	(a|b)c*d	&-\1	bcd-b
abcde	(.*)c(.*)	&-\1-\2	abcde-ab-de
cde	(ab|cd)e	&-\1	cde-cd
	([0-9,A-Z,a-z,\ ]*)(STOP:)([0-9,A-Z,a-z,\ ]*) -> \1\2	foo bar STOP: lkasdfkjakjlf	foo bar STOP:

Alkaline has command line parameters, such as *rxmatch* and *rxrepl* to test regular expressions. For more information, please refer to the Testing Regular Expressions section.

Example of Global RegExp Option

Exclude the entire /bar section from http://www.foo.com and both words, Foo and foo. Also, replace www by ns in all urls.

```
RegExp=Y
UrlExclude=http://www.foo.com/bar/.*
ExcludeWords=foo/words.regexp
UrlReplace (.*)(www)(.*)=\1ns\3
```

The words.regexp file contains:

```
[Ff]oo
```

Example of Scoped RegExp Option

Exclude the entire /bar section from http://www.foo.com. The list of words is not a list of regular expressions. Also, replace www by ns in all urls.

```
RegExp UrlExclude=http://www.foo.com/bar/.*
RegExp UrlReplace (.*)(www)(.*)=\1ns\3
ExcludeWords=foo/words.txt
```

Notes

This feature was added in version 1.3 (02-Jul-2000). The regular expressions replacements were added in version 1.3 (12-Jul-2000). Support for *RegExp [Option]* expressions was added in version 1.6.0824.0.

If you use an ExcludeWords directive with RegExp=Y, the default English dictionary supplied in the distribution cannot be used. It contains a .* expression which will exclude all possible word combinations.

CustomMetas

Name

`CustomMetas` — define custom meta tags for search results

Synopsis

CustomMetas `* / http-equiv1 [, http-equiv2][, ...]`

Description

Define all, one or several custom meta-tags to store in the index. The meta tag values are accessible using their names in the <!–SET MAP–...–> and other tags alike.

This is very useful to output a different modification date, an another url or offer more extensive document descriptions or site-specific information.

This option was added in version 1.32. The **CustomMetas=*** extension was added in version 1.6.0825.0.

Example

```
CustomMetas=MS.LOCALE,AUTHOR
```

```
CustomMetas=*
```

ReplaceLocal

Name

`ReplaceLocal` — local url string replacement

Synopsis

ReplaceLocal *`source = target`*

Description

Define a local text replacement for all indexed urls. The replacement is performed before the url is retrieved remotely. If a replacement occurred, a local file with the resulting url will be tentatively retrieved. In case of failure, including non-existent files and read errors, the remote url will be retrieved normally.

Using this option, you might dramatically speed-up the indexing process. Moreover, this directive supports virtual if-Modified-Since. A file that has a modification time-stamp in the past from the one in the current database, the contents will be reported as 304 Not Modified. When used in conjunction with the Index.html directive, documents such as http://www.foo.com/ can be retrieved locally as well.

With the RegExp option enabled, the source and target parameters must be regular expressions. For regular expressions syntaxes, see the RegExp section. This option has the *RegExp ReplaceLocal* extension since version 1.6.

This option was added in version 1.32.

Example

```
RegExp=N
```

```
ReplaceLocal http://foo.com=/home/web/foo
```

SearchPartialLeft

Name

`SearchPartialLeft` — constraint search to left/right wildcards

Synopsis

SearchPartialLeft = *`Y / N`*

Description

Enable partial left searches. When loading large indexes, set this option to N - this will remove deep searches such as *foo* and enable foo* searches only, producing less relevant, but much faster results.

Default

```
SearchPartialLeft=Y
```

ParseContent

Name

`ParseContent` — define html content-types

Synopsis

ParseContent = *`[*,]content-type[,[-]content-type]`*

Description

Define html-parseable content-types. When this option is defined, only pages matching the defined content-type or with the content-type omitted will be parsed for rich html content.

Adding * to the list will enable any content-type. Prefixing an entry with - will exclude a content-type. Prefixing an entry with + will include a content-type (+ can be omitted).

> Not all servers return content-type headers. In this case, Alkaline will parse the document. Using ReplaceLocal in conjunction with this option will most likely void this option as local filesystem is not aware of the document content-type.

This option was added in version 1.4.

Default

```
ParseContent=*
```

Example

```
ParseContent=-text/plain,+text/html,text/wml
```

ParseMetas

Name

`ParseMetas` — define parseable html meta content

Synopsis

ParseMetas = `[*,]name[,[-]name]`

Description

Define html-parseable meta content. When this option is defined, only meta tags matching an entry will be parsed.

Adding * to the list will enable any meta name. Prefixing an entry with - will exclude a meta name. Prefixing an entry with + will include a meta name (+ can be omitted).

This option is the first criteria applied to the meta tags being parsed. If you exclude all meta tags, Alkaline-specific meta data will not be available either.

This option was added in version 1.5.

Default

```
ParseMetas=*
```

Example

```
ParseMetas=-description,+filename,custom
```

Weight

Name

`Weight` — ranking parameters

Synopsis

Weight *`Meta = 0 / positive number < 16`*

Description

Specify the weight to give to each of the parameters used to rank the results. Currently supported metas are:

Table 1. Weight Metas

Name	Description	Default Value
Title	document title	10
Keywords	meta keywords	8
Description	meta description	6
Text	document plain text	3

Title and Keywords often reflect accurately the content of the page, you might

> want to give these parameters a higher value. The text in the body of the pages often contains many words not related to the subject, thus the parameter Text should have a lower value than other parameters.

This option was added in version 1.5.

Defaults

```
Weight Title=10

Weight Keywords=8

Weight Description=6

Weight Text=3
```

WeakWords

Name

`WeakWords` — frequent weak words

Synopsis

WeakWords = *`list of words separated by a comma`*

Description

List of words which will be given a weak weight in the ranking. Such words will be searched, however their weight in the ranking calculations will be reduced compared to other words.

The words are matched case insensitive, regular expressions and equivalent WeakWordsFile option are currently not supported for this directive.

This option was added in version 1.5.

Example

```
WeakWords=it,she,for
```

MaxWordSize

Name

`MaxWordSize` — maximum size of indexed words

Synopsis

MaxWordSize = *`positive number`*

Description

Defines the maximum size of the words in the index. Words bigger than this size will not be indexed.

There is an absolute limit of 5000 characters to the size of words. It is impossible to index words longer than 5000 characters with Alkaline.

Default

```
MaxWordSize=32
```

SearchCacheLife

Name

`SearchCacheLife` — cache record life span

Synopsis

SearchCacheLife = *`positive number`*

Description

Maximum number of seconds to keep a record in the search cache. The search cache can dramatically increase the performance and the responsiveness of your search engine, especially for large queries. It is in average 10 to 100 times faster to fetch an existing record in the cache rather than performing an exhaustive search in the index. The cache is adjusting its size depending on the server load and the amount of requests per second performed.

This option was added in version 1.7.1113.0.

Default

```
SearchCacheLife=60
```

II. Known global.cnf Variables

Variables from this section were previously defined in the equiv/equiv.struct file. The later is supported for legacy compatibility only in all versions.

Known global.cnf Variables

LogPath

Name

`LogPath` — location of the log files

Synopsis

LogPath = *`path`*

Description

Generate a log file called asearch-port.log in the path defined by this option. Such a log file contains all HTTP requests and all system level messages produced by Alkaline.

Example

```
LogPath=/usr/local/alkaline/log
```

Proxy

Name

`Proxy` — define an HTTP proxy

Synopsis

Proxy = *`server:port`*

Description

System level proxy used to retrieve all HTML documents, such as the template pages or server-side virtual includes. The Alkaline spider will use the Proxy entry in the asearch.cnf file if any. Otherwise it will use this entry.

Example

```
Proxy=proxy.foo.com:80
```

CacheTemplates

Name

`CacheTemplates` — cache search templates

Synopsis

CacheTemplates = *`Y / N`*

Description

When set to N, templates will not be cached. By default, Alkaline will load a search template on demand and save it for future requests.

This option is mostly useful when using a .aln template reference file which points to a cgi document, or to an html document with server-side includes that needs to be processed by the web server at each request. Setting *CacheTemplates=N* will force Alkaline to retrieve the page for every request and thus execute the remote cgi or ssi for each search.

Default

```
CacheTemplates=Y
```

UsFormats

Name

`UsFormats` — us-formatted dates

Synopsis

UsFormats = *`Y / N`*

Description

When set to Y, us formats apply to dates for the before: and after: options in search query strings. A US date is formatted Month-Day-Year and a European date is of the Day-Month-Year format.

Default

```
UsFormats=N
```

Realm

Name

`Realm` — basic auth realm

Synopsis

Realm = *`string`*

Description

The Realm to show for basic authentication on the popup that appears when trying to access the management section. The default header returned is *WWW-Authenticate: BASIC Realm="Alkaline Certified Server"*. Browsers save passwords and usernames according to the realm.

Default

```
Realm=Alkaline Certified Server
```

ErrorFooter

Name

`ErrorFooter` — error footer string

Synopsis

ErrorFooter = *`string`*

Description

The bottom text of a server error. You can use this option with any html or text value. It can allow, for example, to redirect client to a particular page on any system error.

Default

```
ErrorFooter=<hr> Alkaline Search Engine © 1994-
2000 Vestris Inc., All Rights Reserved
```

KeepAlive

Name

`KeepAlive` — allow to keep-alive clients

Synopsis

KeepAlive = *Y / N*

Description

Keep-alive enables HTTP clients that support the HTTP/1.1 protocol to keep the connection to the server opened and pipeline multiple requests over it. This can significantly improve the speed at which search results appear, as browsers do not need to re-negotiate a new session for every request.

This option was added in version 1.32.1028.0.

Although adding KeepAlive=Y may improve overall performance of the software, it is considered experimental.

Default

```
KeepAlive=N
```

Nagle

Name

`Nagle` — disable nagle algorithm

Synopsis

Nagle = *Y / N*

Description

The Nagle algorithm says that the server should delay sending partial packets in hopes of getting more data. There are bad interactions between persistent connections and Nagle's algorithm that have very severe performance penalties.

This option was added in version 1.4.0317.0.

Although adding Nagle=N may improve overall performance of the software, it is considered experimental.

Default

`Nagle=Y`

Ssi

Name

`Ssi` — enable server-side includes

Synopsis

Ssi = *`Y / N`*

Description

Server-Side includes is an NCSA standard that allows users to create documents that provide simple information to clients on the fly. Such information can include the current date, the file’s last modification date, and the size or last modification of other files. In it’s more advanced usage, it can provide a powerful interface to CGI and /bin/sh programs.

Alkaline has limited support for server-side includes, which must be enabled using this option or the command-line equivalent. The SSI support is fully described in this section.

This option was added in version 1.4.0423.0.

Default

```
Ssi=N
```

RampupSearchThreads

Name

`RampupSearchThreads` — rampup search thread pool threads

Synopsis

RampupSearchThreads = *`positive number or zero`*

Description

Alkaline provides a pool of threads for the search operations. This pool can have a maximum of *MaxSearchThreads* threads. New jobs are added to a queue of *MaxSearchQueueSize* elements. As a new job arrives and all threads a busy, at least one new thread will be created as long as the thread pool size is below *MaxSearchThreads*.

Optionally *RampupSearchThreads* will be created in addition to service anticipated load. Each search thread will die if idle for over *MaxSearchThreadIdle* seconds.

This option was added in version 1.4.0423.0.

Default

```
RampupSearchThreads=0
```

MaxSearchThreads

Name

`MaxSearchThreads` — maximum search thread pool threads

Synopsis

MaxSearchThreads = *`positive number`*

Description

Alkaline provides a pool of threads for the search operations. This pool can have a maximum of *MaxSearchThreads* threads. New jobs are added to a queue of *MaxSearchQueueSize* elements. As a new job arrives and all threads a busy, at least one new thread will be created as long as the thread pool size is below *MaxSearchThreads*. Optionally *RampupSearchThreads* will be created in addition to service anticipated load. Each search thread will die if idle for over *MaxSearchThreadIdle* seconds.

This option was added in version 1.4.0423.0 and is equivalent to the *–mt* command line option.

Default

```
MaxSearchThreads=25
```

MaxSearchQueueSize

Name

`MaxSearchQueueSize` — maximum search thread pool queue

Synopsis

MaxSearchQueueSize = *`positive number`*

Description

Alkaline provides a pool of threads for the search operations. This pool can have a maximum of *MaxSearchThreads* threads. New jobs are added to a queue of *MaxSearchQueueSize* elements. As a new job arrives and all threads a busy, at least one new thread will be created as long as the thread pool size is below *MaxSearchThreads*. Optionally *RampupSearchThreads* will be created in addition to service anticipated load. Each search thread will die if idle for over *MaxSearchThreadIdle* seconds.

This option was added in version 1.4.0423.0.

Default

```
MaxSearchQueueSize=24
```

MaxSearchThreadIdle

Name

`MaxSearchThreadIdle` — maximum search thread idle time

Synopsis

MaxSearchThreadIdle = *`zero or positive number`*

Description

Alkaline provides a pool of threads for the search operations. This pool can have a maximum of *MaxSearchThreads* threads. New jobs are added to a queue of *MaxSearchQueueSize* elements. As a new job arrives and all threads a busy, at least one new thread will be created as long as the thread pool size is below *MaxSearchThreads*. Optionally *RampupSearchThreads* will be created in addition to service anticipated load. Each search thread will die if idle for over *MaxSearchThreadIdle* seconds.

This option was added in version 1.4.0423.0.

Default

```
MaxSearchThreadIdle=15
```

MaxIndexThreads

Name

`MaxIndexThreads` — maximum index threads

Synopsis

MaxIndexThreads = *`positive number`*

Description

Alkaline provides a pool of threads for indexing. This pool can have a maximum of *MaxIndexThreads* threads. As a new job arrives and all threads a busy, at least one new thread will be created as long as the thread pool size is below this number. Optionally *RampupIndexThreads* will be created in addition to service anticipated load.

This option was added in version 1.4.0423.0 and is equivalent to the *-mi* command line option.

Default

```
MaxIndexThreads=25
```

RampupIndexThreads

Name

`RampupIndexThreads` — rampup index threads

Synopsis

RampupIndexThreads = *`positive number or zero`*

Description

Alkaline provides a pool of threads for indexing. This pool can have a maximum of *MaxIndexThreads* threads. As a new job arrives and all threads a busy, at least one new thread will be created as long as the thread pool size is below this number. Optionally *RampupIndexThreads* will be created in addition to service anticipated load.

This option was added in version 1.4.0423.0.

Default

```
RampupIndexThreads=0
```

III. Known global.cnf Passwords

Known global.cnf Passwords

Pass Root

Name

`Pass Root` — general purpose password

Synopsis

Pass Root = *`password`*

Description

General purpose password valid for all server protected operations.

Never set this password to your system root, superuser or Administrator password. If you are concerned about this password being trasmitted over a non-secure HTTP channel, you can hide Alkaline behind an Apache SSL server as described in the Alkaline FAQ.

Example

```
Pass root=rootpassword
```

Pass Alkaline-Restart

Name

`Pass Alkaline-Restart` — restart the server password

Synopsis

Pass Alkaline-Restart = *`password`*

Description

Password valid to restart the server (when available).

Example

```
Pass alkaline-restart=foopassword
```

Pass Alkaline-Add

Name

`Pass Alkaline-Add` — add urls password

Synopsis

Pass Alkaline-Add = *`password`*

Description

Password valid for adding, removing and reindexing urls from the online management.

Example

```
Pass alkaline-add=barpassword
```

Pass Alkaline-Manage

Name

`Pass Alkaline-Manage` — access the online management

Synopsis

Pass Alkaline-Manage = *`password`*

Description

Password valid for a read-only access to the online admin section.

Example

```
Pass alkaline-manage=foopassword
```

AdminPath

Name

`AdminPath` — location of the administrative path

Synopsis

AdminPath = *`relative path`*

Description

Define the relative location of the administrative section files. This cannot be an absolute location, a different drive or share. This is the only path where Alkaline will prompt for the administrative password. You should copy the contents of the *admin* directory from the Alkaline distribution to this location.

Default

```
AdminPath=admin
```

DocumentPath

Name

`DocumentPath` — plain document paths

Synopsis

DocumentPath = *`comma-separated list of relative paths`*

Description

Define the relative locations of plain documents. This allows to server plain http documents via Alkaline by using it as a standard web server. The paths must be relative to the directory from which Alkaline is started.

This option was added in version 1.7.1223.0.

Example

```
DocumentPath=docs,server/stats
```

ForwardAlnHeaders

Name

`ForwardAlnHeaders` — forward headers for aln templates

Synopsis

ForwardAlnHeaders = *`comma-separated list of header names`*

Description

Forward client headers for retrieving .aln templates. This option should be used with the CacheTemplates option set to N.

This option was added in version 1.8.2215.0.

Example

```
ForwardAlnHeaders=Accept-encoding,Accept-language
```

Redirect

Name

`Redirect` — default redirect

Synopsis

Redirect = *`url`*

Description

Defines the default url to redirect clients to when not accessing a particular configuration. By default, the client is redirected to the Alkaline administrative section.

Default

```
Redirect=admin
```

Ping

Name

`Ping` — enable ping thread

Synopsis

Ping = *`Y / N`*

Description

Enables the self ping monitoring process. The background thread will attempt to request the PingUrl document from the running Alkaline in order to detect any anomaly in server responsiveness. The request will be performed every PingInterval seconds and at most PingRestart failures in a row. If this number is exceeded, the server will be restarted.

Enabling the background ping thread can be done using the *–enableping* command line option. The ping mechanism is functional on UNIX platforms only. Ping options were added in Alkaline 1.7.1112.0.

Default

```
Ping=N
```

PingInterval

Name

`PingInterval` — ping thread request interval

Synopsis

PingInterval = *`positive number`*

Description

The background Ping thread will attempt to request the PingUrl document from the running Alkaline in order to detect any anomaly in server responsiveness. The request will be performed every PingInterval seconds and at most PingRestart failures in a row. If this number is exceeded, the server will be restarted.

The ping mechanism is functional on UNIX platforms only. Ping options were added in Alkaline 1.7.1112.0.

Default

```
PingInterval=30
```

PingRestart

Name

`PingRestart` — failed ping restart count

Synopsis

PingRestart = *`positive number`*

Description

The background Ping thread will attempt to request the PingUrl document from the running Alkaline in order to detect any anomaly in server responsiveness. The request will be performed every PingInterval seconds and at most PingRestart failures in a row. If this number is exceeded, the server will be restarted.

The ping mechanism is functional on UNIX platforms only. Ping options were added in Alkaline 1.7.1112.0.

Default

```
PingRestart=3
```

PingUrl

Name

`PingUrl` — url to ping periodically

Synopsis

PingUrl = *`url`*

Description

The background Ping thread will attempt to request the PingUrl document from the running Alkaline in order to detect any anomaly in server responsiveness. The request will be performed every PingInterval seconds and at most PingRestart failures in a row. If this number is exceeded, the server will be restarted.

The ping url is automatically detected and defaults to the server name and the port that Alkaline is binding to.

The ping mechanism is functional on UNIX platforms only. Ping options were added in Alkaline 1.7.1112.0.

IV. Search Templates Tags and Options

Search Templates Tags and Options

SEARCH-RESULTS

Name

`SEARCH-RESULTS` — search results

Synopsis

<!– **SEARCH-RESULTS** –>

Description

Output search results if any. Each result is shown with the format of the SET MAP option.

Example

```
Alkaline has found the following results:<hr>
<!-SEARCH-RESULTS->
```

SEARCH-GENERAL

Name

`SEARCH-GENERAL` — search operation variables

Synopsis

<!– **SEARCH-GENERAL** – *`expression`* –>

Description

Maps global search operation variables, such as the amount of results.

Table 1. Variables

time	search time elapsed in seconds
search	search string used; use <!–SET FREECHARSET–1–> if you do not want this string to be quoted
query	cleaned search string, without url:, path: and other parameters; use <!–SET FREECHARSET–1–> if you do not want this string to be quoted
total	total results found
size	number of total indexed documents (added in 1.7.0527.0)
quant	total results shown
start	index of the first result output on this page (added 02-Jul-2000)
end	index of the last result output on this page (added 02-Jul-2000)
sort.size	url to sort results by size
sort.isize	url to sort results by size in the ascending order
sort.date	url to sort results by date

sort.idate	url to sort results by date in the ascending order
sort.title	url to sort results by title
sort.url	url to sort results by title in the ascending order
sort.domain	group results by domain name
sort.	url to sort results by relevance
url	form posted *url* restrictions
host	form posted *host* restrictions
path	form posted *path* restrictions
other	form posted *other* restrictions
before	form posted *before* restrictions
after	form posted *after* restrictions
post.[name]	all variables resulting from the post method are defined with the *post.* prefix; you can thus pass any variable to the Alkaline search results mapping engine
requesturl	current query url (added in 1.7.1211.0)

Example

Classic example:

```
<!-SEARCH-GENERAL
 $search~[searching for <b>]#[</b>]^[nothing to search]
->
```

Passing a value from <input type="hidden" name="session" value="foo">:

```
<a href="http://www.foo.com/reg/<!-SEARCH-GENERAL
 £POST.SESSION~[
  ?session=$POST.SESSION
 ]->"
 ONMOUSEOVER="switchOn (1); return true"
 ONMOUSEOUT="switchOff (1); return true"
>
 Register!
</a>
```

SEARCH-NEXT

Name

`SEARCH-NEXT` — links to more results

Synopsis

<!– **SEARCH-NEXT** –>

Description

Shows links to more pages of results. This tag expands into a link to the previous page (formatted using the SET PREV option), a list of 10 (defined by the SET NEXT-DIVISION option) pages and a link to the next page (formatted using the SET NEXT option).

Example

```
<!-SET PREV-Previous page->
<!-SET NEXT-Next page->
<!-SEARCH-NEXT->
```

SET MAP

Name

`SET MAP` — format each search result

Synopsis

<!– **SET MAP** – *`expression`* –>

Description

Defines the output of each individual result in SEARCH-RESULTS.

With server-side includes it is possible to execute commands for each individual result. This enables rendering per-result content with custom scripts (added in version 1.7.1522.0).

```
<!-SET MAP-<-#exec cmd="echo %DATE% - result $index"-?> ...->
```

Table 1. Variables

url	the url of the result
words	words used for searching
modif	last modification date of the result of format defined by SET DATE
date	date when the page was fetched by the spider or, if the server does support creation dates, date of page creation; of format defined by SET DATE
modif.french	*$modif* in French; of format *$dayfrench, $day $monthfresh $year ($hour:$min)*

date.french	*$date* in French; of format *$dayfrench, $day $monthfresh $year ($hour:$min)*
title	title of the page; if not available, *$url*
hltitle	*$title* with *$query* keywords highlighted with SET HIGHLIGHT-OPEN and SET HIGHLIGHT-CLOSE (added in version 1.7)
header	the *meta description* tag value or the first HeaderLength characters from the document's text
hlheader	*$header* with *$query* keywords highlighted with SET HIGHLIGHT-OPEN and SET HIGHLIGHT-CLOSE (added in version 1.7)
index	search result index, starting from 1
recent	value assigned by SET RECENT, when the result is not older than X days, assigned by SET RECENT-COUNT
expired	value assigned by SET EXPIRED, when the result is older than X days, assigned by SET EXPIRED-COUNT
age	number of days since the last time the document was modified
quality	quality, in percent, of the current result relative to other results and searched keywords
group.size	when grouped by domain (sort:domain), size of the group
sort	sorting option

...	all variables defined by the CustomMetas directive in the asearch.cnf
...	all variables available in the SEARCH-GENERAL section

Example

```
<!-SET MAP-
 <dl><dt>
 <a href="$url">
  <b>$title</b>
 </a>
 <dd>$header ... <br>
 <font color="gray">$size bytes; modified: $modif</font>
 <br>url: <a href="$url">$url</a>
 </dl>
->
```

SET NEXT-DIVISION

Name

`SET NEXT-DIVISION` — number of page links

Synopsis

<!– **SET NEXT-DIVISION** – *`positive number`* –>

Description

Defines the amount of pages shown with SEARCH-NEXT when available.

Default

```
<!-SET NEXT-DIVISION-10->
```

SET NEXT

Name

`SET NEXT` — link to the next page

Synopsis

<!– **SET NEXT** – *`expression`* –>

Description

Defines the text output as the link to the next page of results when available.

Default

```
<!-SET NEXT-next->
```

Example

```
<!-SET NEXT-
 <img src="http://www.foo.com/images/next.gif">
->
```

SET PREV

Name

`SET PREV` — link to the previous page

Synopsis

<!– **SET PREV** – *`expression`* –>

Description

Defines the text output as the link to the previous page of results when available.

Default

```
<!-SET PREV-prev->
```

Example

```
<!-SET PREV-
 <img src="http://www.foo.com/images/prev.gif">
->
```

SET NEXT-INHERIT

Name

`SET NEXT-INHERIT` — pass form values to search pages

Synopsis

<!– **SET NEXT-INHERIT** – *`expression`* –>

Description

Allows to pass form values to search results pages.

The link to other than the first search results pages is a GET rather than a POST. To preserve random values of the posted form, this option must be used.

Example

To append the value of the *dummy* option defined as <input type="hidden" name="dummy" value="dummyvalue"> on the search form, use:

```
<!-SET NEXT-INHERIT-&dummy=$post.dummy->
```

The following url request string will be generated:

```
/?DataAlias+HTMLAlias+search=searchstring+startXX&dummy=dummyvalue
```

SET SEARCH-NORESULTS

Name

`SET SEARCH-NORESULTS` — output when no results found

Synopsis

<!– **SET SEARCH-NORESULTS** – *`expression`* –>

Description

Expression to use when replacing the SEARCH-RESULTS tag, when no results were found.

Example

```
<!-SET SEARCH-NORESULTS-
 <b>No matching document found!</b>
->
```

SET SEARCH-BASE-HREF

Name

`SET SEARCH-BASE-HREF` — base url of links to search results

Synopsis

<!– **SET SEARCH-BASE-HREF** – *url* –>

Description

Use *url* as base href of search results. Normally, Alkaline generates links relative to the server root. This option is useful when wrapping Alkaline with a search CGI or when using server proxy on a different port.

Example

```
<!-SET SEARCH-BASE-HREF-
 http://search.foo.com/cgi-bin/search.cgi?query=
->
```

SET SEARCH-BASE-ABS

Name

`SET SEARCH-BASE-ABS` — absolute base of search results links

Synopsis

<!– **SET SEARCH-BASE-ABS** – *expression* –>

Description

Use *expression* as the absolute base href of search results. Links to previous and next pages are formed of this value, *start* and *quant* parameters along with any other inherited option. This option is useful when wrapping Alkaline with a search CGI or when using server proxy on a different port in addition to a search query mapper.

This option was added in version 1.7.1211.0.

Default

```
<!-SET SEARCH-BASE-ABS-$requesturl->
```

SET C-BEFORE

Name

`SET C-BEFORE` — insert before the current page

Synopsis

<!– **SET C-BEFORE** – *`expression`* –>

Description

Insert *expression* before the current page in links to pages of results generated by SEARCH-NEXT.

Example

```
<!-SET C-BEFORE-<font color=red>->
<!-SET C-AFTER-</font>->
```

SET C-AFTER

Name

`SET C-AFTER` — insert after the current page

Synopsis

<!– **SET C-AFTER** – *`expression`* –>

Description

Insert *expression* after the current page in links to pages of results generated by SEARCH-NEXT.

Example

```
<!-SET C-BEFORE-<font color=red>->
<!-SET C-AFTER-</font>->
```

SET N-BEFORE

Name

`SET N-BEFORE` — insert before links to results pages

Synopsis

<!– **SET N-BEFORE** – *`expression`* –>

Description

Insert *expression* before a link to a page of results output by SEARCH-NEXT, except to the current page.

Example

```
<!-SET N-BEFORE-<b>->
<!-SET N-AFTER-</b>->
```

SET N-AFTER

Name

`SET N-AFTER` — insert after links to results pages

Synopsis

<!– **SET N-AFTER** – *`expression`* –>

Description

Insert *expression* after a link to a page of results output by SEARCH-NEXT, except to the current page.

Example

```
<!-SET N-BEFORE-<b>->
<!-SET N-AFTER-</b>->
```

SET DATE

Name

`SET DATE` — format date fields

Synopsis

<!– **SET DATE** – *`expression`* –>

Description

Format $date and $modif fields in SET MAP, SEARCH-GENERAL, etc.

Table 1. Variables

DayEnglish	full day of week in English
DayEng	short day of week in English
DayFrench	full day of week in French
DayFre	short day of week in French
MonthFrench	full month in French
MonthFre	short month in French
MonthEnglish	full month in English
MonthEng	short month in English
Year	4 digits year
Day	day of month
Month	2 digits month
Hour	2 digits hour
Min	2 digits minutes
Sec	2 digits seconds

Default

```
<!-SET DATE-
 $dayenglish, $monthenglish $day, $year ($hour:$min)
->
```

SET RECENT-COUNT

Name

`SET RECENT-COUNT` — consider a document recent

Synopsis

<!– **SET RECENT-COUNT** – `0 / positive number` –>

Description

Number of days a document is considered recent, used by SET MAP *$recent* variable.

Example

```
<!-SET RECENT-COUNT-3->
```

SET EXPIRED-COUNT

Name

`SET EXPIRED-COUNT` — consider a document expired

Synopsis

<!– **SET EXPIRED-COUNT** – `0 / positive number` –>

Description

Number of days after which a document is considered expired, used by SET MAP *$expired* variable.

This option was added 03-Aug-2000.

Example

```
<!-SET EXPIRED-COUNT-365->
```

SET RECENT

Name

`SET RECENT` — recent value

Synopsis

<!– **SET RECENT** – *`expression`* –>

Description

Assign *expression* as value of the *$recent* variable of the SET MAP option when the document has been recently modified. A document is considered recent if it has been modified during the past number of days set by the SET RECENT-COUNT option.

Example

```
<!-SET RECENT-
 <img src="http://www.foo.com/images/new.gif">
->
```

SET EXPIRED

Name

`SET EXPIRED` — expired value

Synopsis

<!– **SET EXPIRED** – *`expression`* –>

Description

Assign *expression* as value of the *$expired* variable of the SET MAP option when the document has not been modified for a period of time, defined by the SET EXPIRED-COUNT option.

This option was added 03-Aug-2000.

Example

```
<!-SET EXPIRED-
 <img src="http://www.foo.com/images/old.gif">
->
```

SET QUANT

Name

`SET QUANT` — amount of results per page

Synopsis

<!– **SET QUANT** – *`-1 / positive number`* –>

Description

Define the amount of results to show per page. A negative value means show all available results.

This might be tens of thousands of results which is probably not what any user ever wants to see.

Default

```
<!-SET QUANT-10->
```

SET FREECHARSET

Name

`SET FREECHARSET` — do not quote output values

Synopsis

<!– **SET FREECHARSET** – *`0/1`* –>

Description

Avoid quoting results. Use this option in conjunction with *FreeCharset=Y* in the asearch.cnf.

Default

```
<!-SET FREECHARSET-0->
```

SET QUOTE

Name

`SET QUOTE` — force quoting of search results

Synopsis

<!– **SET QUOTE** – *`0/1`* –>

Description

Force quoting search results; é will become &eacute;. This option does not depend from the FREECHARSET one and quotes the text for each search page result.

Default

```
<!-SET QUOTE-0->
```

SET HIGHLIGHT-OPEN

Name

`SET HIGHLIGHT-OPEN` — left highlight

Synopsis

<!– **SET HIGHLIGHT-OPEN** – *`string`* –>

Description

Sets the left side of a highlight for the *$hlheader* rendered variable. Allows to highlight keywords from the search results within each of the results header.

This option was added in version 1.7.

Default

```
<!-SET HIGHLIGHT-OPEN-<b>->
```

SET HIGHLIGHT-CLOSE

Name

`SET HIGHLIGHT-CLOSE` — right highlight

Synopsis

<!– **SET HIGHLIGHT-CLOSE** – *`string`* –>

Description

Sets the right side of a highlight for the *$hlheader* rendered variable. Allows to highlight keywords from the search results within each of the results header.

This option was added in version 1.7.

Default

```
<!-SET HIGHLIGHT-CLOSE-</b>->
```

#tag name=value

Name

`#tag name=value` — server side includes

Synopsis

<!– **#tag** *`name=value`* –>

Description

Server side includes partial support. Please read the Server Side Includes section for details.

V. Database Formats

Alkaline index databases are simple text files with no limitation in number of elements or size of each field. Alkaline uses fully dynamic structures for all objects. This should be taken in consideration developing side products using indexes generated by the spider.

The siteidx?.* files contain a digit instead of the ?, for example siteidx1.ndx. This defines the version of the index files. The current documentation explains version 1 only. In case a structural modification is needed, the version is incremented - you will never find a running Alkaline with different numbers in different siteidx?.* files.

Database Formats

siteidx1.ndx

Name

`siteidx1.ndx` — unique words index (obsolete)

Description

The siteidx1.ndx file is the main index file containing words and pages containing those words. Each line is one entry. Each field of the entry is separated by a space. The first field is the word, all other fields are unique page ids. Each page id is the number corresponding to an entry in the siteidx1.urt. Url lists are not written as *1 2 3 4 5 8 14 15 16*, but as intervals, *1-5 8 14-16*. Alkaline is capable of automatically loading lists that are not intervals. There must be no duplicate values in the array.

Example

```
word 1 4-16 234 567
```

siteidx2.ndx

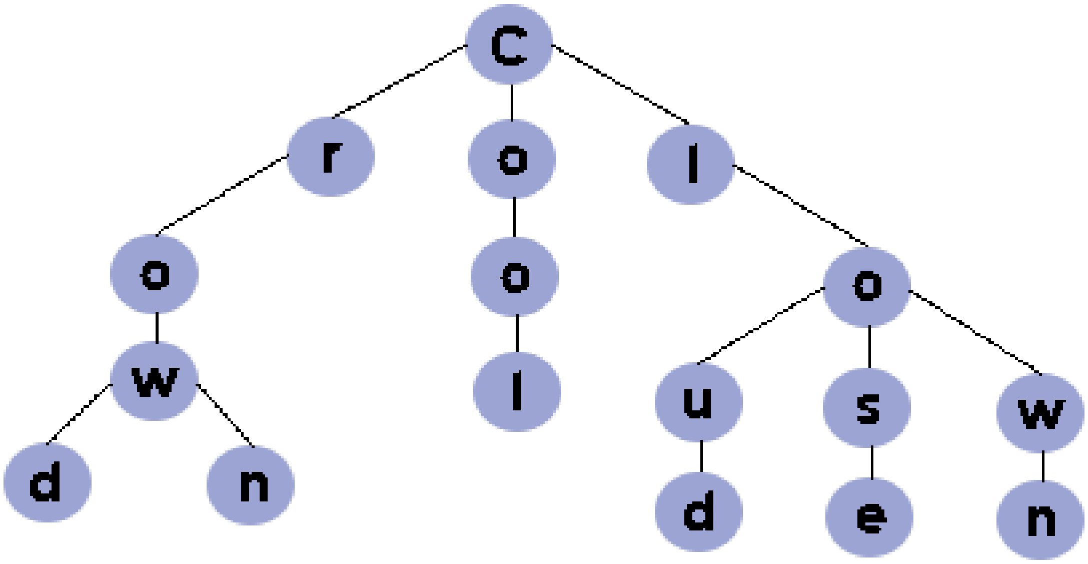
Name

`siteidx2.ndx` — unique words index

Description

The siteidx2.ndx file is a pure binary file. The format of the index was changed in version 1.4 to allow faster loading and writing of the index. In order to read the index, you must write a tool which will parse the binary file and extract the information you require. This index format is not backward compatible, an Alkaline older than version 1.4 will *not* be able to read a siteidx2.ndx format.

The idea behind the structure of the index is that you can represent all the words as a tree of letters, using one letter per node :



Some nodes will represent the last letter of a word so an array of page numbers will be associated to this node (the pages which contain this word), and other nodes which are not a final node for any word will not have any array of pages associated with them.

Such node can be defined as:

```
struct Node
{
    unsigned char Letter;
    unsigned int ChildNodesNumber;
    unsigned int PagesNumber;
    (PagesArray Pages;)
    (NodesArray ChildNodes;)
};
```

- *PagesArray* is an array of *unsigned int*. It is present only if *PagesNumber* is different from zero.
- *NodesArray* is an array of *Node*. It is present only if *ChildNodesNumber* is different from zero.

To read the tree, read the first node of the tree. The array of nodes present in the first node contains the second level nodes. Each second level node contains an array of nodes which represent the third level nodes, etc.

The structure of the *siteidx2.ndx* index file is the following:

```
unsigned char BinaryType;
unsigned int FirstTreeWordNumber;
(Tree FirstTree;)
unsigned int SecondTreeWords;
(Tree SecondTree;)
unsigned int ThirdTreeWordNumber;
(Tree ThirdTree;)
...
```

Since a tree has only one root and words can begin by many different letters, the number of trees present depends on the words contained in the index. For example, if the index contain the words *cool, crowd, bowling*, there will be two trees: one for the letter *c* and one for the letter *b*.

- *BinaryType* specifies if the index was written on a machine with a big-endian processor (value = 1) or a machine with a little-endian processor (value = 0).
- *FirstTreeWordNumber* specifies the number of words in this tree.
- *FirstTree* represents the tree of nodes for the first tree. This tree is present only if *FirstTreeWordNumber* is different from zero.

There are 256 possible trees specified, one for each value of the ASCII table, however only trees containing words will be written in the index. Trees which do not contain any word will have their value *XXXTreeWordNumber* set to zero, which means that there is no tree to read.

siteidx1.url

Name

`siteidx1.url` — index of uniform resource locators (obsolete)

Description

The siteidx1.url stores full urls that the spider has encountered. It also stores urls that might never be indexed. Each entry is a full url defined by RFC 1738: Uniform Resource Locators (T. Berners-Lee, L. Masinter, and M. McCahill, December 1994). Each entry is on a separate line. The line number corresponds to the url's unique id.

The .url files have been replaced by url trees (.urt). Alkaline will automatically convert any .url file into a .urt.

siteidx1.urt

Name

`siteidx1.urt` — uniform resource locator tree

Description

The siteidx1.urt file stores uniform resource locators (urls) that the spider has encountered. It also stores urls that might never be indexed. Each entry is a full url defined by RFC 1738: Uniform Resource Locators (T. Berners-Lee, L. Masinter, and M. McCahill, December 1994). Each entry is on a separate line and is of the following format:

```
[absolute index][space][level index][space][encoded url]
```

The level index is the depth in the tree. The absolute index is either the index of the terminal node (thus the index of the full url that can be obtained by adding all parent node values) or -1. Alkaline uses a fully linked tree to store URL information - a tree where each element is linked to it’s parent, next, previous, first and last child node.

Loading such a tree is extremely simple, here’s the pseudo-code:

```
Declare HorizontalVector as Array of Nodes;
Declare HorizontalPosition as Integer = [level index]
For each line in the file do {
 NewNode = Create new Node ( Unescape[encoded url] )
 if ( HorizontalVector.Size > 0 ) and ( HorizontalPosition == 0 ) {
       // this is a top node
       HorizontalVector.Add( NewNode )
       TreeHead = NewNode
 } else if ( HorizontalPosition == HorizontalVector.Size - 1) {
       // same level, add after element
       Add NewNode after Node in HorizontalVector[ HorizontalPosition ]
       HorizontalVector[ HorizontalPosition ] = NewNode
 } else if (HorizontalPosition > HorizontalVector.Size - 1) {
       // sub level, add child
       Add NewNode as Last Child of Node in HorizontalVector[ HorizontalPosition - 1 ]
       Add NewNode to HorizontalVector
 } else if (HorizontalPosition < HorizontalVector.Size - 1) {
       // previous level, remove and add
       do {
          Remove Last Element in HorizontalVector
       } while HorizontalPosition < HorizontalVector.Size;
       Add NewNode after Node in HorizontalVector[ HorizontalPo-
sition ]
       HorizontalVector[ HorizontalPosition ] = NewNode
 } else Error

 if [absolute index] != -1 {
   Mark that NewNode has an [absolute index].
   // Alkaline uses an associated array
 }
```

siteidx1.inf

Name

`siteidx1.inf` — digest information

Description

The siteidx1.inf file stores the digest information on each page. Each entry is on a separate line. Entry fields are separated by tabs. An entry with no or empty fields has not been indexed. Fields are listed in the following order:

- Date of creation, for example Fri, 05 Mar 1999 10:00:43 GMT.
- Date of modification, may be empty or same as date of creation.
- Page title, defined by the <TITLE> field.

- Part of unformatted text of the page, length is at most of asearch.cnf HeaderLength option value, default is 255; note that when HeaderLength is changed with an existing index, previous entries can be longer.
- Server where the page has been referenced from.
- MD5 digest of the page contents as defined by RFC 1321: MD5 Digest Algorithm, R. Rivest, MIT Lab for CS and RSA Data Security, Inc., 1992. Each control character (between 0 and 32) is encoded by a backslash and followed by a one or two digit number.

siteidx1.lnx

Name

`siteidx1.lnx` — id cross-index

Description

The siteidx1.lnx file contains links found from a page to other pages. Each entry is on a separate line, numbered in the ascending order. Each entry has fields separated by spaces and stored as intervals. This consists in writing a list like *1 2 3 4 5 8 14 15 16* as *1-5 8 14-16*. Each field is a page uid linked from the page with the uid of the entry number.

For example, if line 3 has entries such as *1-5 6 9*, it means that the page with the id of 3 has links to pages *1-5 6 9*.