ASPseek User's Guide

Table of Contents

NAME

ASPseek - Advanced Internet Search Engine

DESCRIPTION

ASPseek is an Internet search engine with all the bells and whistles you expect from such a product. This page introduces **ASPseek** for the new users, explains its basic concepts and provides references to more information.

ASPseek is written in C++, using STL library and mix of SQL server tables and binary files as a data storage. **ASPseek** consists of indexing program **index**(1), search daemon **searchd**(1), and CGI search front-end **s.cgi**(1).

index(1) walks across the sites and stores found pages in a special data structures (some data is stored in SQL, some other data that affects search speed are stored in binary files in `/usr/local/aspseek/var` directory). **searchd**(1) listens to and performs search queries, issued by search front-end **s.cgi**(1). Front-end then formats the results into nice-looking HTML page.

ASPseek is optimized for multiple sites and medium loads, and can be used for search within several million pages (URLs). User can search for words and phrases, use wildcards, and do a Boolean search.

Search scope can be limited to time period given, site, subdirectory of site, Web space (set of sites), or particular parts of HTML documents (notably title, body, description and keywords). Search results can be sorted by relevance or by date.

Below is a list of **ASPseek** features.

Ability to index and search through several millions of documents

Using **ASPseek**, you can build a database and search through many sites, and results for each query will be returned fast even if you have a few millions of documents indexed. Of course, this depends on hardware, so don't expect "good old" i486 machine to handle every site in .com domain. Everything depends on CPU(s), memory, disk speed etc. So do your own tests before you buy dedicated hardware.

The fact that **ASPseek** is optimized for high volumes should not stop you from using it to search your own site that contains few hundred of documents - it works there as well.

Very good relevancy of results

The purpose of search engine is to find what user wants. There can be thousands of URLs found as a result of search query, but it can all be irrelevant, so user will be unsatisfied.

Output results in **ASPseek** are sorted by relevancy (or rank), but rank calculation is not an easy task. Developers tried their best to incorporate greatest and latest techniques into **ASPseek** engine while maintaining good search speed.

Advanced search capabilities

User can ask search engine to search not only for the set of words, but also for the whole phrase. To search for phrase, just surround it with quotation marks, like this:

```
"many years ago"
```

Phrase searching considerably improves results, and this feature is rated to be the most useful by people.

If you do know the exact phrase, but forgot a singe word in the middle of the phrase, you can use askerisk mark (*****) instead of that word. So, query

```
"many * ago"
```

will return results with phrases like "*many years ago*", "*many days ago*" etc.

Boolean search is a search with logical expression. Expression can be composed using AND and OR operators, subexpressions can be grouped using parenthesis. Example:

```
(some OR any) AND (days OR months OR years)
```

Subexpression in parenthesis can be another boolean expression or just word, pattern, or phrase.

You can exclude the word from search by putting a minus sign before it. So, pages with that word will be excluded from search results. Example:

```
search engine -proprietary
```

Search by pattern allows to search documents containing words that match specified pattern. Character '?' means any character, character '*' means sequence of any characters. For example, to find all documents containing words beginning with provider, type:

```
provider*
```

ASPseek allows to narrow search up to one or few sites. For example, to find all documents containing word *bubble* on site *www.mysite.org*, type:

```
bubble site: www.mysite.org
```

The same way, if you want results from all sites parked at *mysite.com* (like *www.mysite.com*, *lib.mysite.com*, *smth.other.mysite.com*), you can just type:

```
bubble site: mysite.org
```

You can even use **site: org** to get results from all *.org* domains that are indexed.

Excluding the results from given site(s) are done in the same way:

```
bubble -site: mysite.org
```

Several **site:** limits can be used together.

You can also narrow result to pages modified (or created) within specified time period, which can be set in few ways: some time back from now, before/after given date, or between two dates. For example, you can narrow results to pages those were modified not earlier than one week ago.

And finally, you can find all pages those link to specified page, like this:

```
link: www.aspseek.org
```

It will show you all pages in the database that have links to *www.aspseek.org*.

You can combine any of the above facilities, as far as it makes sense.

Ispell support

When **ASPseek** is used with ispell support, **searchd**(1) can optionally find all forms for all specified words (example: *create* --> *create* OR *created* OR *creates*). So, it allows you to find the word in all of different forms.

Unicode storage mode

ASPseek can store information about documents in Unicode, thus making possible to implement a multi-language search engine. So, you can index and search the documents in English, Russian and even Chinese, all in one database.

HTTP, HTTPS, HTTP proxy, FTP (via proxy) protocols

As **ASPseek** is a Web search engine, it uses HTTP protocol to index sites. **ASPseek** also supports secure https:// protocol. FTP protocol is not supported directly, but you can use proxy (like squid) and index FTP sites via proxy.

ASPseek supports "basic authorization" feature of HTTP so you can index password-

protected areas (for example private information in your intranet).

Text/html and text/plain document types support

ASPseek can understand documents written in HTML, and plain text documents. These are the most popular formats in Internet.

Other formats, such as PDF, RTF, etc, can be supported with the help of any external program/script which is able to convert that formats to HTML or plain text.

Multithreaded design, async DNS resolver etc

ASPseek uses POSIX threads, that means that one process have many threads running in parallel. So index downloads documents from many sites, and search daemon processes many search queries simultaneously. This not only helps **ASPseek** to scale well on SMP (multiprocessor) systems, but also improves indexing speed, because in case of one thread most time will be spent on waiting for data from network.

One thing that slow indexing process down a lot is DNS lookup (a process of determining IP address using server name). To avoid delays, asynchronous lookups (lookup is done by separate dedicated processes) and IP address cache are implemented.

Stopwords

Stopwords are a words that have no meaning by itself. Examples: *is*, *are*, *at*, *this*. Searching for *at* is useless, so such words are excluded from search query. Stopwords are also excluded from database during indexing, so database becomes smaller and faster.

There is no "built-in" stopwords in **ASPseek**, they are loaded during start-up from files. Many stopword files for different languages comes with **ASPseek**.

Charset guesser

Some broken or misconfigured servers don't tell clients the charset in which they provide content. If you are indexing such servers, or using **ASPseek** to index ftp servers (FTP protocol does know nothing about charsets), charset guesser can be used to deal with it. Charset guesser uses word frequency tables (called langmaps) to determine correct charset.

Robot exclusion standard (robots.txt) support

ASPseek fully supports this standard. It is intended for web site authors for telling the robot (for example, **ASPseek**'s **index**(1)) to skip indexing some directories of their sites.

For more information see `http://www.robotstxt.org/wc/robots.html`

Settings to control network bandwidth usage and Web servers load

You can precisely control network bandwidth that **index**(1) uses. Exactly, you can limit the bandwidth (expressed in bytes per second) used by **index**(1) for given time-of-day. For example, you can limit the bandwidth during business hours so people at your office will not experience slow Internet.

You can also set the minimum time between two queries to the same Web server, so it will not be overloaded and got down to its knees while you run **index**(1).

Real-time asynchronous indexing

Some search engines requires that search should be stopped for the time of database update. **ASPseek** does not need it, so you can search non-stop.

More to say, there is a special mode of indexing called "real-time" indexing. You can use it for small number of documents, and as far as such document is downloaded and processed, changes are immediately visible in search interface. This feature is a great help if you are building search engine for pages with rapidly-changing content such as online news etc.

Note that number of documents in "real-time" database is limited. It's about 1000 on our hardware (your mileage may vary), and the more documents you have in "real-time"

database, the slower will be speed of indexing into that (and only that) database. This will not affect search speed though.

Documents from "real-time" database are moved to normal database after running **index**(1) in a normal way.

Sorting results by relevance or by date

Search engines usually returns most relevant results first. But if you are looking for latest pages, you can tell **ASPseek** to sort results by last modification date, so recently modified (or created) pages will be displayed first.

Excerpts, query words highlighting

Excerpt is a piece of found document with words searched for highlighted, just to give an idea of what the document is about. You can customize the number of excerpts displaying and their length. If you will disable excerpts, the beginning of document will be displayed.

Every found document is accompanied with the "Cached" link. **ASPseek** keeps a local compressed copy of every document processed, so user can see the the whole document with (optional) highlighted words that were searched for, even if it has been removed from original site (that happens sometimes).

Grouping results by site

Results from one site can be grouped together. If grouping by sites is on, only two results are displayed from the same site by default, and user can see other pages from the same site by following a "More results from ..." link.

Clones

Clones are identical documents at different locations. They are detected and grouped together, so user will not be presented with a page full of URLs to the identical documents.

Clone detection is usually limited by one site (so identical documents from different sites are not counted as clones), but you can change this by recompiling **ASPseek** with `--disable-clones-by-site` option.

Spaces and subsets

Space is the set of sites. So, if you want to provide the search narrowed to some area, you can create a space and search within that space. Only whole sites (e.g. *http://www.mysite.com/*) are allowed to be included in space.

Subsets can also be used to restrict the search. You can create subset and put URL mask (like *http://www.mysite.com/mydir/%*) into that, and then limit search scope to only given subset.

You can restrict search scope to not only one but several subsets or spaces.

HTML templates for easy-to-customize search results

You can customize your search pages, so they will look like and be seamlessly integrated with the rest of your site. This is done by simple editing of search template file.

SEE ALSO

index(1), **searchd**(1), **s.cgi**(1), **aspseek.conf**(5), **searchd.conf**(5), **s.htm**(5), **aspseek-sql**(5), **http://www.aspseek.org/**.

AUTHORS

This man page by Kir Kolyshkin <kir@asplinux.ru>

NAME

index − indexing program

SYNOPSIS

index [**-a**] [**-m**] [**-o**] [**-n** *num*] [**-q**] [**-N** *num*] [**-s** *status*] [**-t** *tag*] [**-u** *pattern*] [**-r** *file*] [**-g** *file*] [*configfile*]

index -i [**-u** *url* | **-f** *file*] [**-r** *file*] [**-g** *file*] [*configfile*]

index -T *url* [**-A** *num*] [**-r** *file*] [**-g** *file*] [*configfile*]

index -S [**-s** *status*] [**-t** *tag*] [**-u** *pattern*] [**-r** *file*] [*configfile*]

index -M [**-a**] [**-m**] [**-q**] [**-N** *num*] [**-r** *file*] [**-g** *file*] [*configfile*]

index -C [−w] [**-s** *status*] [**-t** *tag*] [**-u** *pattern*] [*configfile*]

index -E | **-D** | **-B** | **-K** | **-U** | [**-r** *file*] [**-g** *file*] [*configfile*]

index -X1 | **-X2** | **-H** [**-g** *file*] [*configfile*]

index -A *num* **-u** *pattern* [*configfile*]

index -P *URL* [*configfile*]

index -h | **-?**

DESCRIPTION

index is a component of **ASPseek** that performs Web crawling, documents downloading, parsing and storing. It can also be used to manipulate the **ASPseek** database.

During indexing process, **index** walks across the sites and stores found pages in a special data structures called delta files, and in SQL database.

When there is no more pages to index (or upon executing **index -D**), it sorts delta files and merges information from delta files into searchable database).

index supports HTTP and HTTP over SSL (https) protocols, and can parse documents in HTML and plain text formats. Support for other formats can be achieved via external converter programs.

The operation of **index** is mostly controlled by its configuration file **aspseek.conf**(5), which is read upon startup. You can give configuration file name as a last argument to **index**.

OPTIONS

Indexing options

-n *number*

Index only *number* of documents and exit. Note that you should run **index -D** manually after running **index -n**. Actual number of documents indexed can be a little higher than value requested if you use many threads.

-N *number*

Run *number* of **index** threads. It makes sense if you have many different sites to index, since no two threads are allowed to index the same site.

-R *number*

Run *number* of resolver processes. Default is (argument of **-N** option)/5 + 1. It makes sense to increase the default value if your name server is slow.

-i Insert new URLs to database. URLs to insert can be given using **-u** or **-f** options.

Re-indexing control

-a Re-index all documents regardless of their expiration status. Normally (without this option) only documents that have indexed earlier than **Period** time ago are re-

indexed.

-m Store words and hrefs found in documents regardless of their modification status. Normally (without this option) only those documents that have changed since last re-indexing are parsed.

-o Index documents with less hops first. Here "hops" means the "depth" value of the document.

-q Don’t add URLs from **Server** configuration command (and their corresponding **robots.txt** URLs) to database. This can be used if you haven’t changed your **aspseek.conf**(5) after last **index** run and is believed to speed up **index** startup in case you have several thousands **Server** entries in config.

-M Index URLs which were indexed by previous indexing session. These URLs are stored in **tmpurl** SQL table. Used mostly for debugging purposes.

Indexing to real-time database

-T *URL*

Index *URL* to real-time database, so it will be available for searching in seconds. Note that you can’t add too many documents to real-time database, otherwise the subsequent indexing to real-time database will be extremely slow. Actual limit of documents in real-time database is hardware dependent; well, about 1000 URLs should work OK. Documents from real-time database are merged to main database upon executing **index -D**.

This option is used to frequently re-index ever-changing pages (like first pages of news sites), or to re-index URL out-of-the-order (when you know it has just been changed) and see results immediately. Note that you can use **-A** option together with this one.

Clearing the database

-C Clear the database. You can use subsection control options (described below) to limit clearing to some part of the database. Note that clearing with limits may be quite slow on large database.

-w Used together with **-C** to disallow asking for confirmation before clearing.

Statistics

-S Print simple database statistics. You can use subsection control options (described below) together with this option.

Subsection control

In most cases you can combine any of **-u**, **-s** and **-t** options.

-s *status*

Limit index to documents matching *status* (HTTP Status code, or 0 for documents that were not yet indexed.

-t *tag* Limit index to documents matching *tag*. Tags can be set in **aspseek.conf**(5) file.

-u *pattern*

Limit index to documents with URLs matching *pattern* (supports SQL LIKE wild-card characters ’**%**’ and ’**_**’).

-f *file* Read URLs to be indexed/inserted/cleared from *file*. You can use - as file name, in that case URL list will be read from **stdin**.

Output

-r *file* Redirect output to *file*.

-g *file* Sets indexing statistics log file name to *file*. Default is `/usr/local/aspseek/var/`*DBName*`/logs.txt`.

Stopping index

-E Safely stop already running **index** process. Usable from scripts.

Database repairing

-X1 Check the inverted index for URLs for which **deleted** field in **urlword** SQL table is non-zero, or **status** field is not 200, or **origin** field is not 1.

-X2 Fix the above case by appending information about deleted keys to delta files. So, if you want to remove such records, run **index -X2**, **index -D** and finally perform SQL statements to delete unnecessary records.

-H Recreate citation indexes and ranks file from **urlwords***NN***.hrefs** fields in case of citation index corruption.

Database operations

-D Merge delta files into main database. This implies **-B**, **-K** and **-U**.

-B Generate subsets and spaces.

-K Calculate PageRanks.

-U Calculate total number of non-empty URL, which is saved to `/usr/local/aspseek/var/`*DBName*`/total` file).

Miscellaneous

-P *URL* Prints path to specified *URL*. Here path means the way by which index found that URL by outgoing links.

-A *space_id*

Add/delete a site to/from web space (use together with **-u** or **-A** options).

Getting help

-h, **-?** Print short help page.

FILES

`/usr/local/aspseek/etc/aspseek.conf`
`/usr/local/aspseek/var/`*DBName*`/logs.txt`

SEE ALSO

aspseek(7), **aspseek.conf**(5), **aspseek-sql**(5).

AUTHORS

Man page by Kir Kolyshkin <kir@asplinux.ru>

NAME

aspseek.conf − index configuration file

SYNOPSIS

/usr/local/aspseek/etc/aspseek.conf

DESCRIPTION

aspseek.conf is a configuration file for **index**(1). It completely defines all the aspects of ASPseek indexing process - what to index and how to do it.

The following can be defined:

General

DBAddr DBType:[//[**User**[:**Pass**]@]**Host**[:**Port**]]/**DBName**/

Defines SQL server connection parameters.
DBType is SQL server type, it can be *mysql* or *oracle8* for now.
User is a SQL server's user to connect as.
Pass is a **User**'s password. If this field is omitted, no password is used.
Host is a host name or IP address of host to connect to. If you are running SQL server on the same machine, use *localhost*.
Port is a port number on which SQL server is listening at. Default is the same as default port of used SQL server.
DBName is a name of the database used.

DBLibDir */some/dir*

Adds */some/dir* to list of directories to search for database backend library (lib*db-name-version*.so). Default library search path is `/usr/local/aspseek/lib`. Several such options can be used, each adding one more directory to the list. Last added directory is used first; compiled in path is last.

DataDir */some/dir*

Sets directory in which delta files and files with information about words, subsets, spaces will be stored. Default is `/usr/local/aspseek/var`.

DebugLevel *none* | *error* | *warning* | *info* | *debug*

Sets the level of debugging. If set to *none*, nothing will be logged. If set to *debug*, you will get a bunch of messages. Default value is *info*.

Include *file*

Includes the contents of *file* at this point, so you can specify some parameters in that included *file*. File name is relative to ASPseek etc directory (`/usr/local/aspseek/etc`).

Parameters that affects memory usage vs. performance

These parameters can be tuned to achieve the better performance on boxes with enough memory. They can also be used to reduce the amount of memory used by **index**(1).

DeltaBufferSize *kilobytes*

Size of buffer for each of 100 delta files, in kilobytes. Setting of low value for this parameter can result in big fragmentation of delta files. Value of this parameter affects used memory. If default value is used, then 50 Mb of memory is used for buffers. Default value is *512*.

UrlBufferSize *kilobytes*

Size of read and write buffer allocated during inverted index merging for **ind** files, in kilobytes. Value of this parameter affects used memory during inverted index merging. Default value is **DeltaBufferSize** * 8.

WordCacheSize *number*

Maximum word count in the word cache. Word cache is used to reduce database load for converting word to its word ID. Default value is *50000*.

HrefCacheSize *number*

Maximum URL count in the href cache. Href cache is used to reduce database load for converting URL of outgoing hyperlink to its URL ID. Default value is *10000*.

NextDocLimit *number*

Maximum number of URLs loaded from database at each request. Default value is *1000*.

This option is used only if URLs to be indexed are ordered by next index time; otherwise, if **-o** option to **index**(1) is used, all URLs for current hop is taken at once.

Database format parameters

These parameters set different modes of storing indexed information. Note that database format is different if you change these options, so the same value **must** be set in **searchd.conf**(5) file, and you **must not** change the values on a non-empty database.

HiByteFirst *yes* | *no*

Sets the byte ordering used in field **wordurl[1].word** (only in Unicode version). Default is *no*.

IncrementalCitations *yes* | *no*

Sets whether to build citation index, ranks of pages and lastmod incrementally. If value of this parameter is set to *yes*, then calculating of citations, ranks of pages and lastmod file will require less memory and take less time on large databases. So it is very handy if you want to index large number of URLs and have relatively small amount of memory. Default is *yes*.

CompactStorage *yes* | *no*

Sets the storage mode of reverse index. In compact storage mode, file/BLOB is not created for each word. Instead, information about all words is stored in 300 files. In this mode, updating of reverse index is generally much faster and requires a bit less memory than in the old mode. Default is *yes*.

UtfStorage *yes* | *no*

This parameter has sense only in Unicode version and only for MySQL back-end. In UTF8 storage mode fields **wordurl[1].word** are stored in UTF8 encoding. This mode can reduce sizes of data and index files for **wordurl** table. To convert existing Unicode database to this mode, run **index -b**. Default value is *no*.

Bandwidth control

MaxBandwidth *bytes* [*starttime* [*endtime*]]

Sets maximum used bandwidth for incoming traffic to *bytes* per second for the specified period of time of day. Arguments *starttime* and *endtime* are in seconds from midnight (0:00). If *endtime* is omitted, then it is implied to be the end of the day (86400). If both *starttime* and *endtime* are omitted, then the limit is for the whole day. You can use several **MaxBandwidth** commands. Note that if *endtime* is less than *starttime*, **index**(1) will handle it correctly, setting two intervals from *starttime* to midnight and from midnight to *endtime*. By default bandwidth is not limited.

Indexing

Server *URL*

Add *URL* as an URL to start indexing from. You can specify many **Server** commands, and set the different options for different sites - see below. Note that if *URL*

contains path, the whole site will be indexed nevertheless, so to limit indexing to some subdirectory of site use **Disallow** parameter described below.

Global indexing parameters

Each of the below parameters can be specified only once in configuration file and takes the global effect for the whole **index**(1) session.

MaxDocSize *bytes*

Sets the maximum document size in bytes, so if document size is bigger than *bytes*, only the first *bytes* of the document will be processed. Default value is *1048576* bytes (1Mb).

HTTPHeader *header*

Add *header* to headers that **index**(1) sends in HTTP request. You should not use *If-Modified-Since* or *Accept-Charset* headers here, as **index**(1) sends it anyway. Header *User-Agent: aspseek/1.2.10* is sent too, but you may override it.

Clones *yes* | *no*

Sets whether to enable clones eliminating. Clone is a document which is absolutely the same as another document. If this set to *yes*, clone is not parsed/stored in the database, instead word information for original document is used. Default value is *yes*.

MinWordLength *number*

Sets the minimum length of word to be stored in the database, so words shorter than *number* is not stored. Default value is *1*.

MaxWordLength *number*

Sets the maximum length of word to be stored in the database, so words longer than *number* is not stored. Default value is *32*. Note that you can't set the value higher than 32.

DeleteNoServer *yes* | *no*

Sets whether to delete URLs which have no correspondent "Server" commands. Default value is *yes*.

AddressExpiry *time*

Sets expiration time for "DNS name -> IP" entry in address cache. After entry is expired, resolver will make DNS lookup again. Argument *time* can be set in seconds, or the same way as in **Period** command below. Default value is 1 hour.

Indexing scope

These parameters can is used to limit the scope of indexing. **index**(1) will compare all URLs against all **CheckOnly**, **CheckOnlyNoMatch**, **Allow**, **AllowNoMatch**, **Disallow** and **DisallowNoMatch** directives in the order specified in configuration file, so order is important. Note that by default everything is allowed.

Some directives below use POSIX regular expressions (regexp) for flexibility. For description of what regexp is, see **regex**(7), **grep**(1), **awk**(1).

FollowOutside *yes* | *no*

Sets whether **index**(1) should index outside sites defined in **Server** directives. Default is *no*. If you set it to *yes*, be sure to limit the scope of indexing in some other way (for example, with **MaxHops**).

CheckOnly *regexp* [*regexp*...]

Use **HEAD** request instead of **GET** for URLs matching *regexp*. So, such URLs will not be downloaded, just information about them will be stored in **urlword** table.

CheckOnlyNoMatch *regexp* [*regexp*...]
Use **HEAD** request instead of **GET** for URLs **not** matching *regexp*. So, such URLs will not be downloaded, just information about them will be stored in **urlword** table.

Allow *regexp* [*regexp*...]
Allows to index URLs matching *regexp*.

AllowNoMatch *regexp* [*regexp*...]
Allows to index URLs **not** matching *regexp*.

Disallow *regexp* [*regexp*...]

DisallowNoMatch *regexp* [*regexp*...]
Disallows to index URLs **not** matching *regexp*.

Countries *file*
Loads countries IP information from *file*. File consists of lines in the form "sss.sss.sss.sss - eee.eee.eee.eee cc", where *sss.sss.sss.sss* is starting IP address, *eee.eee.eee.eee* is ending IP address, and *cc* is a country code (like *ru*, *de*, etc.). Note that value of ending address should be more than starting address.

AllowCountries *cc1* [*cc2*...]
Specifies to index only sites from countries specified by *cc1*, *cc2*, etc. Should be used together with the **Countries**.

Indexing parameters - local

Each of the below parameters can be specified many times in configuration file, applies to all **Server** parameters below it, and valid till next parameter with the same name, or till the end of configuration file.

Period *time*
Sets the re-index period to *time*. Value can be set just in seconds, or using a special characters right after the number (no spaces allowed): **s** for seconds, **M** for minutes, **h** for hours, **d** for days, m for months and **y** for years. You can combine several values together, for example string *1m12d* means "one month and twelve days". You can also specify negative numbers, say *1m-10d* stands for "one month minus ten days". Default value is *7d*.

Tag *number*
Use this field to "tag" several **Server**s with value *number*, which can later be used with option **-t** *number* of **index**(1) command. Note that if you want to group several sites together for searching purposes, you should use "spaces" or "subsets" features of ASPseek, not tag.

MaxHops number
Sets the maximum hops ("mouse clicks") from URL specified by **Server** command, so documents that are "deeper" will not be indexed. Default value is *256*.

IncrementHopsOnRedirect *yes | no*
Sets whether **index**(1) should increment hops value when HTTP redirect is encountered. Applies only to redirects generated by "**Location:**" HTTP headers. Setting this option to *no* allows a greater number of documents to be indexed for sites that redirect frequently (e.g. for cookie testing, typically on each page). Default value is *yes*.

RedirectLoopLimit *number*
Allow no more than *number* of contiguous redirects. This option is especially useful if you set **IncrementHopsOnRedirect** to *no*, because **index**(1) can fall in an endless redirect loop. Limiting the number of redirects prevents **index** from such redirect loops. Default value is *8*.

MaxDocsPerServer *number*

Sets that no more than *number* of documents will be indexed from one site during one run of **index**(1). Default value is *-1*, which means no limits.

MaxDocsAtOnce *number*

Sets the maximum number of pages to be downloaded from the same host before switching to the next host. Large values are believed to increase indexing performance when number of indexed sites is large. Default value is *1*.

ReadTimeOut *time*

Sets the maximum timeout to *time* for downloading a document from site. Argument can be expressed in seconds, or in the same form as in **Period** command above. Default value is 90 seconds.

Robots *yes* | *no*

Sets whether the robot exclusion standard (`robots.txt` file and META NAME="robots") will be honored. Default is *yes*.

DeleteBad *yes* | *no*

Sets whether to delete bad (not found, forbidden etc.) URLs from the database. Default value is *no*.

Index *yes* | *no*

Sets whether to store words into database. Default value is *yes*.

Follow *yes* | *no*

Sets whether to store links found into database. Default value is *yes*.

Charset *charset*

Usable to set charset for the servers that do not return it. Argument should be known charset name (see below for charset configuration). Alternatively, you can use charset guesser feature of **index**(1).

Replace [*regexp* [*replacement*]]

This parameter allows to replace URL matching *regexp* by *replacement*, or by empty string if *replacement* is not specified. This is useful for sites with dynamic contents where the same information can be obtained by many different URLs. **Replace** without arguments disables any replacements for subsequent **Server** commands.

As in **sed**(1) command **s**, the *replacement* can contain *N* (*N* being a number from 1 to 9, inclusive) references, which refer to the portion of the match which is contained between *N*th '\(' and its matching '\)'. To include a literal '\', precede it with another '\'.

MinDelay *time*

Sets minimum time between finishing of access to server and beginning of next access to the server. This is useful if site owner blames you for "bombing" his site with your **index**(1) queries. Argument *time* can be set in seconds, or in the same way as described in **Period** command above. Default value is *0*.

Proxy [*host.com*[:*port*]]

Use proxy rather than direct connection. You can also index FTP sites via proxy. If *port* is not specified, default is *3128* (sqiud). **Proxy** without arguments disables proxy.

External converters

index(1) has an ability to deal with document types other than **text/plain** and **text/html**. It does so with the help of an external programs or scripts, which can convert from some format

to **text/plain** (or **text/html**), so you are able to index `.ps`, `.pdf` etc.

Converter *from/type to/type*[; **charset**=*cset*] *command line*

Specifies that for converting documents with MIME-type *from/type* to MIME-type *to/type* the command specified by *command line* will be used. Argument *from/type* can be any type returned by Web server. Argument *to/type* can be either *text/plain* or *text/html*. If you add **;charset**=*cset* string after *to/type*, **index** will know that resulting document has a charset *cset*, otherwise it is assumed to be **us-ascii**.

In the *command line* you usually specify program or script to run, together with its options. Program is expected to to read from stdin and write the converted document to stdout.

If your program can't deal with stdin/stdout streams, you should use **$in** and **$out** strings in **command line**, and they will be substituted with two file names in `/tmp` directory. **index**(1) will create files with unique names, write the document downloaded to the first file (referenced as **$in**), run the */bin/prog*, read the second file (referenced as **$out**) into memory, and then delete both files.

You can also use **$url** in *command line*, it will be substituted with the actual URL of downloaded document. You can use it in your own scripts to distinguish between a different document variations, or to be able to write one script for many different MIME-types.

Please note that **index**(1) relies on a **Content-Type** header returned by a Web server. Some Web-servers are misconfigured and give wrong info (for example, return header **Content-Type: audio/x-pn-realaudio-plugin** for `.rpm` files).

Examples:

```
Converter app/ps text/plain; charset=iso8859-1 ps2ascii
# ps2ascii can't deal with PDF files from stdin
Converter application/pdf text/plain ps2ascii $in $out
```

Charset configuration for non-Unicode version

Charset configuration for non-Unicode version is usually stored in file `/usr/local/aspseek/etc/charsets.conf`. Charset files for non-Unicode version can be found in `/usr/local/aspseek/etc/charsets` directory. Langmap files can be found in `/usr/local/aspseek/etc/langmap` directory.

CharsetTable *charset lang file* [*lmfile*]

Loads the table for *charset* of language *lang* from *file*. Optionally load langmap file *lmfile*, which is used for charset guesser.

CharsetAlias *charset alias1* [*alias2...*]

Defines *alias1*, *alias2*, ... as aliases (alternative names) for *charset*. This is needed because in many cases there is no "one true name" for the charset - different web servers and page authors use different names.

LocalCharset *charset*

Sets the local charset for ASPseek, so all data in the database is assumed to be in that charset.

Charset configuration for Unicode version

Charset configuration for Unicode version is usually stored in file `/usr/local/aspseek/etc/ucharset.conf`. Charset files for Unicode version can be found in `/usr/local/aspseek/etc/tables` directory.

CharsetTableU1 *charset lang file* [*lmfile*]

Loads the Unicode mapping for *charset* of language *lang* from *file*. Optionally load langmap file *lmfile*, which is used for charset guesser.

CharsetTableU2 *charset lang file* [*lmfile*]

Loads the Unicode mapping for multibyte *charset* of language *lang* from *file*. Optionally load langmap file *lmfile*, which is used for charset guesser.

Dictionary2 *lang file* [*charset*]

Loads dictionary for *lang* from *file*. If *charset* is not specified, it is assumed that the file is in Unicode. Dictionary is used for tokenizing of text in Chinese, Japanese and Korean languages.

Stopwords

Stopwords configuration is usually stored in the file `/usr/local/aspseek/etc/stopwords.conf`. Stopword files for different languages can be found in `/usr/local/aspseek/etc/stopwords` directory.

StopwordFile *lang file* [*charset*]

Loads stopwords for language *lang* from *file*. If *charset* is not specified, file contents is assumed to be in **LocalCharset**, otherwize it is in *charset*.

FILES

```
/usr/local/aspseek/etc/aspseek.conf
/usr/local/aspseek/etc/charsets.conf
/usr/local/aspseek/etc/ucharset.conf
/usr/local/aspseek/etc/stopwords.conf
```

BUGS

Many parameters are the same in **searchd.conf** and in **aspseek.conf**(5).

SEE ALSO

index(1), **aspseek.conf**(5), **regex**(7), **http://www.robotstxt.org/wc/robots.html**.

AUTHORS

Man page by Kir Kolyshkin <kir@asplinux.ru>

NAME

searchd − search daemon

SYNOPSIS

searchd -D [**-R**] [**-l** *logfile*] [*configfile*]

searchd -h

DESCRIPTION

searchd is a search daemon. Its purpose in ASPseek is to search the database created by **index**(1), cache search results etc. It listens on a port for **s.cgi**(1) queries, does the search and returns results found to **s.cgi**(1).

searchd loads some data from the database to memory to speed up search. If that data is changed on disk, **searchd** reloads it.

OPTIONS

-D is used to run **searchd** as a daemon. You need it every time you run **searchd**.

-R is used to auto-restart **searchd** in case of its failure. So using this option you will get non-stop search engine.

-l *logfile* sets that **searchd** should write its log into specified *logfile*. By default log is written into `/usr/local/aspseek/etc/dlog.log`.

-h is used to get a help message from **searchd**.

configfile is the name of configuration file to be read. Default is `/usr/local/aspseek/etc/searchd.conf`.

FILES

`/usr/local/aspseek/etc/searchd.conf`
`/usr/local/aspseek/var/dlog.log`

BUGS

If you try to start **searchd**, but the port used by it is already in use by another program (most probably another **searchd**), you will not get error message to console, only to log file. So every time before trying to start **searchd** check that there's no **searchd** running, and kill it if necessary. Or, after starting **searchd**, check the log file for errors.

SEE ALSO

index(1), **s.cgi**(1), **searchd.conf**(5), **aspseek-sql**(5).

AUTHORS

Man page by Kir Kolyshkin <kir@asplinux.ru>

NAME

searchd.conf − searchd configuration file

SYNOPSIS

/usr/local/aspseek/etc/searchd.conf

DESCRIPTION

searchd.conf is a configuration file for **searchd**(1). The following parameters can be defined:

General

DBAddr DBType:[//[**User**[:**Pass**]@]**Host**[:**Port**]]/**DBName**/

Defines SQL server connection parameters.
DBType is SQL server type, it can be *mysql* or *oracle8* for now.
User is a SQL server's user to connect as.
Pass is a **User**'s password. If this field is omitted, no password is used.
Host is a host name or IP address of host to connect to. If you are running SQL server on the same machine, use *localhost*.
Port is a port number on which database is listening for SQL queries. Default is the same as default port of used SQL server.
DBName is a name of the database used.

Port *nnn*

Sets the port number on which **searchd**(1) is listens to **s.cgi**(1) queries. Default is *12345*.

DBLibDir */some/dir*

Adds */some/dir* to list of directories to search for database backend library (lib*db-name-version*.so). Default library search path is `/usr/local/aspseek/lib`. Several such options can be used, each adding one more directory to the list. Last added directory is used first; compiled in path is last.

AllowFrom *some.host.com* | *xxx.xxx.xxx.xxx*[/*yy*]

This implements access control list, so **searchd**(1) will only accept connections from host(s) specified. Several such options can be used. You can specify hostname, IP address, or subnet (IP address with mask in CIDR notation).

DataDir */some/dir*

Sets directory in which delta files and files with information about words, subsets, spaces will be stored. Default is `/usr/local/aspseek/var`.

DebugLevel *none* | *error* | *warning* | *info* | *debug*

Sets the level of debugging. If set to *none*, nothing will be logged. If set to *debug*, you will get a bunch of messages. Default value is *info*.

MinFixedPatternLength *nnn*

Sets the minimal length of fixed part of word with pattern (like someth*) to be allowed in search query. Words shorter than this value will be rejected with appropriate error. Setting this to less than 3 will open ASPseek to DoS attacks. Default value is *6*.

MaxThreads *nnn*

Sets the maximum number of threads that search daemon can run simultaneously to process queries. Setting high value can result in big memory consumption. Setting low value can result in big response time for queries in high load conditions (as "extra" queries are queued). Default value is *10*.

MultipleDBConnections *yes* | *no*

Sets whether to use separate connection to the database for each thread. If multiple connections are used, this leads to better concurrency between threads, especially when one or more threads perform pattern search and the other is trying to perform simple search. Default is *yes*.

Include *file*

Includes the contents of *file* at this point, so you can specify some parameters in that included *file*. File name is relative to ASPseek etc directory (`/usr/local/aspseek/etc`).

Database format parameters

These parameters tells **searchd**(1) what database format is used by **index**(1), so their values should be set to the same values as in **aspseek.conf** file.

HiByteFirst *yes* | *no*

Sets the byte ordering used in field **wordurl[1].word** (only in Unicode version). Default is *no*.

IncrementalCitations *yes* | *no*

Sets whether the data produced by **index**(1) is in "incremental citations" format. Default is *yes*.

CompactStorage *yes* | *no*

Sets the index storage mode. Default is *yes*.

UtfStorage *yes*|*no*

This parameter has sense only in Unicode version and only for MySQL back-end. In UTF8 storage mode fields **wordurl[1].word** are stored in UTF8 encoding. This mode can reduce sizes of data and index files for **wordurl** table. To convert existing Unicode database to this mode, run **index -b**. Default value is *no*.

Ispell support parameters

When **ASPseek** is used with ispell support, **searchd**(1) can optionally find all forms for all specified words (example: 'create' -> 'create' OR 'created' OR 'creates'). This scheme retains exact search possibility. Note that only ispell suffixes are supported by now; prefixes are usually change the word meanings, for example if somebody searches for the word **tested** he hardly wants *untested* to be found.

Ispell affixes file contains rules for words and has the following format:

```
flag V:
 E     >  -E,IVE     # As in create > creative
 [^E]  >  IVE        # As in prevent > preventive

flag *N:
 E     >  -E,ION     # As in create > creation
 Y     >  -Y,ICATION # As in multiply > multiplication
 [^EY] >  EN         # As in fall > fallen
```

Ispell dictionary file contains words themselves and has format like this:

```
wop/S
word/DGJMS
wordage/S
wordbook
wordily
```

```
wordless/P
```

Note that if you add ispell support to already existing database, re-indexing is not required.

You may also use ispell flags in this file if you know how to do it. This will allow not to write the same word with different endings to the rare words file, for example "webmaster" and "webmasters". You may choose the word which have the same changing rules from existing ispell dictionary and just to copy flags from it. For example, English dictionary has this line:

```
postmaster/MS
```

So, webmaster with MS flags will be probably OK:

```
webmaster/MS
```

You can get ispell affix and dictionary files for different languages from `http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell-dictionaries.html`

To make **ASPseek** support ispell the following parameters are used. *lang* argument is two letters language abbreviation. File names used are relative to ASPseek etc directory (`/usr/local/aspseek/etc`). Absolute paths can be also specified.

Affix *lang affix-file* [*charset*]
: Load ispell affixes for language *lang* from file *affix-file*. If *charset* is given, file contents is assumed to be in that charset, otherwise the value from **LocalCharset** is used.

Spell *lang dict-file* [*charset*]
: Load ispell dictionary for language *lang* from file *dict-file*. If *charset* is given, file contents is assumed to be in that charset, otherwise the value from **LocalCharset** is used.

WordForms *on* | *off* | *lang*[,*lang*[,...]]
: Sets whether to search for different word forms by default. Argument can be *on*, *off*, or comma-separated list of languages. Value can be overridden by **fm** parameter of **s.cgi**(1).

Ranking parameters

SiteWeight *http://www.site.com nnn*
: Specifies the priority for particular site. Default priority for all sites is 0. If priority of site is greater than 0, hen it will always be displayed before all the other results. If priority of site is less than 0, then it will always be displayed after all the other results.

AccountDistance *on* | *off*
: Specifies whether **searchd**(1) should that into account distance from the beginning of document section to search terms for ranking calculations. If this parameter is on, then documents with search terms closer the beginning of section have higher priority over others, otherwise distance doesn't matter. Default is *on*. Value can be overridden by **ad** parameter of **s.cgi**(1).

Results cache parameters

searchd can implement results cache, so results for next page queries and for queries that are the same as were before will be taken from cache. The following parameters are used.

Cache *on*

If this line is present, results cache will be enabled. By default cache is disabled.

CacheLocalSize *number*

Size of cache, in entries (one entry for one query). Default value is *100*.

CachedUrls *number*

Number of resulting URLs to be stored in one cache entry. Default value is *200*.

Charset configuration for non-Unicode version

Charset configuration for non-Unicode version is usually stored in file `/usr/local/aspseek/etc/charsets.conf`. Charset files for non-Unicode version can be found in `/usr/local/aspseek/etc/charsets` directory. Langmap files can be found in `/usr/local/aspseek/etc/langmap` directory.

CharsetTable *charset lang file* [*lmfile*]

Loads the table for *charset* of language *lang* from *file*. Optionally load langmap file *lmfile*, which is used for charset guesser.

CharsetAlias *charset alias1* [*alias2*...]

Defines *alias1*, *alias2*, ... as aliases (alternative names) for *charset*. This is needed because in many cases there is no "one true name" for the charset - different web servers and page authors use different names.

LocalCharset *charset*

Sets the local charset for ASPseek, so all data in the database is assumed to be in that charset.

Charset configuration for Unicode version

Charset configuration for Unicode version is usually stored in file `/usr/local/aspseek/etc/ucharset.conf`. Charset files for Unicode version can be found in `/usr/local/aspseek/etc/tables` directory.

CharsetTableU1 *charset lang file* [*lmfile*]

Loads the Unicode mapping for *charset* of language *lang* from *file*. Optionally load langmap file *lmfile*, which is used for charset guesser.

CharsetTableU2 *charset lang file* [*lmfile*]

Loads the Unicode mapping for multibyte *charset* of language *lang* from *file*. Optionally load langmap file *lmfile*, which is used for charset guesser.

Dictionary2 *lang file* [*charset*]

Loads dictionary for *lang* from *file*. If *charset* is not specified, it is assumed that the file is in Unicode. Dictionary is used for tokenizing of text in Chinese, Japanese and Korean languages.

Stopwords

Stopwords configuration is usually stored in file `/usr/local/aspseek/etc/stopwords.conf`. Stopword files for different languages can be found in `/usr/local/aspseek/etc/stopwords` directory.

StopwordFile *lang file* [*charset*]

Loads stopwords for language *lang* from *file*. If *charset* is not specified, file contents is assumed to be in **LocalCharset**, otherwise it is in *charset*.

FILES

```
/usr/local/aspseek/etc/searchd.conf
/usr/local/aspseek/etc/charsets.conf
/usr/local/aspseek/etc/ucharset.conf
/usr/local/aspseek/etc/stopwords.conf
```

BUGS

Many parameters are the same in **searchd.conf** and in **aspseek.conf**(5).

SEE ALSO

searchd(1), **aspseek.conf**(5).

AUTHORS

Man page by Kir Kolyshkin <kir@asplinux.ru>

NAME

s.cgi − aspseek search CGI front-end

SYNOPSIS

http://your.host.com/cgi-bin/s.cgi[?**parameter=***value*[&**parameter**=*value*...]]

DESCRIPTION

s.cgi is a search front-end of aspseek. It is a CGI program, so usually it is run by web server. **s.cgi** parses its options, reads its configuration and template from **s.htm**(5) file residing in /usr/local/aspseek/etc directory, makes a query to **searchd**(1), formats the results according to template, and outputs HTML page containing results found, or error, or empty search form in case no query was specified.

Functionality of **s.cgi** related to template and meta symbols is described in **s.htm**(5).

OPTIONS

q=*query* Query string. Input line in search form, used for query entering, should be named "**q**".

ul=*sql_pattern* [*sql_pattern* ...]

ul=*site* [*site* ...]

URL mask. Several masks can be used, delimited by exactly 1 space. This parameter can also be used multiple times, values are concatenated. *sql_pattern* specifies subdirectory of site, and record in **subsets** table containing value of *sql_pattern* in the field **mask** must exist. *site* specifies the whole site, it should be in the form *http://www.sitename.com/*. Note that *sql_pattern*s and *site*s can not be mixed together. Examples:
ul=*http://www.aspstreet.com/*
ul=*http://www.aspstreet.com/directory/%*

r=*query* Previous query string. Used for searching in results, when **in** parameters is equal to *on*.

s=*rate* | *date*

Result sorting type. *rate* means sorting by rate (relevance), and *date* means sorting by last modification date. Sorting order is descending, so results with greater rate or latter modification date will be first ones.

in=*on* Used for "search within results" feature. If value of this parameter is *on*, then actual query is composed from parameters **q** and **r** combined by AND. Any other value is ignored. Checkbox "Search within results" in query form, that is displayed in search results, should be named **in**. Also, the following string should be put in query form:

```
<INPUT TYPE="hidden" NAME="r" VALUE="$q">
```

t=*query* Used for "Take me there" feature. If this parameter is present, then **s.cgi** finds first results matching query and generates output for redirection to that URL.

ch=*url* "Cached" URL. This parameter is used for displaying "cached" copy of web page. Can be combined with **q**, **cs** and **tmpl** parameters. If **q** and **cs** parameters are present, then appropriate query terms will appear highlighted in "cached" page.

np=*number*

Result page number. Default value is *0*.

ps=*number*

Number of results per page. Overrides value set by "preferences" cookie.

gr=*off* Grouping results by site. If value of this parameter is *off*, then results are not grouped by site. Any other value is ignored.

st=*number* Site ID. Value of this parameter is used to restrict search by specified site. Generated in search results, as result of $SH meta symbol, which is used in **moreurls** template.

tmpl=*file* Template file name. *file* is relative to `/usr/local/aspseek/etc` directory. Default is constructed from the part of **s.cgi** name before point with *.htm* added, so **s.cgi** will look for **s.htm**, and if you rename it to **another.cgi**, it will look for **another.htm**.

sp*N*=*on* Space ID. *N* is one or more digits value. If value of this parameter is *on*, then search is restricted by web space with ID equal to *N*. Any other value is ignored. This parameter can be used multiple times, all spaces are being ORed.
To assign particular site to web space, insert record with appropriate values or site ID and space ID into table **spaces**.

cs=*charset* Source charset. Tells **s.cgi** which *charset* is used in input query. This is required parameter if non-ascii characters are used in query. Results of query will also be presented in that *charset*.

fm=*on* | *off* | *lang*[,*lang*,..]
Word forms. Can be comma-separated list of languages or just *on* or *off*. In case it is not set to *off*, **s.cgi** will search for all forms of specified words, and results with exact word forms will be displayed first.
Example: if word 'create' is specified, then documents containing either 'create' or 'creates' or 'created' will be found.

ad=*on* Account distance. If value is *on*, then **s.cgi** ranks documents with search terms closer to the beginning higher. Any other value is ignored. Overrides **AccountDistance** parameter in **searchd.conf**(5).

ln=*url* Link to page. Tells to **s.cgi** to find pages which have link to the specified *url*. If protocol in *url* is omitted, then http:// is implied. If *url* is not found and last symbol is not '/', than **s.cgi** tries to find URL with '/' at the end.

Limiting search scope to certain parts of HTML pages

Parameters **bd**, **ds**, **kw** and **tl** can be used together.

bd=*on* Search in body. If value of this parameter is **on**, then advanced search will be performed only within HTML <BODY>...</BODY>. Any other value is ignored.

ds=*on* Search in description. If value of this parameter is **on**, then advanced search will be performed only within HTML <META NAME="DESCRIPTION"...>. Any other value is ignored.

kw=*on* Search in keywords. If value of this parameter is **on**, then advanced search will be performed only within HTML <META NAME="KEYWORDS"...>. Any other value is ignored.

tl=*on* Search in title. If value of this parameter is **on**, then advanced search will be performed only within HTML <TITLE>...</TITLE>. Any other value is ignored.

Limiting search scope to certain period of dates

dt=*back* | *er* | *range*
Type of time limit. See below.

dp=*date* If 'dt' is 'back', that means you want to limit result to recent pages, so you specify that "recentness" in *date* value, that is specified in the form *xxxA*[*yyyB*...],

there *xxx*, *yyy* are numbers (which can also be negative), and *A*, *B* can be one of the following (the letters below are the same as in **strptime**(3) and **strftime**(3) functions):

Character	Meaning
s	second
M	minute
h	hour
d	day
m	month
y	year

Examples of values for **dp** parameter:

String	Meaning	Value, s
4h30m	2 hours and 30 minutes	16200
1Y6m-15d	1 year and six month minus 15 days	45792000
1h-60M+1s	1 hour minus 60 minutes plus 1 second	1

Note that **ASPseek** do not use minutes and seconds of document's last modification date, so specifying something more precise than hour is useless (but still allowed).

dx=*1* | *-1* If **dt** is *er* (which is short for newer/older), that means the search will be limited to pages newer or older than date given. Parameter **dx** is newer/older flag, value *1* means "newer" or "after", and value *-1* means "older" or "before". The actual date is separated into **dd**, **dm**, **dy** fields as follows.

dd=*number*

Day of month (1...31)

dm=*number*

Month (0 - January, 1 - February, .., 11 - December)

dy=*number*

Year (four digits, for example 2001)

db=*dd/mm/yyyy*

de=*dd/mm/yyyy*

If **dt** is *range*, that means search within given range of dates. Parameters **db** and **de** are used in this case and stands for beginning and ending date, respectively. Each date is in the form *dd/mm/yyyy*, there *dd* is day of month number (1...31), *mm* is month number (1...12), and *yyyy* is four-digits year.

fr=*value*

to=*value* These parameters are passed to **s.cgi** in subsequent search pages; they contains date and time in internal format used by **s.cgi**.

Advanced search

iq=*words* Include *words*. Used in advanced search. If this parameter is not empty, then parameter **q** is not used.

xq=*words* Exclude *words*. Used in advanced search. All words found in value of this parameter will be added to query with - sign before them.

im=*p* Include mode. If value of this parameter is equal to *p*, then value of parameter **iq** is double-quoted, which means pages containing phrase must be found.

xm=*p* Exclude mode. If value of this parameter is equal to *p*, then value of parameter **xq** is double-quoted, which means pages containing phrase must be excluded.

is=*site_pattern* [*site_pattern* ...]
Include sites. This parameter is used to restrict search by sites those names contains *site_pattern*. If several patterns delimited by space are used, then they are ORed. Examples: **is**=*www.google*, **is**=*.com*.

xs=*site_pattern* [*site_pattern* ...]
Exclude sites. This parameter is used to to exclude sites those names contains *site_pattern* from results. If several patterns delimited by space are used, then they are ORed.

o=*number* Use "*number*+1"th template sections if possible. See "Defining different output formats" subsection in **s.htm**(5).

ENVIRONMENT

QUERY_STRING
This variable is usually set by web server and contains all the parameters for **s.cgi** described in OPTIONS above.

ASPSEEK_TEMPLATE

UDMSEARCH_TEMPLATE
Template file to use. Overridden by **tmpl** parameter. If both variables are used, **ASPSEEK_TEMPLATE** is preferable.

HTTP_COOKIE
Cookies can be sent by client browser together with request. Web server forms **HTTP_COOKIE** environment variable from it. **s.cgi** parses it and uses value of **PS** section as the number of results per page. This is overridden by **ps** parameter.

HTTP_HOST
Value is set by web server and is used together with self name to form HREF references to other result pages.

SCRIPT_NAME
Value is set by web server and is used to determine the self name of the script and the name of template file to load (see description of **tmpl** parameter).

REDIRECT_STATUS

REDIRECT_URL

PATH_TRANSLATED
These variables can possibly be set by web server (for example, by Apache using its "AddHandler" and "Action" directives). If **REDIRECT_STATUS** is set, **s.cgi** will get the self name from **REDIRECT_URL**, and template file name from **PATH_TRANSLATED**.

FILES

`/usr/local/aspseek/etc/s.htm`

BUGS

It appears that **tmpl** parameter is broken since version 1.2.8. Use either **ASPSEEK_TEMPLATE** environment variable, or rename **s.cgi**.

NOTES

When parameters are passed in URL, they should be encoded according to RFC 1738. If values are coming from form input fields, they are encoded by browser (so, for example, space character becomes either '+' or '%20'). **s.cgi** does all appropriate decoding of parameter values and encoding of links to the other pages it generates whenever needed. In this page values are shown in non-encoded form.

SEE ALSO

s.htm(5), **searchd**(1), **http://www.aspseek.org/.**

AUTHORS

Man page by Kir Kolyshkin <kir@asplinux.ru> and Alexander F. Avdonkin <al@asplinux.ru>.

NAME

s.htm − template and configuration for s.cgi

SYNOPSIS

`/usr/local/aspseek/etc/s.htm`

DESCRIPTION

s.htm acts both as a configuration file and as a HTML template for **s.cgi**(1). It looks like HTML file, and is divided into sections. **s.cgi**(1) takes some sections, makes meta variables substitution in them, and outputs the resulting HTML.

Section is a piece of **s.htm** file that is started with line `<!--sectionname-->` and ends with line `<!--/sectionname-->`. Section consists of HTML formatted text containing meta variables. Meta variable is one or two letters preceded by $ sign (example: $DX). Meta variable names are case-sensitive. **s.cgi**(1) substitutes meta variables with corresponding string.

Special “variables” section

This is a special section in which you define various configuration parameters in the form "**Parameter** *value*". Unlike all the other sections, it starts with the string `<!--variables` and ends with `-->`. The contents of this section is never included in **s.cgi**(1) output.

The following parameters are defined here.

DaemonAddress *xxx.xxx.xxx.xxx*[:*nnn*]

Defines the IP address (*xxx.xxx.xxx.xxx*) and port (*nnn*) of search daemon **searchd**(1). If port is not set, ASPseek’s default port is used.

You *should* put at least one **DaemonAddress**. You *can* put several **DaemonAddress** lines; in this case **s.cgi**(1) will connect to all **searchd**(1) daemons, retrieve results from all of it, and show merged results.

Note that you can’t give DNS names (like **www.myhost.com**) as an argument to **DaemonAddress**, only numeric IP addresses are allowed here. This is done because name lookup needed in case of DNS name will slow down **s.cgi**(1) considerably.

MaxExcerpts *num*

Defines the maximum number of excerpts that are shown in results.

MaxExcerptLen *num*

Defines the maximum length (in characters) of each excerpt string.

PagesPerScreen *num*

Defines the maximum number of links to other search result pages to be shown if there are many results found.

ResultsPerPage *num*

Defines how many results will be shown on one page. Note that this value is overwritten by **PS** cookie if set, and **ps** parameter to **s.cgi**(1). Less value makes search results page to appear faster, so don’t set it too high.

Clones *no*

If this line is present, clones detecting and showing is disabled.

You can also redefine error messages that are used in cases **s.cgi**(1) encounters an error. The following error messages are used.

ER_STOPWORDS *string*

Displayed if only stopwords are used in query.

ER_EXTRASYMBOL *string*
Displayed if some extra symbols were found at the end of query.

ER_EMPTYQUERY *string*
Displayed if the query is empty.

ER_TOOSHORT *string*
Displayed if the pattern is used, and not enough characters are provided in it. See also the description of **MinFixedPatternLength** parameter in **searchd.conf**(5).

ER_NOQUOTED *string*
Displayed if unmatched quote was found.

ER_NOPARENTHESIS *string*
Displayed if unmatched parenthesis was found.

ER_NOSEARCHD *string*
Displayed if **s.cgi**(1) was unable to connect to **searchd**(1).

META VARIABLES

Meta variables described here are **not** position-dependent, that is to say they can be used in any template sections. Variables that can be used only in some sections described in the **TEMPLATE SECTIONS** below.

$$ This is actually not a meta variable, but a way to include $ sign to **s.cgi**(1) output. So, if you want to put a *$* in template, write it as *$$*.

$AV ASPseek version (like 1.2.10).

$P String "*&tmpl=*" concatenated with value of **tmpl** parameter of **s.cgi**(1), URL-escaped, or empty string if there was no **tmpl** parameter given.

$c Current value of **cs** (charset) parameter of **s.cgi**(1). Works only in Unicode version.

$f Number of first document displayed on this page.

$fm Current value of **fm** (word forms) parameter of **s.cgi**(1).

$ps Current value of **ps** (results per page) parameter of **s.cgi**(1).

$s Current value of **s** (sorting type) parameter of **s.cgi**(1). Can be either *rate* or *date*.

$np Current value of **np** (page number) parameter of **s.cgi**(1).

$gr Current value of **gr** (grouping by site) parameter of **s.cgi**(1). Can be either *on* or *off*.

$ad Current value of **ad** ("account distance") parameter of **s.cgi**(1). Can be either *on* or *off*.

$bd Current value of **bd** (search in body) parameter of **s.cgi**(1). Can be either *on* or *off*.

$ds Current value of **ds** (search in description) parameter of **s.cgi**(1). Can be either *on* or *off*.

$kw Current value of **kw** (search in keywords) parameter of **s.cgi**(1). Can be either *on* or *off*.

$tl Current value of **tl** (search in title) parameter of **s.cgi**(1). Can be either *on* or *off*.

$CC Template section **cached** if document is text/html or text/plain, or template section **textversion** if document was converted from other format.

$DB Document size in kilobytes.

$DC Document Content-type (like *text/html*).

$DD Document description (taken from META DESCRIPTION tag).

$DE URL-escaped document URL. Useful for building "cached link".

$DK Document keywords (taken from META KEYWORDS tag).

$DM Document Last-Modified date.

$DN Document number (in order of appearance).

$DS Document size in bytes.

$DT Document title.

$DR Document rating (as calculated by **ASPseek**).

$DU Document URL, unescaped.

$DX Document text. If excerpts are enabled and found, several excerpts from URL content, otherwise the first couple of lines, to give an idea of what the document is about.

$DZ Contents of template section **sizeb** if document size is less than 1024, otherwise contents of template section **sizek**.

$l Number of last document displayed on this page.

$t Total number of documents found.

Search statistics

$W Information about the number of word forms found. For example, if query was where first number is the URL count and second number is the total count). You can use **$W** only in **restop** section.

$Y Total time spent to perform query.

The following meta variables gets substituted by corresponding template sections, if some conditions are met.

$w Contents of **inres** template section if more than one result is found. **inres** template section should define either checkbox "Search within results" or link to "search within results" page.

$V Contents of **navigator** template section, if more than one page of results are found, otherwise empty.

$M

$M*n* Contents of **moreurls** template section, if grouping by sites is enabled and more than *n* results are found from the site, where *n* is number following **$M**, usually 2. If *n* is not specified, it is set to 1.

$R Contents of **ressites** section, if more that one site is found; otherwise contents of **resurls** section.

Navigator is a piece of HTML code used for presenting links to other search result pages. The following variables are used.

$NL Contents of **navleft** section, if current page > 0, otherwise contents of **navleft0** section.

$NB Contents of **navbar1** template section, repeated *pages per screen* times. For page number equal to current, contents of **navbar0** section is printed instead.

$NR Contents of **navright** section, if this page is not last one, otherwise contents of **navright0** section.

$NH URL of the another results page, used as a value for "<A HREF" in **navleft** and **navright** sections.

$NP Number of page, useful in **navleft***, **navbar*** and **navright*** sections.

$E Text of error message. See description of **variables** section for info about error messages.

TEMPLATE SECTIONS

top This section is included first on every page. So, you should begin this section with standard HTML preamble: `<HTML><HEAD>...` and so on. Here in this section you also provide search form.

The following meta variables are used:

$Q HTML-escaped search query.

$QF URL-escaped search query, concatenated with previous query if search was made within results).

$q The same as **$QF** if it is encountered in one of **res*** section, otherwise HTML-escaped search query, concatenated with previous query if search was made within results).

$A Argument for FORM ACTION tag with host name and protocol (example: `http://www.aspseek.com/cgi-bin/s.cgi`).

Example:

```
<!--top-->
<HTML>
<HEAD>
 <TITLE>ASPseek: $Q</TITLE>
</HEAD>
<BODY>
<FORM METHOD=GET ACTION="$A">
 Search for: <INPUT TYPE="text" NAME="q" SIZE=30 VALUE="$Q">
 <INPUT TYPE="submit" VALUE="Search"><BR>
 <INPUT TYPE="submit" NAME="t" VALUE="Take me there">
</FORM>
<!--/top-->
```

bottom This section is always comes last in every page. So you should provide all the closing tags which have their counterparts in **top** section. Although it is not obligatory to place this section at the end of template file, but doing so will help you to view your template as an ordinary HTML file in a browser to get the idea how it is looks like.

Example:

```
<!--bottom-->
</BODY>
</HTML>
<!--/bottom-->
```

restop This section is included just before the search results. It's not a bad idea to provide some common search results in this section. You can do so by using the following meta variables: **$f**, **$l**, **$t**, **$W**.

Example:

```
<!--restop-->
<TABLE BORDER=0 WIDTH=100%>
<TR>
<TD><small>Matches $f-$l of total <B>$t</B>  
Search took $Y seconds.</TD>
<TD><small>$W</small></TD>
</TR>
</TABLE>
<!--/restop-->
```

res

res2 These sections are used for displaying various information about every found document. **res** is used to display first document for each found site or for all documents is only 1 site is found (or grouping by sites is off). **res2** is used to display second document for each site if more than 1 site is found.

The following meta variables are used: **$D*** (where ***** is any character), **$CC**, **$CL**.

Example:

```
<!--res-->
<DL><DT>
<b>$DN.</b><a href="$DU" TARGET="_blank"><b>$DT</b></a>
[<b>$DR</b>]<DD> $DX...<BR>
<b>URL: </b><A HREF="$DU" TARGET="_blank">$DU</A>($DC)<BR>
$DM, $DS bytes<BR>
<b>Description: </b>$DD<br>
<b>Keywords: </b>$DK<br>
</DL>
<UL>
$CL
</UL>
<!--/res-->
```

moreurls

This section is displayed when more than 1 or 2 URLs is found on particular site (and grouping by sites is on). It is displayed in result just instead of $M2, $M1 or $M meta variables. The following parameters are used in this section:

$SS URL of home page of site of found document.

$SH URL to get more results from this site.

$SC Total number of documents found on this site.

ressites This section is displayed in case more than one site is found. See description of **$R**.

Example:

```
<!--ressites-->
Displaying sites $f-$l of total <B>$t</B> found.
<!--/ressites-->
```

resurls This section is displayed in case only one site is found. See description of **$R**.

Example:

```
<!--resurls-->
```

```
Displaying documents $f-$l of total <B>$t</B> found on site $SS.
<!--/resurls-->
```

Excerpts and cached copy highlighting

It is very convenient for user to have query words highlighted in search result excerpt and in cached copy. This is done using several sections.

hiopen

hiclose Used to to highlight the words in "cached" copy. Contents of these sections are printed before and after the found word.

hiexopen

hiexclose

Used in displaying excerpts, works the same was as **hiopen** and **hiclose**.

exopen

exclose Contents of this sections are displayed just before and after each excerpt found.

hicolors

Each line of this section should contain value of color for each search term. Value of color is taken from line with number equal to N mod C, where N is the search term sequential number and C is the total number of lines in this section.

The following meta variables are used:

$H String value of color to highlight, taken from "hicolors" templates section.

Below is an example of highlighting-related sections.

```
<!--hiopen-->
<B style="color:black;background-color:#$H">
<!--/hiopen-->
<!--hiclose-->
</B>
<!--/hiclose-->
<!--hiexopen-->
<B>
<!--/hiexopen-->
<!--hiexclose-->
</B>
<!--/hiexclose-->
<!--exopen-->
<br>...
<!--/exopen-->
<!--exclose-->
 ...
<!--/exclose-->
<!--hicolors-->
ffff66
ff66ff
66ffff
ff6666
6666ff
66ff66
<!--/hicolors-->
```

Clones

Clones are HTML documents with the same contents. **ASPseek** found such documents by comparing calculated MD5 checksums.

The following meta variables are used:

$CL List of clones.

Cached pages

The following section names are used to customize the appearance of "cached" pages (e.g. the pages with page contents stored in ASPseek's database).

cachetop

This section is included first on every "cached" page. The following meta variables can be used: **$DU**.

Displaying error messages

The following template sections are used to display various error messages.

notfound

As its name implies, this section is displayed in case when no documents are found. You usually give a little message saying that and maybe some hints how to make search less restrictive.

Below is an example of notfound section:

```
<!--notfound-->
Sorry, but search hasn't returned results.<P>
<I>Try to produce less restrictive search query,
or check words spelling</I>.
<!--/notfound-->
```

error This section is displayed in case some internal error occurred while searching. See the list of errors in description of **variables** section. Use the **$E** meta variable to print error message text.

Example of error section:

```
<!--error-->
<FONT COLOR="#FF0000">An error occurred.</FONT><P>
<B>$E</B>
<!--/error-->
```

queryerror

This section is displayed in case if query contains error like unmatched quote or parenthesis. You should use **$E** meta variable here.

Example of error section:

```
<!--queryerror-->
<FONT COLOR="#FF0000">Error in query: $E</FONT><P>
<B>$E</B>
<!--/queryerror-->
```

complexPhrase

This section is displayed in case if boolean expression is used inside phrase. Intended just to warn user, and probably give some hints.

complexExpression

This section is displayed in case if boolean expression is used in query. Intended just to warn user, and probably give some hints.

Including a file into template

include *filename*

If line `<!--include filename -->` is encountered, then **ASPseek starts loading template definitions from file, specified by** *filename*, then resumes processing of current file. Example of use: if two or more template definitions differ only in **top** and **bottom** sections, then create one file with sections other than **top** and **bottom**, then create two different files with **top** and **bottom** sections.

Special OPTION SELECTED substitutions

In order to save the value for option user selected across the different search result pages, the following "selected magic" hack is used. **s.cgi**(1) scans the template for all of <OPTION> tags in <SELECT> which do have attribute **SELECTED** followed by = sign and some value, and removes all of the **SELECTED** fields with the attribute different of that in **VALUE**.

In the following example the string `SELECTED` will only be presented in line that have its `VALUE` equal to value of **$ps** meta variable.

```
<SELECT NAME="ps">
<OPTION VALUE="10" SELECTED="$ps">10
<OPTION VALUE="20" SELECTED="$ps">20
<OPTION VALUE="50" SELECTED="$ps">50
</SELECT>
```

Defining different output formats

You can have several section with the same name in template. Normally, the first encountered section is used. This behavior can be overridden by supplying **o**=*n* parameter to **s.cgi**(1). If value of *n* is more than zero, then "*n*+1"th sections are used. If number of occurrences of particular section is less then value of "*n*+1", then last section with the needed name is used.

The following meta variables are used:

$o Substituted with the current value of **o** parameter of **s.cgi**(1).

User-defined template sections

Besides the standard template sections, user-defined sections can be used. **s.cgi**(1) treats line `<!--sectionname-->` as the start of user-defined template section and line `<!--/sectionname-->` as the end of user-defined template. Note that name of section must begin with alphanumeric character, otherwise this line is treated as regular HTML comment belonging to the current template section, so it is important to put space after "<!--" in comments.

User-defined templates can be used from other templates with the following meta variable.

$T*sectionname*

Gets substituted with user-defined template section named *sectionname*.

Random number generation

This is a feature of **s.cgi**(1) that allows you to include random numbers to resulting HTML, for example in order to get a random banner. Numbers are generated in the range [0; *max*], where *max* is equal to value of **Random***N*=*max* in **variables** section (see above). So, you put meta variable **$r***N*, which will be substituted to generated random number. *N* should be in the range 0...127, that means you can use up to 128 different random number variables.

In the example below **$r1** will be substituted with random number in the range from 0 to 100, **$r2** - from 0 to 500.

```
<!--variables
 ....
Random1=100
Random2=500
-->
 ....
<A HREF="http://www.my.com/click?id=$r1"><IMG
SRC=http://www.my.com/getbanner.cgi?id=$r1></A>
<A HREF="http://www.other.com/url?n=$r2"><IMG
SRC=http://www.other.com/pic?n=$r2></A>
 ....
```

BUGS

This documentation is incomplete. Not all variables can be used in all template sections. If you are unsure or see some strange misbehavior, consult the source code (file `templates.cpp`).

SEE ALSO

s.cgi(1), **aspseek**(7).

AUTHORS

Man page by Kir Kolyshkin <kir@asplinux.ru>

NAME

aspseek-sql - the structure of SQL database tables used by ASPseek

SQL TABLES

wordurl

This table keeps information about each word in main and real-time database, one record per word.

Field	Description
word	Word itself.
word_id	Numeric ID of **word**.
urls	Information about sites and urls, in which **word** is encountered. Empty if size of info is greater than 1000 bytes, in this case info is stored in separate file.
urlcount	Number of URLs in which **word** is encountered.
totalcount	Total count of this **word** in all URLs.

Last 3 fields are used only if **CompactStorage** is set to *no*, and updated after finishing of crawling, or then **index**(1) is run with **-D** option.

wordurl1

This table keeps all information about each word in real-time database, one record per word.

Field	Description
word	Word itself.
word_id	Numeric ID of **word**, refers to **wordurl.word**.
urls	Information about sites and urls in which **word** is encountered. Always not empty regardless of size.
urlcount	Number of URLs in which **word** is encountered.
totalcount	Total count of this **word** in all URLs.

Last 3 fields are updated immediately after downloading of the URL by **index**(1) when it is run with **-T** option.

urlword

This table keeps information about all encountered URLs, both indexed and not indexed yet which match specified conditions in configuration files.

Field	Description
url_id	ID of URL.
site_id	ID of site, refers to **sites.site_id**.
deleted	Set to 1 if server returned 404 error and **DeleteBad** is set to *yes*, or if **robots.txt** or configuration rules disallow to index this URL.
url	URL itself.
next_index_time	Time of next indexing in seconds from UNIX epoch.
status	HTTP status returned by server or 0 if document has not been indexed yet.
crc	MD5 checksum of document.
last_modified	"Last-Modified" HTTP header returned by HTTP server.
etag	"ETag" header returned by HTTP server.
last_index_time	Time of last indexing in seconds from UNIX epoch.
referrer	ID of URL which first referred this URL.
tag	Arbitrary tag.
hops	Depth of URL in hyperlink tree.
redir	URL ID, where current URL is redirected or 0 if this URL is not redirected.
origin	URL ID of document which is origin of this cloned document, or zero if this is not clone.

urlwords*NN* (where *NN* is 2-digit number from 00-15)

These tables contain additional info about existing indexed URLs. Number *NN* in table name is URL_ID mod 16.

Field	Description
deleted	Set to 1 if server returned 404 error and **DeleteBad** is set to *yes*, or if **robots.txt** or configuration rules disallow to index this URL.
wordcount	Count of unique words in the indexed part of URL.
totalcount	Total count of words in the indexed part of URL.
content_type	Content-Type HTTP header returned by server.
charset	Document charset taken from **Content-Type** HTTP header or META.
title	First 128 characters from pages title.
txt	First 255 characters from page body, stripped from HTML tags.
docsize	Total size of URL.
keywords	First 255 characters from page keywords.
description	First 100 characters from page description.
lang	Not used now.
words	Zipped content of URL.
hrefs	Sorted array of outgoing href IDs from this URL.

robots

This table contains information parsed from robots.txt file for each site.

Field	Description
hostinfo	Host name.
path	Path to exclude from indexing.

sites

This table contains IDs for all indexed sites.

Field	Description
site_id	ID of site.
site	Site name with protocol, like `http://www.my.com/`.

stat

This table contains information about query statistics for each completed query.

Field	Description
addr	IP address of computer, from which query was requested.
proxy	IP address of proxy server, through which query was requested.
query	Query string.
ul	URL limit used to restrict the query.
sp	Web spaces used to restrict the query.
site	Site ID used to restrict the query.
np	Results page number requested.
ps	Results per page.
sites	Number of found sites matching query.
urls	Number of found URLs matching query.
start	Query processing start in seconds from UNIX epoch.
finish	Query processing finish in seconds from UNIX epoch.
referer	URL of web page from which query was requested.

subsets

Table describing all subsets, which can be used to restrict the search. Populated manually with URL masks. Subset is the set of URLs from the particular directory of site. Putting masks describing whole site is not necessary.

Field	Description
subset_id	ID of subset.
mask	URL mask. Example: *http://www.my.com/dir/%*. Examples of wrong use: *http://www.aspstreet.com/%*, *http://www.aspstreet/%*.

spaces

Table describing web spaces. Web space is the set of sites. Each site belonging to particular space must be put into separate record. Populated manually or using **-A** option of **index**. If populated manually, run **index -B** after changing this table.

Field	Description
space_id	ID of web space.
site_id	ID of site belonging to the space, refers to sites.site_id.

tmpurl

Table describing URLs indexed since start of last indexing. Used for debugging.

Field	Description
url_id	URL ID.
thread	Ordinal thread number, which indexed URL.

wordsite

Auxiliary table used when search is restricted to site pattern. Built at the end of indexing from **sites** table.

Field	Description
word	Word used in site name between dots.
sites	Array of site IDs, where this word is encountered.

citation

This table contains reverse index of hyperlinks. It is used only if **IncrementalCitations** is set to *no*.

Field	Description
url_id	URL ID.
referrers	Array of URL IDs, which have hyperlink to this URL.

BLOBS

wordurl.urls, wordurl1.urls

Sites information, ordered by site_id.		
Offset	**Length**	**Description**
0	4	Offset of URL info for 1st site.
4	4	ID of 1st site where **word** is encountered.
8	4	Offset of URL info for 2nd site.
12	4	ID of 2nd site where **word** is encountered.
...		
(N-1)*8	4	Offset of URL info for Nth site, where N is the total number of sites in which **word** is encountered.
(N-1)*8+4	4	Offset of URL info for Nth site.
(N-1)*8+8	4	Offset of URL info end for Nth site. Must point to the end of blob or file.
URLs information. Follows sites information immediately. Offsets are counted from 0.		
Offset	**Length**	**Description**
0	4	URL ID of first site in sites info section.
4	2	Word count in this URL.
6	2	First position.
8	2	Second position.
...		
6+(N-1)*2	2	Nth position, where N is the total word count in the URL.
Repeated with info for URLs from the same site, with ID greater than previous.		
...		
Repeated with info for URLs for next sites from sites info section.		

urlwordsNN.words

This field contains gzipped content of URL.

Offset	Length	Description
0	4	Size of URL content before zipping or 0xFFFFFFFF if content is not zipped.
4	Zipped size	Zipped or original URL content.

wordsite.sites

This field contains array of sites/positions for word. Sorted by site IDs.

Structure of array element:

Bits	Description
24-31	Bitmap of positions, highest bit is set to 1 is word is first-level domain.
0-23	Site ID.

FILES

`/usr/local/aspseek/etc/`*DBType*`/tables.sql`

SEE ALSO

aspseek(7), **index**(1), **searchd**(1).

AUTHORS

Man page by Kir Kolyshkin <kir@asplinux.ru> and Alexander F. Avdonkin <al@asplinux.ru>.