SEARCHLIGHT

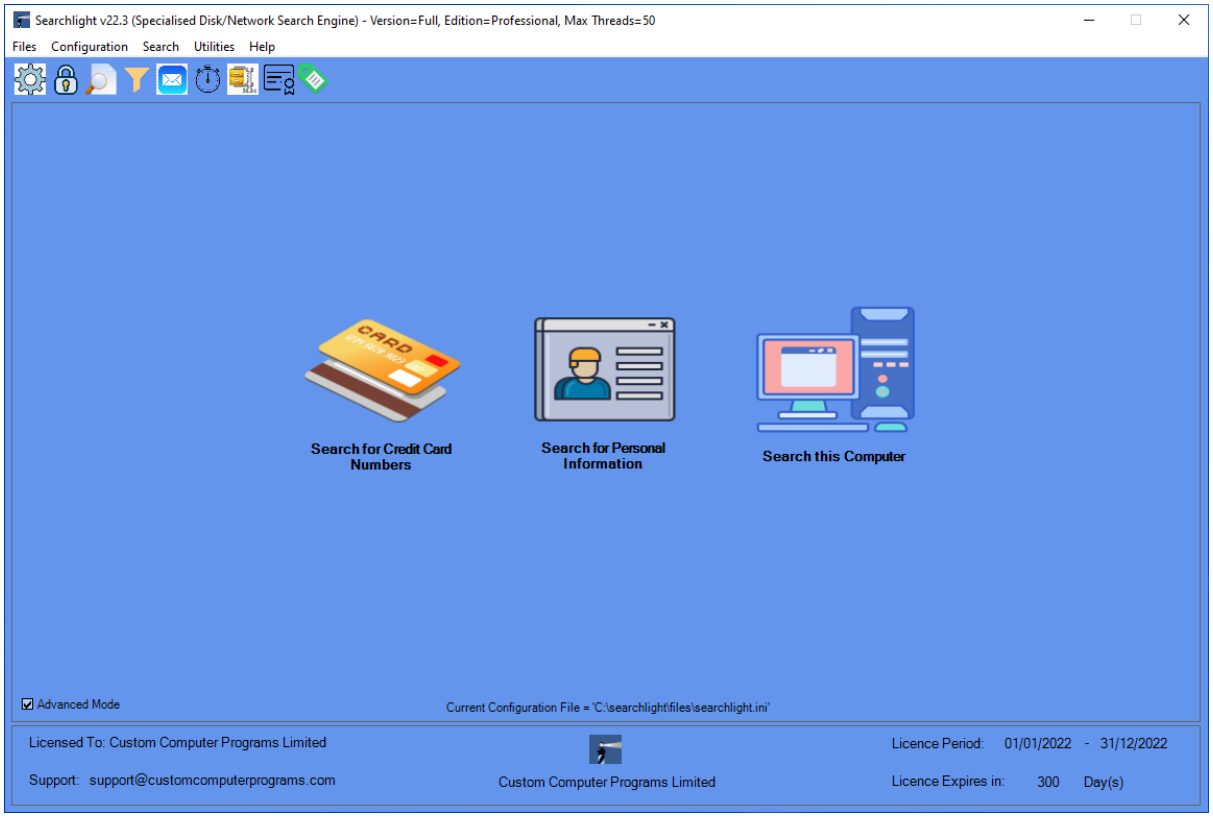
A Specialised Disk/Network Search Engine

Table of Contents

1. Overview

Searchlight is a premium file content search system that can search the contents of any type of file that is visible within Windows File Explorer and, unlike many file content search systems, it is fully multi-threaded, has unlimited scalability and can perform remote distributed, and federated searches via a network connection.

Searchlight's main screen is as shown below:



Searchlight was originally developed to search the contents of files stored within high grade government and military computer networks to identify files that contained inappropriate content. For instance, on a network accredited to hold information up to a classification of 'Restricted', Searchlight would be used to find any file that existed on that network with a classification that exceeded 'Restricted'.

1.1 Selection of Searchlight Use Cases

Searchlight can be used for a multitude of purposes including but not limited to:

- Scanning for classified material of an inappropriate classification
- Scanning legal files looking for names, addresses
- Scanning commercial contract files looking for commercial in confidence
- Scanning files for project names, part numbers, company names etc
- Scanning log files for keywords, IP addresses, Errors and host names etc
- Scanning files looking for passwords
- Scanning files looking for personal data such as telephone numbers, email addresses, post codes, bank details, credit card numbers etc.

There are many file content search programs available and many of them are free. However, Searchlight, and its sister console program slsearch, have been specifically designed to provide many features that other programs simply don't have. In addition, Searchlight has been developed and tested in some of the most demanding environments, over many years, and is fully supported with regular free software updates available to valid licence holders.

Some of the key features that Searchlight provides, that many other search programs do not, are as follows:

1.2 Searchlight Features

1.2.1 Multi-Threaded

Searchlight has been specifically written to scale and exploit multi-core processors and the number of available 'threads' is limited only by the power of the host computer and the number of threads declared within the licence. Thus, setting the configuration to 100 threads allows Searchlight to search up to 100 files 'in parallel' delivering a dramatic improvement in speed, and scalability, over non multi-threaded programs.

1.2.2 Ability to Save Multiple Search Definitions

It is quite common, over a period of time, to perform the same search multiple times. For instance, you may wish to search a particular directory at regular intervals or you

may receive a CD/DVD on a regular basis and want to search it (sheep dip it!) before loading it on to a network.

Searchlight enables an unlimited number of search definitions to be saved and subsequently reloaded and executed on demand.

1.2.3 Ability to Search Against a list of Search Definitions

Most file content search systems only search for a single keyword/phrase at a time. However, Searchlight can search for an unlimited number of keywords/phrases all at the same time as part of a single search. In addition, Searchlight can maintain an unlimited number of search definition files with each one being associated with a different configuration if required.

1.2.4 Ability to search for ASCII/Hex byte sequences

Searchlight can search for multiple sequences of ASCII and HEX bytes (referred to as Signatures'). Each search definition can contain a searchword, signature or phrase and a mixture of up to 5 'anded' or 'noted' searchwords, signatures or phrases. In addition, each of the search definitions can be case aware.

1.2.5 Statistical breakdown of search results

Searchlight provides a complete statistical breakdown of its search results. This is particularly useful if a network has been contaminated and the extent of the contamination is to be determined in order to decide on what action to take. For instance, if a network is not permitted to contain confidential information and confidential information is found then management can decide whether to scrub individual files or to replace the affected disks. Without having the statistic breakdown showing the extent of any contamination it would be difficult to decide on a particular course of action.

1.2.6 Automatically Email Search Results

If appropriately configured, Searchlight can automatically send its results log file to a nominated email address. This feature is particularly useful when Searchlight's sister console program (slsearch) is executed via a scheduler.

1.2.7 Context Viewing of Results

On finding a targeted searchword, signature or phrase, Searchlight extracts a portion of the surrounding text, from each side of the found searchword, and presents this for viewing in a results listbox. This allows the search results to be reviewed 'in context' and in many cases negates the need to open the 'found file' for further investigation.

1.2.8 Searching Remote Hosts (via Encrypted Links

The Enterprise version of Searchlight has the ability to search the contents of files stored on an unlimited number of remote computers. When appropriately configured, Searchlight communicates to its remote search slaves via an encrypted link thus ensuring that the confidentiality of any search results, whilst in transit, is maintained.

1.2.9 Encrypting Search Results Log Files

In certain environments, it would not be wise to expose Searchlight results in clear text. For instance, it wouldn't be good practice for a results file, containing a list of filenames that contained the string 'password=' or a series of credit card numbers, to be stored in plain text. As such, Searchlight provides the option of storing its log files in encrypted format.

1.2.10 Federated Search

The 'Federated Search' is a highly specialised feature that sets Searchlight apart from its competitors, enabling extremely large data stores to be searched utilising the distributed power of potentially an unlimited number of host computers.

Searchlight utilises a proprietary algorithm that can efficiently search a large remote data store, connected/mapped to an unlimited number of hosts, without having to prebuild an index. In addition, Searchlight ensures that each remote host instantiates its own number of threads in accordance with its configuration as stored within Searchlight.

1.2.11 Search Using Regular Expressions

Regular Expressions are definitions that are used to pattern match strings. Searchlight can use Regular Expressions not only as a main search string but also as any one of a searchword's associated compound search criteria.

1.2.12 Searching Zip Files

Searchlight can 'recursively' search the contents of Zip files. Most other content search programs are not able to search Zip files whilst the few that do can typically only search a single Zip file (not a Zip file within a Zip file within a Zip file etc.) – Searchlight and slsearch CAN.

1.2.13 Utilising Apache Tika

Searchlight, if appropriately configured, can call the open source Apache Tika program to extract text from hundreds of file types which Searchlight then subsequently searches.

2. Configuration

Searchlight does not use the Windows registry to store its configuration but instead uses a series of value equal pairs stored in a number of simple text files.

A key design principle of Searchlight's was to keep the system as simple as possible and to avoid complexity wherever it is not needed. One way in which Searchlight strongly adheres to this design principle is in the way that it stores, makes available, and manages its configuration.

Many Windows systems today store their configuration values in the system registry which, although may be desirable in some circumstances, it is seen by many as an over complication. Thus, in line with the design principle of keeping things simple, Searchlight uses a number of text based configuration files utilising the simple format of value equal pairs. The value equal pair format is very reminiscent of the way in which Windows used to store its configuration using files such as win.ini and system.ini in days of old. Files in the value equal pair format can be easily accessed and edited quickly using a simple text editor, such as notepad, although Searchlight does have its own built in configuration editor.

The two main Searchlight configuration files are:

searchlight.ini

searchlight_admin.ini

The 'searchlight.ini' file is Searchlight's principal configuration file and is the first file that is read when Searchlight starts up.

Once the 'searchlight.ini' file has been read, Searchlight then reads the 'searchlight_admin.ini' file and any configuration value read from this file will replace any similarly named configuration value previously read from the 'searchlight.ini' file.

Thus, Searchlights configuration is established as follows:

Initially, all configuration values are set to a default value, as defined internally within Searchlight/slsearch and these values are fixed.

Secondly, Searchlight/slsearch reads the contents of file 'searchlight.ini'/'slsearch.ini' and overrides any similarly named value that was set from the internal default configuration.

Finally, Searchlight/slsearch reads its associated admin file 'Searchlight_admin.ini' or 'slsearch_admin.ini' overriding any previously read configuration value with the same name. The admin file is mandatory and if it is not found then Searchlight/Slsearch will not run.

Thus, any configuration value can be enforced by simply setting the configuration value in the admin file and then securing the file appropriately using Windows file permissions to ensure that such values cannot be edited.

2.1 Select Configuration File

On selecting this menu option, from the 'Configuration' menu, Searchlight displays a file dialog box displaying all of the configuration files available (i.e. those files with a file extension of '.ini').

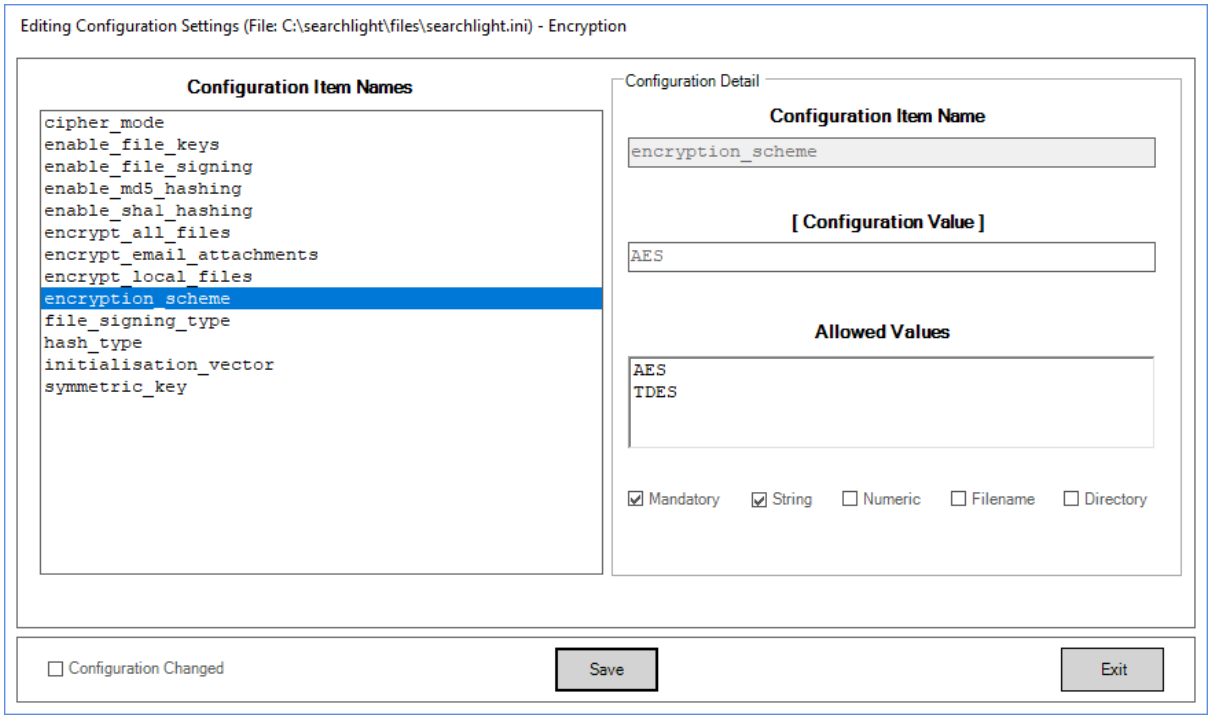
On selecting a configuration file, its contents are read and Searchlight's configuration is adjusted accordingly. In order to maintain a level of security, Searchlight's/Slsearch's admin file is then subsequently read, as previously described, and its configuration values are loaded, overwriting any previously loaded configuration item with the same name.

2.2 Edit Current Configuration

On selecting the <Configuration><Edit Current Configuration> option, Searchlight displays a sub menu displaying the configuration categories available. As there are

literally hundreds of configuration settings, this menu provides an easy way of accessing a subset of related configuration items.

E.g. on selecting the configuration sub menu 'Encryption', the following screen is displayed providing easy access to the encryption configuration settings:

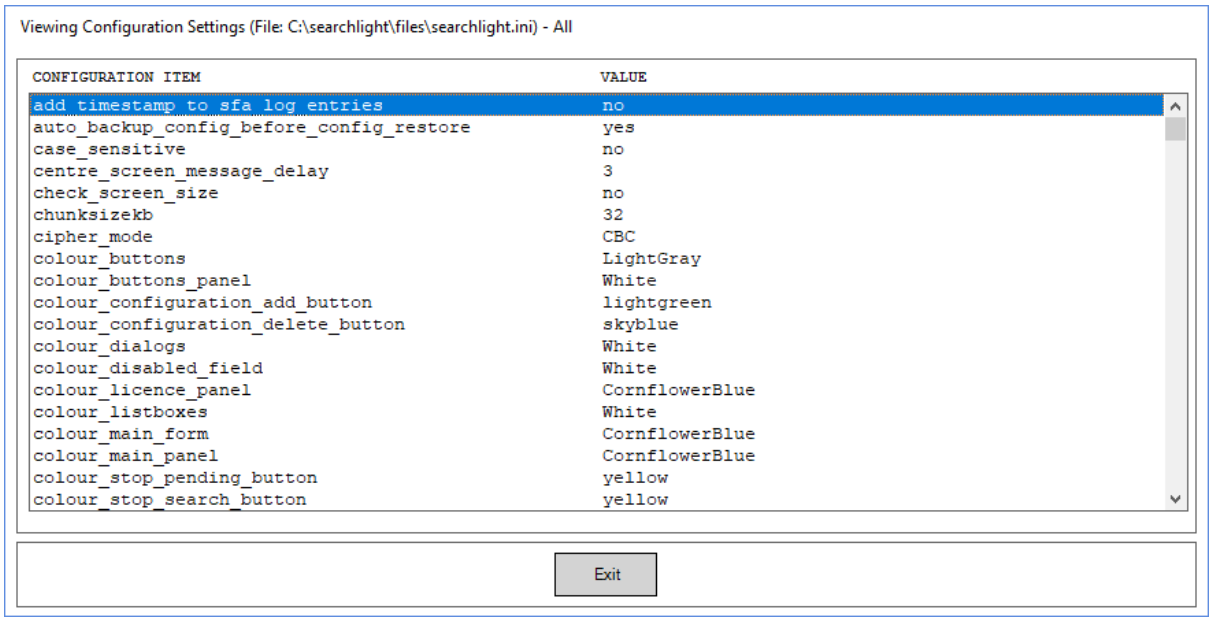


This configuration edit interface provides a controlled window over the currently selected configuration file which by default is 'searchlight.ini'. When a configuration item is selected, from the left hand side listbox, the configuration item, along with its current value, is displayed over on the right hand side of the screen where it can be edited as required.

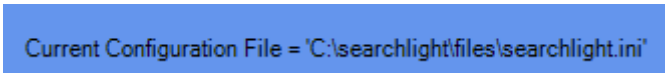
Certain configuration items, such as the selected item above 'encryption_scheme', can only be set to a defined number of values. In the example above, it can be seen that the 'encryption_type' configuration item can only be set to 'AES' or 'TDES'.

2.3 View Current Configuration

On selecting the <Configuration><View Current Configuration> option the following screen is displayed:



This screen which is Read-Only, displays in alphabetical order all of Searchlight's configuration items along with their current values. Towards the bottom centre of the main screen, Searchlight displays the full path name of the current/active configuration file as shown below:



2.4 Deep Search

Deep Search is one of Searchlight's most popular and unique features.

Searchlight searches every file, irrespective of its file type or its configuration settings, using its own proprietary search algorithm.

However, the data stored within certain types of files, such as Microsoft Office files and PDF files, is not stored as plain text which makes them difficult to search.

Searchlight's Deep Search is specifically targeted against Microsoft Office files and will only work if Microsoft Office is installed on the same machine as Searchlight.

Prior to each Deep Search, Searchlight automatically creates a work directory and a series of work files under the Searchlight installation directory.

Deep Search increases the search time but it is recommended in those circumstances where extra scrutiny of files is required such as scanning a CD/DVD, provided by a third party, prior to loading on to a secure network.

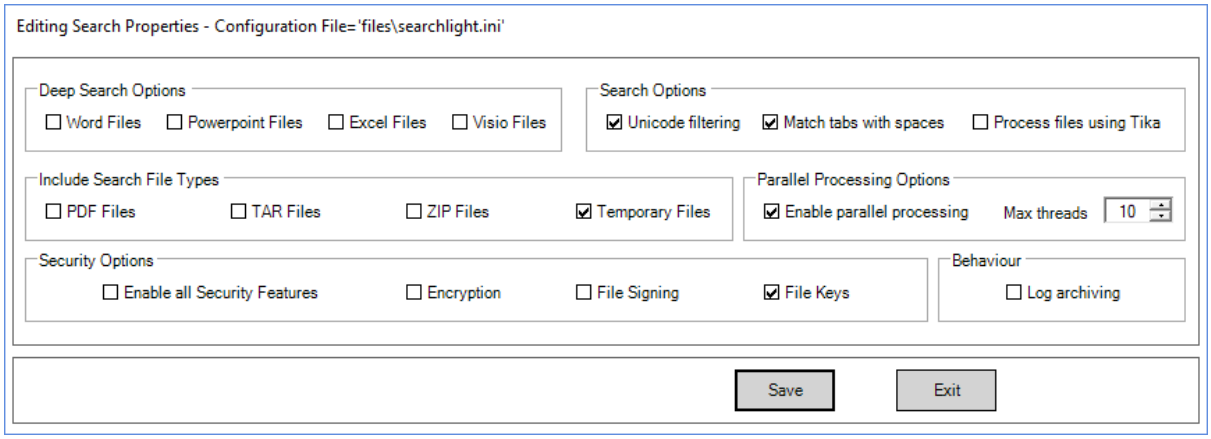
2.4.1 Enabling Deep Search

Deep Search is not enabled by default but can be selectively enabled by setting the following configuration options to 'yes':

- 'deep_search_excel_files'
- 'deep_search_powerpoint_files'
- 'deep_search_visio_files'
- 'deep_search_word_files'

As it can be seen by the configuration options above, Deep Search can be selectively enabled or disabled for a number of individual Microsoft Office file types.

An easier and alternative way of enabling Deep Search is to select the 'Config' button from the main search screen whereupon the following screen is displayed:



The 'Deep Search Options' group box checkboxes provide an easy way to quickly enable and disable Deep Search for specific Microsoft Office file types.

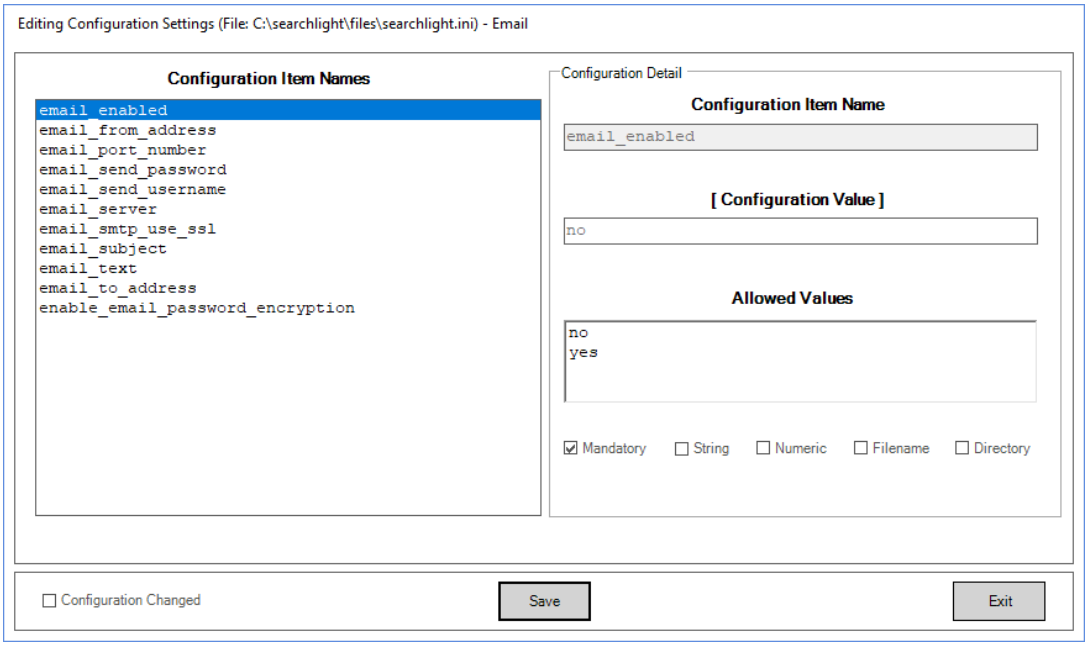
2.5 Email Log Files

On successful completion of a Search, Searchlight can, if appropriately configured, automatically send the results log file to a nominated email account.

This functionality can be configured by selecting the following menu options:

Configuration -> Edit Current Configuration -> Email

On selecting these menu options, the following screen is displayed whereupon the email configuration settings can be defined as described below:



2.5.1 email_enabled

Select 'yes' from the 'Allowed Values' to enable the email functionality.

2.5.2 email_from_address

This is set to an email address that, when the email is received, shows who the email was sent from.

2.5.3 email_port_number

This is the port number that the receiving email server will listen on in order to accept the email being sent.

The port numbers 25, 465 and 587 are provided in the 'Allowed Values' list box. However, any port number can be defined by the user and will depend upon the port number that the receiving email server is listening on.

2.5.4 email_send_password

In order to send an email from a named account it is necessary to login to that account first and this is the password used as part of that login process.

2.5.5 email_send_username

This is the name of the email account that is logged into and used to send the results log file from. This account is logged into using the username provided for this setting in conjunction with the password provided for the 'email_send_password' setting.

2.5.6 email_server (e.g. smtp.gmail.com

This is the name of the email server that the email will be sent to.

2.5.7 email_smtp_use_ssl

This is set to either 'yes' or 'no' and depends on the email server's requirements.

2.5.8 email_subject

This defines the text that is to be entered into the subject line of the email being sent and defaults to 'Searchlight Results'.

2.5.9 email_text

This defines the text that will be copied into the body of the email being sent and defaults to 'Searchlight Results for Date:' with the date being automatically appended to the end of this text.

2.5.10 email_to_address (e.g. logarchive@gmail.com)

2.5.11 enable_email_password_encryption

This defines whether email passwords are stored in encrypted format or not.

3. Found File Utility (Only available in Professional/Enterprise Editions)

Searchlight is not primarily a Forensics Tool. However, the Found File Utility provides a number of powerful features that can be used to benefit a Forensic investigation such as:

- Packaging up all files found during a search into a zip file.
- Creating a CSV record (line), for each found file, recording the full filename, file creation date, file last modified date, file last accessed date, file size and the following file hashes: MD5, SHA-1, SHA-256, SHA-384 and SHA-512.
- Creating a separate file containing just the hash of the CSV file just described.
- Creating a tuple file containing the values outline above in bullet point 2.
- Creating a file containing just a hash of the tuple file mentioned above.
- Optionally, automatically adding all of the files above into a single zip file.
- Creating a text file containing, on each line, the full filename of each file found.
- Creating a batch file containing a command consisting of a filename surrounded by a user defined command prefix and suffix.

3.1 Found File Utility Configuration (searchlight.ini)

Searchlight's Found File Utility configuration settings are maintained in the active main configuration file which, by default, is 'searchlight.ini'.

The Found File Utility configuration values can be viewed and edited by via the menu options:

'Configuration' -> 'Edit Configuration' -> 'Found File Utility'.

Alternatively, a subset of the Found File Utility configuration settings can be viewed and edited via the 'Zip File Configuration' screen which can be accessed via the menu options:

'Utilities' -> 'Zip File Configuration'.

The default Found File Utility configuration values are as follows:

found_file_utility_add_found_files_to_zip_file=yes

found_file_utility_add_csv_file_to_zip_file=yes

found_file_utility_add_data_file_to_zip_file=yes

found_file_utility_add_filenames_file_to_zip_file=yes

found_file_utility_add_batch_file_to_zip_file=yes

These settings define whether certain files are added to the resulting zip file.

found_file_utility_data_hash_type=MD5

found_file_utility_csv_hash_type=MD5

These settings define the hashing algorithm used to create the hash value for the resulting CSV file (*.csv) and tuple metadata file (*.dat).

It should be noted that the hash values for the aforementioned Metadata files are written to a text file with the same filename (as the file the hash was calculated for) but with the name of the hashing algorithm suffixed to the file extension.

For example, if the MD5 algorithm was used to calculate the hash value of Metadata file 'metadata_202004211505.csv' then the resulting MD5 hash would be written to a file called 'metadata_202004211505.csvmd5'.

found_file_utility_delete_batch_file=no

found_file_utility_delete_csv_file=no

found_file_utility_delete_csv_hash_file=no

found_file_utility_delete_filenames_file=no

found_file_utility_delete_zip_filenames_file=no

found_file_utility_delete_data_file=no

found_file_utility_delete_data_hash_file=no

These values define whether the files, produced by the Found File Utility, (assuming that 'found_files_utility_enabled'=yes) are deleted or not. It should be noted that there is no option to delete any resulting zip file.

It can be seen that the Found File Utility could potentially create 7 files (6 Metadata files and a zip file). Thus, to avoid the creation of a lot of transient Metadata files, it is suggested that these values are set to 'no'.

The following configuration settings are self-explanatory and can all be accessed and modified via the Found File Utility Configuration menu except those highlighted in red.

found_file_utility_directory=saved_files

found_file_utility_enabled=no

found_file_utility_internal_compression_level=0

found_file_utility_max_zipfile_sizemb=10000

found_file_utility_md5_enabled=yes

found_file_utility_metadata_filename_prefix=metadata_

found_file_utility_sha1_enabled=no

found_file_utility_sha256_enabled=no

found_file_utility_sha384_enabled=no

found_file_utility_sha512_enabled=no

found_file_utility_zip_buffer_sizekb=32

found_file_utility_zip_engine=internal

found_file_utility_zip_filename_prefix=found_files_

The configuration settings highlighted in red can be edited by selecting the following menu options:

'Configuration' -> 'Edit Configuration' -> 'Found File Utility'

3.2 Found File Utility Configuration Screen

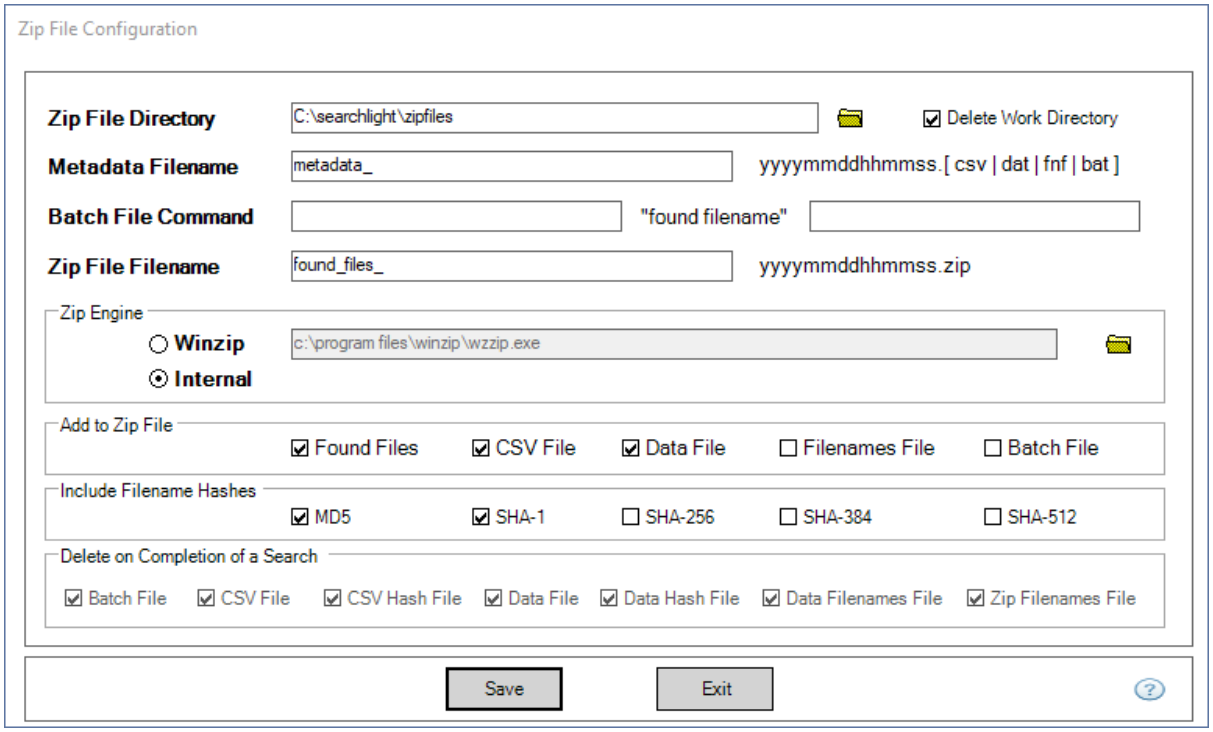
Searchlight's Found File Utility configuration is maintained in the main Searchlight configuration file which by default is 'searchlight.ini'.

A Searchlight configuration file can be defined for each 'saved search definition' and, as such, each search performed by Searchlight could potentially have a different Found File Utility configuration. However, typically the 'searchlight.ini' file is used for all searches.

Although the Found File Utility configuration can be accessed by selecting the menu options:

Configuration -> Edit Configuration -> Found File Utility

it can also be accessed by selecting the 'Zip File Configuration' option from the Utilities menu whereupon the following screen is displayed:



As mentioned, the configuration is derived, and this screen populated, from the active Searchlight configuration file which by default is 'searchlight.ini'.

A description of each screen field is as follows:

3.2.1 Zip File Directory

This field defines the directory in to which Searchlight will write any Found File Utility file(s) that it produces. It should be noted that the Found File Utility definition cannot be saved unless the directory specified in this field exists.

Metadata Filename

This field defines the name of the Metadata files (*.csv, *.dat, *.fnf, *.bat) produced by the Found File Utility. The Metadata filename field defines the filename prefix used to construct the name of the Metadata files produced.

Each Metadata file produced is named with a timestamp suffix of the format: 'yyyymmddhhss'.

Thus, if the Metadata field is defined as 'metadata_' then the following metadata files would be produced (assuming that the date and time of writing the files was 21st April 2020 and 15:05:03):

- metadata_20200421150503.csv
- metadata_20200421150503.dat
- metadata_20200421150503.fnf
- metadata_20200421150503.bat

For each file found by Searchlight, a CSV Metadata record is produced consisting of the following comma separated fields/values:

- Full path filename
- File creation date
- File last accessed date
- File last modified date
- File size (expressed in bytes)
- A MD5 hash value for the file
- A SHA-1 hash value for the file
- A SHA-256 hash value for the file
- A SHA-384 hash value for the file
- A SHA-256 hash value for the file

The same Metadata values, dependent upon the settings of the checkboxes as shown below, are also written to a '.dat' Metadata file.

Thus, in the example above, only the MD5 and the SHA-1 hash values, for each of the found files, will be written to the tuple metadata file (*.dat).

3.2.2 Batch File Command

This is a very powerful feature allowing a batch file command to be created for each of the files found as the result of a search. The example below shows that for each of the filenames found, the following line will be written to the Found File Utility batch file:

copy "the full path filename of a found file" c:\temp

The batch file is created with the .bat extension and, assuming that a valid command line is created, it can be executed just like any other batch file.

3.2.3 Zip File Filename

This field defines the name of the zip file produced by Found File Utility. Like the 'Metadata Filename' field, as described above, it allows the resulting zip file to be named with a user defined prefix and, like the name of the Metadata file, the filename will be suffixed with a timestamp but with a '.zip' file extension.

3.2.4 Zip Engine (Group Box)

This group box defines the compression mechanism to be used in order to produce the zip file, as specified in the 'Zip Filename' field, and consists of the following two options:

- ***Winzip***

  This must be set to the full pathname of an installed version of the Winzip command line utility, which at the time of writing, is 'wzzip.exe'. Thus, if this

option is selected, Searchlight will invoke the 'wzzip.exe' utility to produce the resulting Found File Utility zip file as named in the 'Zip Filename' field.

- ***Internal***

  If this option is selected, then Searchlight will use its own internal compression mechanism to produce the zip file as specified by the 'Zip Filename' field.

  Thus, if the Winzip command line utility 'wzzip.exe' is not installed then this option must be selected.

3.2.5 Add Files to Zip File (Group Box)

The options, shown below, define which files are added to the resulting Found File Utility zip file. The example below shows that the files found by Searchlight, the CSV file (*.csv), the tuple metadata file (*.dat) and the filenames file (*.fnf) will all be added to the resulting zip file but the batch file (*.bat) will not be added.

As mentioned, the initial settings of the Found File Utility Configuration screen, are derived from the values in the active configuration (in 'searchlight.ini' by default).

3.2.6 Include Hashes (Group Box)

These options, shown below, define which file hashes are included in the tuple metadata file (*.dat). It should be noted that, irrespective of these settings, all of the file hash values will always be written to the CSV metadata file (*.csv).

4. Licencing

Searchlight is typically licenced for a period of 12 months and the licence file, 'searchlight.lic' stored in the '.../files' directory, can be viewed by selecting the following menu options:

Utilities -> Licence -> View Licence File

The licence file contains the following values:

product=searchlight
licence_site_name=Public - To be used for evaluation purposes only
copyright=(c) Custom Computer Programs Limited
email_support=support@customcomputerprograms.com
distribution_identifier=000695
max_threads=100
edition=Professional
version=Demo
searchlight_identifier=012722
search_logging_enabled=yes
licence_start_date=20190331
licence_end_date=20200330
SIGNATURE=<A string generated using CCL's private key>

The values in blue are fixed whilst the other values may be different for each issue.

Searchlight validates the licence file at start-up using the SIGNATURE value, stored on the last line of the licence. If the licence fails validation then Searchlight will run but will have a reduced number of menu options available.

4.1 Updating the Licence File

Typically, a new Searchlight licence is received via email with the licence being attached to the email as a simple text file.

The following describes the two ways that a new Searchlight licence file can be installed:

- Using the Searchlight 'Import Licence' option

  By selecting the options "Utilities -> Licence – Import Licence File" a file dialog is presented whereupon a licence file can be selected for import. When a licence file is selected its validity is checked via the SIGNATURE value at the bottom of the file.

  Once a new licence file is successfully validated and subsequently imported, the old licence file is renamed as follows: "searchlight_YYYYMMDDHHMMSS.lic".

  Any changes reflected in the new licence file, for example, an increase in the 'max_threads' value, are immediately reflected in the running program.

- Copying the licence file into the '.../files' directory

  A new licence file can be simply copied into the Searchlight '.../files' directory, overwriting the existing licence file.

  In this case, Searchlight will check the validity of the licence the next time it starts-up and, as a consequence, if the new licence has different values than the old licence then the new values may not be immediately reflected in the running program (a program restart is recommended).

4.2 Protecting Saved Search Definitions

In some circumstances it may be necessary to protect search definitions from change or to enforce users to perform certain searches.

The ability to edit search definitions is governed by the following configuration settings:

- fixed_searches_only (set to 'yes')

  All fields on the search screen are made read only and any search loaded (by selecting the 'Load' button) will also be read-only.

- enable_edit_files_in_edit_search

  When this configuration item is set to 'no' then none of the files shown on the search screen can be edited. This is a catch-all setting and negates having to make each file individually read-only.

- enable_select_searchwords_file

  When this configuration item is set to 'no' then the user is prevented from selecting a different searchwords file from the one shown.

  It should be remembered that a searchwords file is only displayed when the 'What to Search For' dropdown value equals 'File of Searchwords'.

- enter_screen_defaults

  When this configuration item is set to 'yes' then on entry to the 'search screen', a number of fields will be pre-populated with default values (as specified in the active configuration file).

- show_edit_saved_searches_button

When this configuration item is set to ‘no’ then no search definitions can be edited.

It should be noted that no editing restrictions are applied if the configuration setting ‘execution_mode’ is set to ‘admin’.

5. Log Archiving

When Log Archiving is enabled, Searchlight automatically copies log files to a specified directory.

By default Log Archiving is disabled but can be enabled as follows:

- Select the following menu options:

  Configuration -> Edit Current Configuration -> Logging

- Select 'enable_log_archiving' and then select 'yes' from the 'Allowed Values'.

- Select the 'Update' button and then exit and save changes.

When log archiving is enabled, as described above, log files are automatically copied into the directory specified by the configuration setting 'log_archive_directory'.

When a log file is archived, its filename is prefixed with a Date Timestamp, of the format 'YYYYMMDDHHMMSS', to ensure that its filename is unique.

All log files, including archived log files, can be viewed by selecting the following menu options:

Utilities -> View Log Files

6. Performance

Searchlight is a fast Search Engine. However, one of the downsides of a fast process is that it can consume system resources (CPU/Memory) with the consequence of making the machine unresponsive to other user actions.

In order to prevent this, Searchlight provides the following configuration items that can be set to limit Searchlight's ability to consume excessive amounts of system resources:

event_timer_enabled

event_timer_poll_rate_milliseconds

sleep_timer_enabled

sleep_timer_poll_rate_milliseconds

sleep_timer_delay_milliseconds

milliseconds_delay_between_files

milliseconds_delay_between_reads

6.1 Event_timer_enabled

This value defines whether the 'event timer' is enabled or not. The event timer plays an important role allowing the Windows Operating System to process Windows 'events' whilst a Searchlight search is in progress.

It is recommended that the Event Timer is always enabled and the value 'event_timer_poll_rate_milliseconds' is set to a minimum of at least 100 milliseconds.

6.2 Event_timer_poll_rate_milliseconds

This value defines the number of milliseconds between each time that Searchlight will process Windows Operating System events.

6.3 Sleep_timer_enabled

This value defines whether the 'sleep timer' is enabled or not. The sleep timer plays an important role and, when enabled, delays search processing, when the Sleep Timer is fired, by the number of milliseconds, defined by the configuration value 'sleep_timer_delay_milliseconds'. The Sleep Timer is fired, if enabled, every number of milliseconds as defined by the configuration value 'sleep_timer_poll_rate_milliseconds'.

6.4 Sleep_timer_poll_rate_milliseconds

This value defines the number of milliseconds between each invocation of the Sleep Timer.

6.5 Sleep_timer_delay_milliseconds

When the Sleep Timer fires, assuming it is enabled, this value defines the number of milliseconds that the search will delay, thus ensuring that the search processes/threads do not consume all of the system CPU/Memory resources.

6.6 Milliseconds_delay_between_files

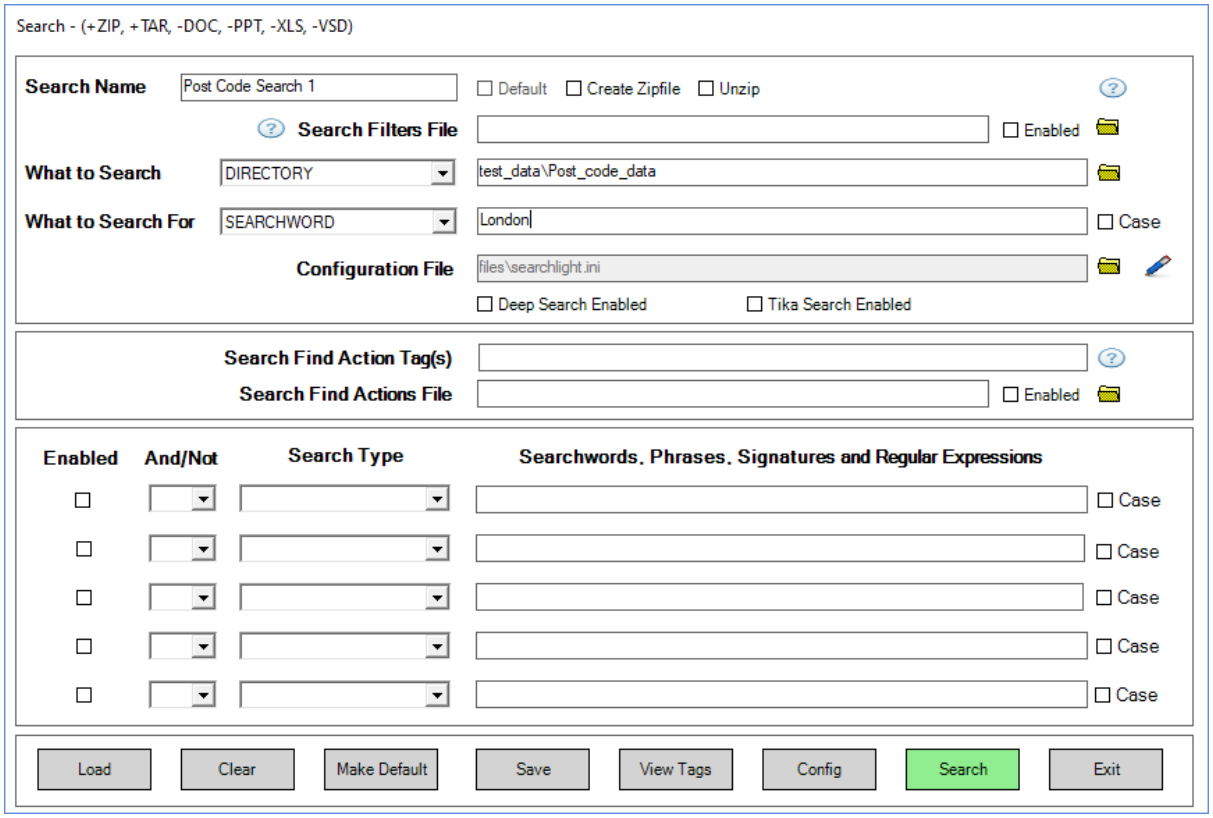
This configuration value defines the number of milliseconds that Searchlight must delay between the processing of each file.

6.7 Milliseconds_delay_between_reads

This configuration value defines the number of milliseconds that Searchlight must delay between each read of a file.

7. Local Search

On selecting a search icon from the main screen, the following search screen is displayed:



All local searches are instigated from this screen which provides a number of search options, all presented in a compact and intuitive format.

If the 'Create Zipfile' checkbox is checked then a zipfile will be automatically created, containing all of the found files, in the directory specified by the configuration item 'zipfiles_directory' .

If the 'Unzip' checkbox is checked then the zipfile will automatically be unzipped into a subdirectory alongside the zipfile itself.

7.1 Search Field Descriptions

7.1.1 Search Name

Each search definition, saved within the file 'savedsearches.ssf' by default, must have a unique name used to uniquely identify each search definition.

7.1.2 Search Filters File

This file defines any associated filters that are to be applied to the search but only if the associated 'Enabled' checkbox is checked.

7.1.3 What to Search

This dropdown box is used to select 'what is to be searched' and offers the following values:

- File
- File of Filenames
- Directory
- File of Directory names

7.1.3.1 File

This specifies the name of the file to be searched.

The 'edit icon' is not displayed for this option as individual files, from within Searchlight, cannot be edited.

7.1.3.2 File of Filenames (.fnf)*

This specifies a text file containing a list of filenames of those files to be searched (one filename per line).

The 'edit icon' allows the contents of the file, as specified by 'File of Filenames', to be edited and is enabled under the following conditions:

- A filename is specified
- The file specified exists
- The menu option 'Edit File of Filenames' is visible
- Configuration option 'enable_file_editing' is set to 'yes'
- Configuration option 'fixed_searches_only' is set to 'no'

7.1.3.3 Directory

This specifies the name of the directory/folder, and all of its subdirectories, to be searched.

7.1.3.4 File of Directory/Folder names (.dnf)*

This specifies a text file containing a list of directory names/folders to be searched (one directory name/folder per line).

The 'edit icon' allows the contents of the file, as specified by 'File of Directories', to be edited and is enabled under the following conditions:

- A filename is specified
- The file specified exists
- The menu option 'Edit File of Directories' is visible
- Configuration option 'enable_file_editing' is set to 'yes'
- Configuration option 'fixed_searches_only' is set to 'no'

7.1.4 What to Search For

This dropdown box specifies 'what you are searching for' and offers the following values:

- Searchword
- Signature (ASCII)

- Signature (HEX)
- Regular Expression
- File of Searchwords
- Filenames (Filtered)

7.1.4.1 Searchword

This can be either a single word or any string of characters.

7.1.4.2 Signature (ASCII)

This allows a series of up to 10 consecutive ASCII byte values to be searched.

7.1.4.3 Signature (HEX)

This allows a series of up to 10 consecutive HEX values to be searched.

7.1.4.4 Regular Expression

This allows a Regular Expression to be used as a search definition.

7.1.4.5 File of Searchwords

This specifies a text file, created and maintained from within Searchlight, that holds one or more Searchword definitions to be searched.

The 'edit icon' allows the contents of the file, as specified by 'File of Searchwords', to be edited and is enabled under the following conditions:

- A filename is specified – Mandatory
- The file specified exists – Mandatory
- The menu option 'Edit File of Searchwords' is visible
- Configuration option 'enable_file_editing'

7.1.5 Configuration File

The configuration used for a search is dependent upon the configuration setting 'search_configuration' which may be any of the following:

- use active configuration

  This instructs the search to use the current active configuration and not the configuration as specified by the value of the field 'Configuration File'.

- use search configuration

  This instructs the search to use the configuration as defined within the file specified in the field 'Configuration File'.

- use search configuration and make active

  This instructs the search to use the configuration as defined within the file specified in the field 'Configuration File' and then to subsequently make the configuration the active configuration.

7.1.6 Search Find Action Tag(s)

This field defines none, one or more Search Find Action tags to be associated with any file that is found. These tags are then processed by the Search Find Actions engine as documented in the 'Search Find Actions' help file.

7.1.7 Search Find Actions File

This is the file containing the Search Find Actions that are to be processed if the associated 'Enabled' checkbox is checked.

7.1.8 Compound Searchword Definition Panel

This panel is only enabled when the Search Target dropdown value is one of the following:

- Searchword
- Signature (ASCII)
- Signature (HEX)
- Regular Expression

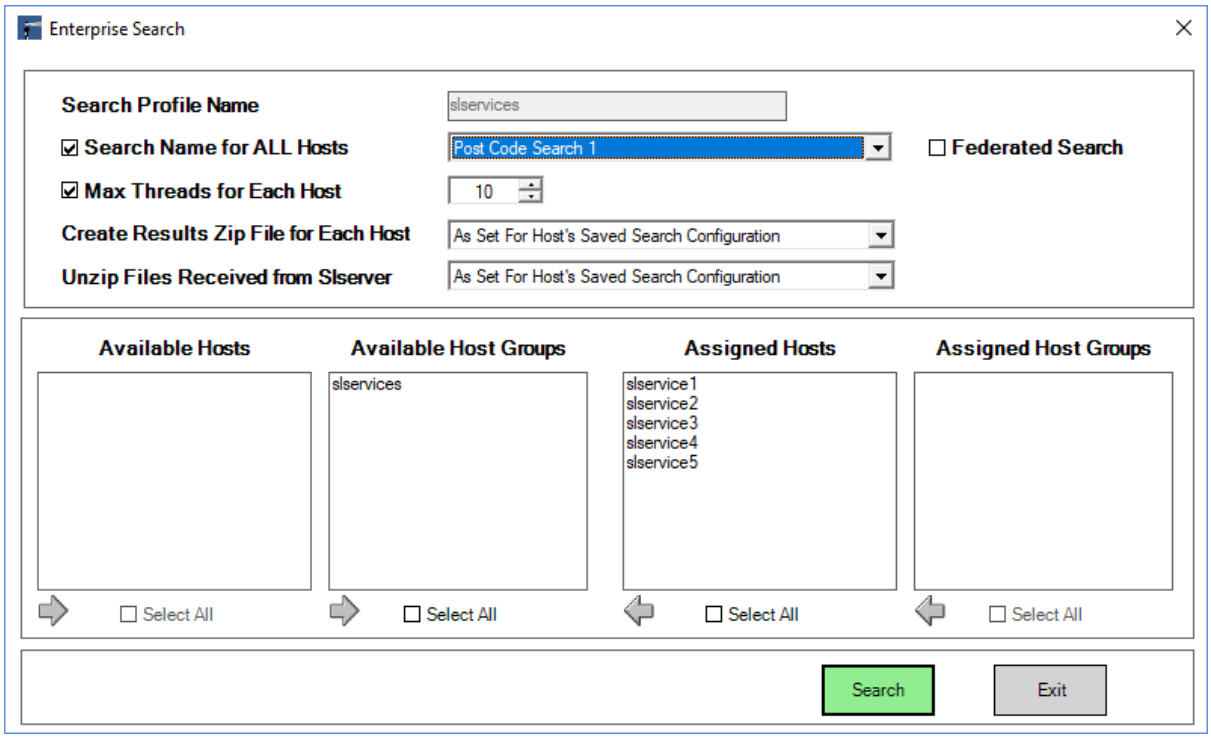
When the value of the 'What to Search For' dropdown box is either 'File of Searchwords' or 'Filenames (Filtered)' then the panel is disabled.

The 'Default' checkbox, displayed near the Search Name, indicates whether the displayed search definition is the default search definition or not.

The default search definition is the search definition that is automatically loaded on entry to the search screen. If the search definition displayed is not the default search definition then the button 'Make Default' is displayed and, if selected, will make the displayed search definition the default search definition.

8. Profile Search (Only available in Enterprise Edition)

A Profile search, accessed via the following menu options Search -> Enterprise Search, defines the parameters for a remote search and on being selected, the following screen is displayed:



Search profiles are stored in the file '...\files\savedsearchprofiles.ssf'.

In order to invoke a Profile search, one or more Hosts must have been defined and, if not already assigned, available hosts will be listed in the 'Available Hosts' listbox on the left hand side of the screen.

The fields of the Profile search are as follows:

Search Profile Name

This is a unique name and is used to identify a particular Search Profile.

Search Name for ALL hosts

If checked, this checkbox defines the name of the saved search that is to be used for all hosts participating in the search. If not checked, then each participating host will use the saved search that it has defined as part of its own definition. It should be noted that if the 'Federated Search' checkbox is checked then the 'Search Name for ALL hosts' checkbox will automatically be checked.

Max Threads for Each Host

If checked, this checkbox defines the maximum number of threads that each of the participating hosts can instantiate. If not checked, then each participating host will instantiate no more than the value of 'Max Threads' defined within its own definition.

It should be noted that the number of threads is limited to the value of 'max_threads' as defined in the licence.

Create Results ZipFile for Each Host

This setting determines whether a zipfile is created or not containing a copy of the found files. The value of this setting can be one of the following:

- As Set for Host's Saved Search Configuration

  This value will create a zipfile in accordance with the Saved Setting of the Host

- Yes – A zipfile will be created irrespective of the Saved Search setting for the Host

- No – A zipfile will not be created irrespective of the Saved Search setting for the host

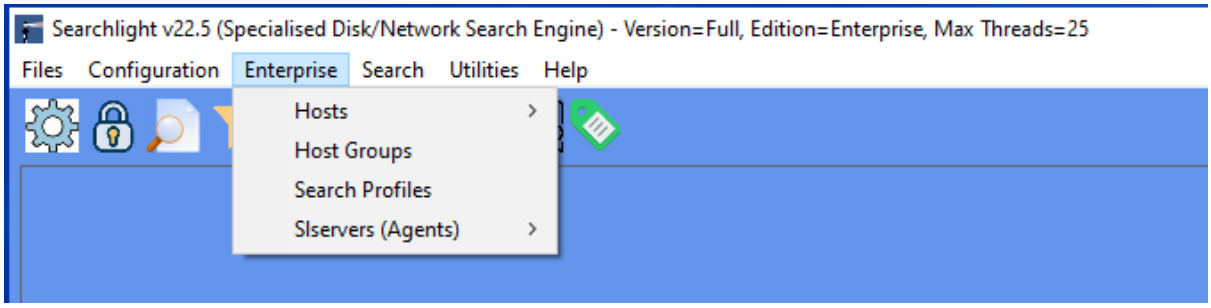
Federated Search

A Federated search is used to search a large data store by utilising the processing power of one or more remote hosts and when checked, all participating hosts must use the same search definition. It is important to remember that each participating remote host must have the target drive mapped to the same letter as that specified in the common search definition else the Federated search will not work.

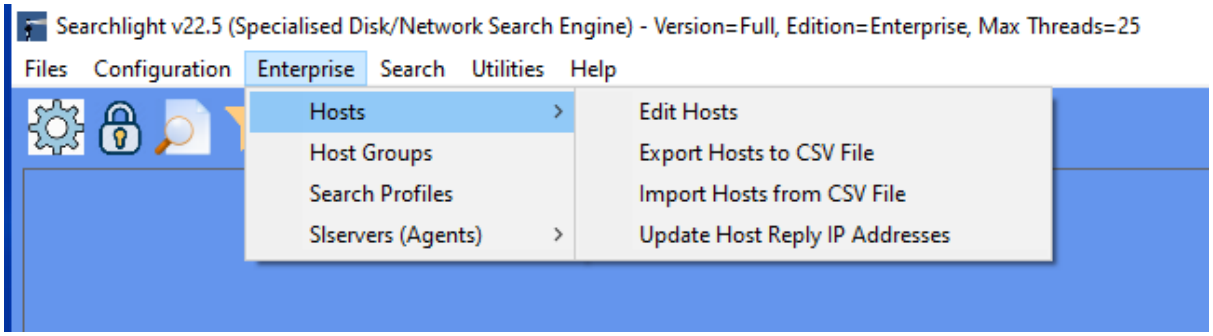
9. Remote Searches

Searchlight Enterprise can search the contents of files stored on remote hosts where 'slserver', as described below, has been installed.

The Enterprise Searchlight features are available via the 'Enterprise' menu as shown below:

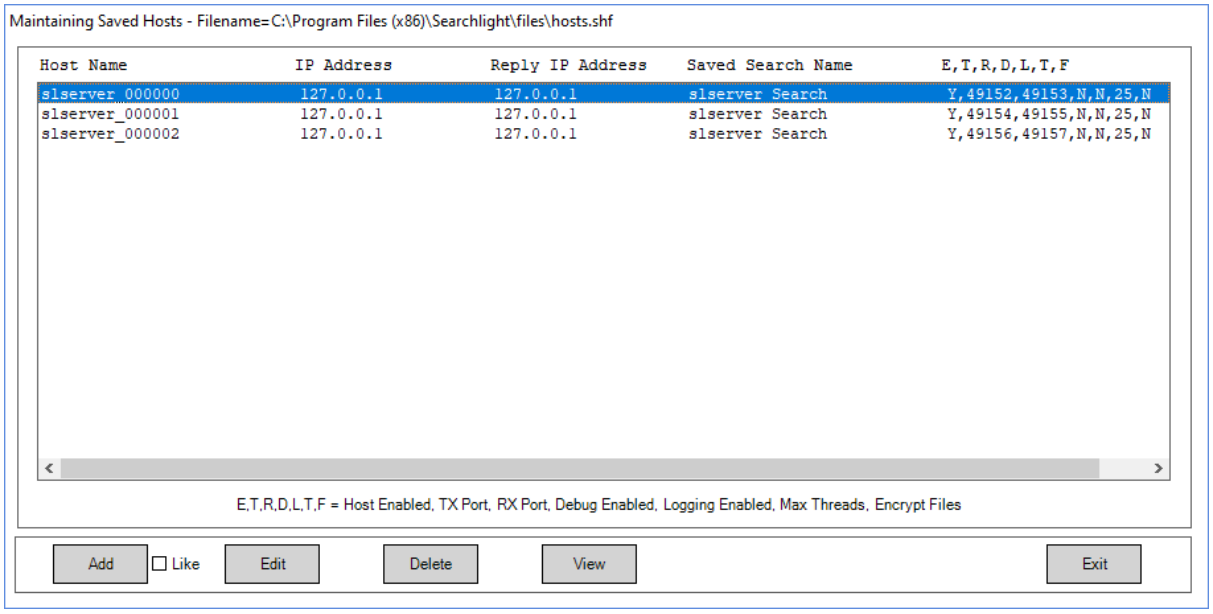


The 'Hosts' menu, has the following options as shown below:

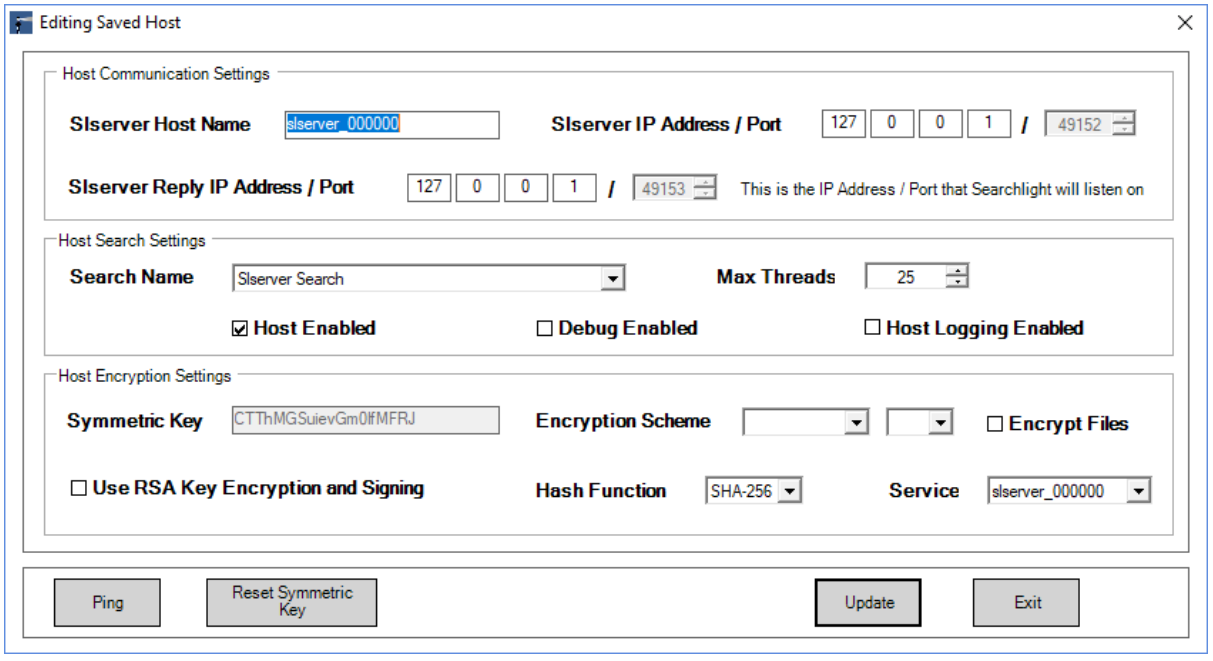


9.1 Edit Hosts

On selecting the Edit Hosts menu option the user is presented with a list of all of the configured Slservers as shown below:



In this example, it can be seen that 3 Slservers have been defined and on selecting one of these, slserver_000000 in this case, the following 'Edit Host' screen is displayed:



If should be noted that, in this example, the Slserver 'slserver_000000' is tied to the Service 'slserver_000000' as shown in the bottom right hand corner of the screen.

However, it should be noted that a Slserver Host definition can be linked to any Service that is available in the 'Service' drop down list.

The Service identifier may be blank. However, if the Service contains the name of a Slserver then the following values will be inherited by the Host being edited from the linked Service – slserver_000000 in this case:

The Port number	- 49152 in this case
The Reply port number	- 49153 in this case
The Symmetric Key	-CTThMGSuievGm0IfMFRJ in this case

The Host field descriptions are as follows:

Host Name

This is an arbitrary name that identifies this particular host and must be unique.

Host IP Address

This is the actual IP address of the remote slserver service

Host Reply IP address

This is the reply IP address that the remote slserver will use when replying to Searchlight/slsearch. It should be noted that when Searchlight is communicating to a remote slserver via a router then the Host Reply IP Address must be set to the address of the router.

Host Listen Port

This is the port number that Searchlight will communicate on to the remote slserver. It is important to remember that the value of 'port_number', stored in slserver's configuration file (.../files/slserver.ini), must match this value.

Host Transmit Port

This is the port number that slserver will communicate back to Searchlight/slsearch on. Thus, this is the port that Searchlight/slsearch will listen on when waiting for a reply from slserver.

Search Name

This is the name of the 'saved search' that will be executed by the remote slserver. Saved searches can be maintained via the following menu options:

Files -> Saved Searches -> Edit Saved Searches File

Max Threads

This is the maximum number of threads that this particular slserver will instantiate. This number is limited and cannot exceed the value of 'max_threads' as defined in Searchlight's/slsearch's licence file (.../files/searchlight.lic, .../files/slsearch.lic).

Host Enabled

This setting enables/disables a specific slserver and is particularly useful when a slserver is part of a group and is unavailable.

Debug Enabled

When this is checked additional/debug information is written to the Windows Event Log and is helpful when debugging the slserver and tracing program flow.

Host Logging Enabled

When this is checked, information is written to the Windows Event Log.

Symmetric Key

This is the encryption key that Searchlight/slsearch uses when encrypting over a communications link. In order for encrypted data to be successfully transferred, between Searchlight/slsearch and a slserver, the same symmetric key value must also be defined in the slserver configuration file (...files/slservice.ini). In order to protect privacy, the slserver.ini file must be accordingly protected using Windows permissions.

Encryption Scheme

This defines the type of encryption that is used between Searchlight/slsearch and a slserver.

Encrypt Files

If checked, this forces Searchlight/slsearch to write log files to disk in encrypted format. It should be noted that the encryption parameters used for encrypting log files are derived from Searchlight's/slsearch's configuration parameters (.../files/searchlight.ini, .../files/slsearch.ini) and not the encryption parameters defined for a slserver.

Use RSA Key Encryption and Signing

If checked, Searchlight/slsearch will generate random cryptographic keys for encrypting/decrypting data each time Searchlight/slsearch and slserver communicate.

The encryption keys are generated using public and private keys and provide perfect forward secrecy as the encryption keys are not reused.

Hash Function

This defines the hashing algorithm that is to be used when calculating the integrity of a message. On receipt of every message, Searchlight/slsearch and slserver all check the integrity of the message received by using the algorithm defined.

Service

A Slserver may, or may not, be tied to a specific Slserver specification.

When a Slserver is tied to a Service, such as slserver_000000 in this case, then the port numbers and the symmetric key of the tied Service will be used during any communication.

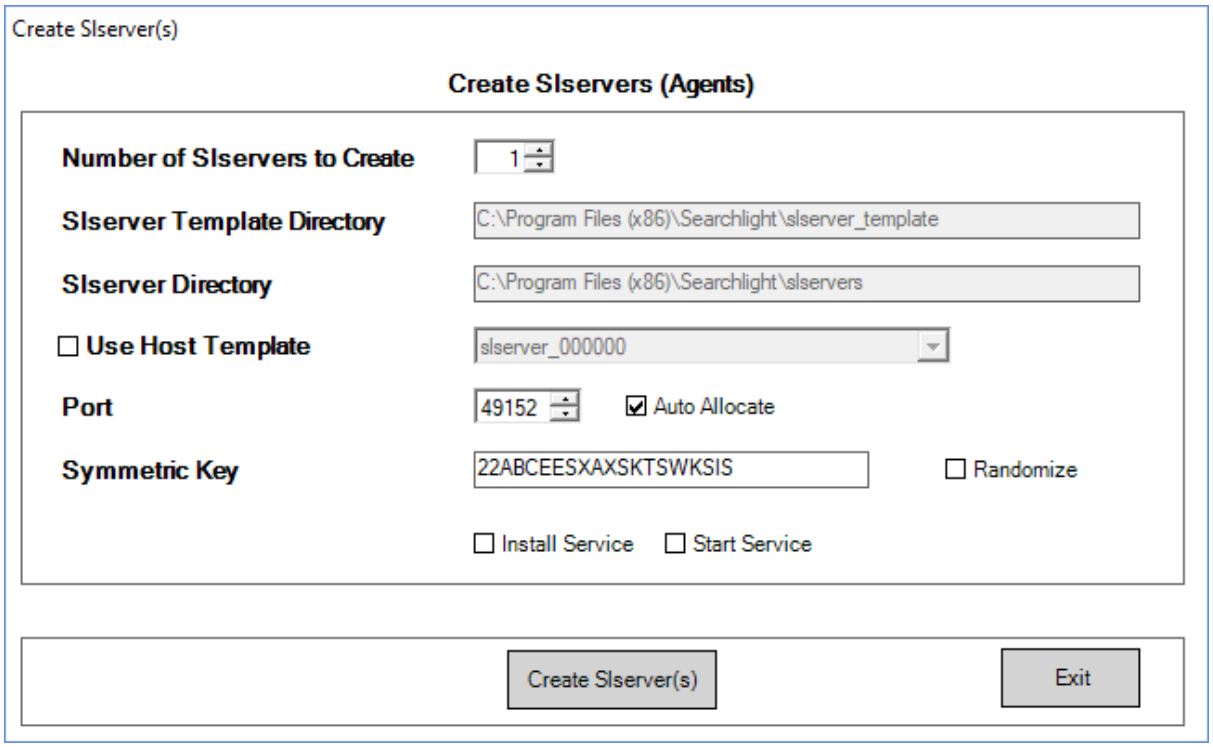
9.2 Creating Slservers/Hosts

The term 'Slservers' and 'Hosts' are closely related. A Slserver is the running Windows Service process whilst, in the context of Searchlight, a Host is the definition of the running Slserver process.

A Slserver is created, by selecting the menu options:

Enterprise -> Slservers (Agents) -> Create Slservers (Agents)

whereupon the following screen is displayed:



This dialog enables 1 or more Slservers to be automatically Created, Installed and Started.

If the 'Use Host Template' option is selected then the creation of the Slservers will be based upon the values defined for the Slserver specified as the Host Template – 'slserver_000000' in this case.

9.3 Deploying/Installing Slservers/Hosts

On creating a Slserver, Searchlight creates a directory and populates it with all of the files needed by the Slserver

As an example, on creating a Slserver named 'slserver_000003' the following directory would be automatically created (assuming Searchlight was installed into the default install directory):

c:\Program Files (x86)\searchlight\slservers\slserver_0000003

The Slserver identifier, 0000003 in this example, is automatically created as the next Slserver serial number following on from the Slserver with the largest serial number. If no Slservers exist then the serial 000000 is used.

It should be noted that all Slservers are created from the template Slserver files stored within directory:

c:\Program Files (x86)\searchlight\slserver_template

Once Searchlight has created the Slserver, it can be deployed to a remote machine as follows:

- Copy the created Slserver directory to the remote machine:

  c:\Program Files (x86)\searchlight\slservers\slserver_0000003

  The name of the above directory may be different

Install the Slserver Windows Service as follows:

- Logon as an administrator
- Change directory into the 'c:\<slserver_name>' directory (or equivalent)
- Run the script 'install_slserver.bat'

Start the Slserver Windows Service as follows:

- Logon as an administrator
- Change directory into the 'c:\<slserver_name>' directory (or equivalent)
- Run the script 'start_slserver.bat'

Note: The Slserver Windows Service can also be started via the Services dialog

Stop the Slserver Windows service as follows:

- Logon as an administrator
- Change directory into the 'c:\<slserver_name>' directory (or equivalent)
- Run the script 'stop_slserver.bat'

Note: The Slserver Windows Service can also be stopped via the Services dialog accessed via the Windows Control Panel.

Uninstall the slserver Windows service as follows:

- Logon as an administrator
- Change directory into the 'c:\<slserver_name>' directory (or equivalent)
- Run the script 'uninstall_slserver.bat'

Note: The Slserver Windows Service can also be uninstalled by the following command (run as an Administrator):

sc delete <slserver_name>

9.4 Coupling of Searchlight and Slservers

Searchlight is directly coupled to every Slserver it creates.

Searchlight's licence file (...files\searchlight.lic) contains a six digit value for key 'searchlight_identifier' and, on creating a Slserver, Searchlight embeds this value into the licence file of every Slserver it creates.

When communicating with a Slserver, Searchlight sends its own 'searchlight_identifier' value to the Slserver as part of the communication setup. On receipt of any communications, a Slserver process compares the value of 'searchlight_identifier' it receives to its own value for 'searchlight_identifier' derived from its own licence file. If

these two value do not match then Slserver replies to the caller with an error stating that a message was received from an invalid source.

Thus, an active remote Slserver will only perform searches requested from the Searchlight installation that created it.

9.5 The 'Slserver' Windows Service

The 'slserver' process is a Windows Service that, once installed and configured, listens for search requests, received via the network, from an installation of Searchlight. On receipt of a valid search request, the slserver process performs the search on its local machine and then replies back to the calling Searchlight process with the results.

Each slserver process has its own licence and will only communicate with the Searchlight process as defined by its licence file value 'searchlight_identifier'.

9.6 Slserver Configuration

Within each Slserver directory, such as 'c:\slserver_<999999>', the Slserver configuration file 'slserver.ini' is stored within the sub directory 'files'.

The main values defined within the slserver configuration file (slserver.ini) are as follows:

port_number=49200

This is the port number that the slserver service will listen on in order to receive search requests from Searchlight (or slsearch)

symmetric_key=hagWmTPHNi71pSxT4Avt

This is the key that slserver uses when communicating with Searchlight using either Triple Des or AES encryption.

write_configuration_to_event_log=yes

This defines whether, on start-up, the configuration values are written to the event log.

public_key_file=files\slserver_999999_public_key.xml

private_key_file=files\slserver_999999_private_key.xml

These are the public and private keys that the slserver uses to exchange symmetric key updates and to sign messages that it communicates back to Searchlight/slsearch.

searchlight_public_key_file=files\searchlight_999999_public_key.xml

This is the public key file of the specific Searchlight installation that slserver is tied to and is used to verify that any message received actually originated from slserver's associated Searchlight/slsearch.

ccpl_public_key_file=files\ccpl_public_key.xml

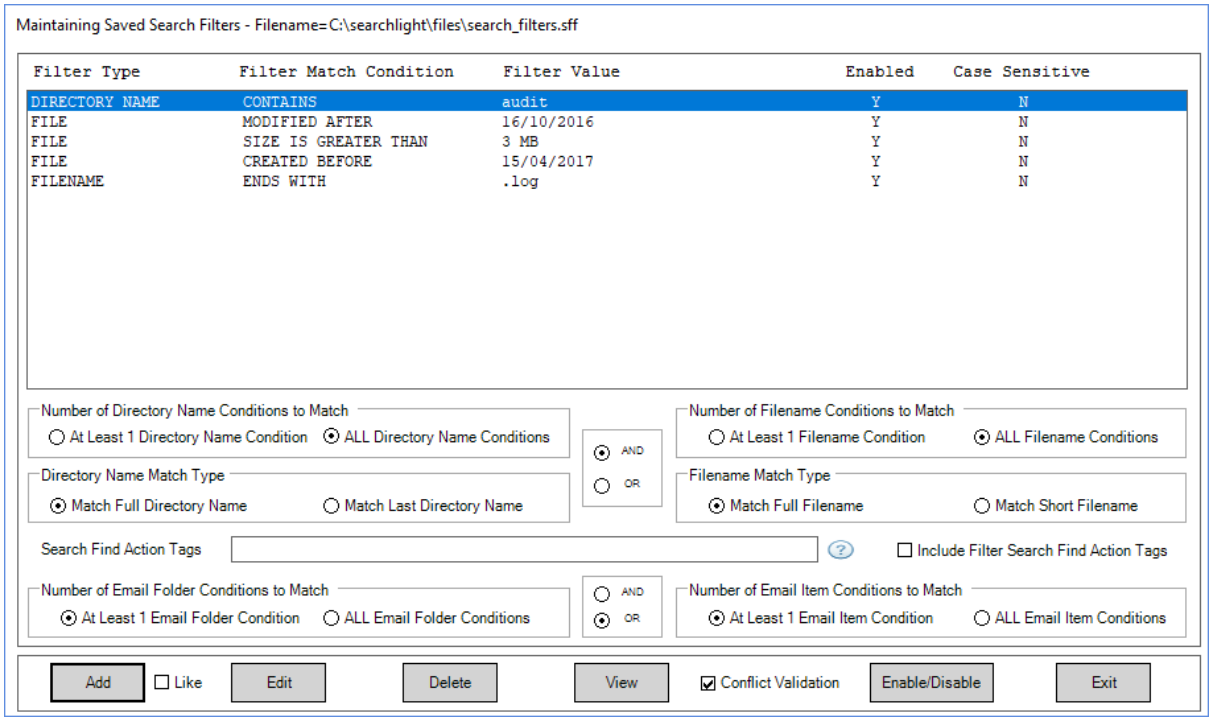
This is the public key file that is used to validate the licence file.

It should be noted that each slserver installation has its own private and public key files and the public key file of its associated Searchlight console.

10. Search Filters

Searchlight provides an extremely powerful file/directory filtering mechanism that can be used to target searches against very specific directories and files.

The search filter screen, as shown below, allows filters to be added, edited and deleted etc. but also defines how the filters, in combination with each other, are to be interpreted.



Filters are defined independently of each other but they collectively form a result that will either be true or false indicating whether a file has been found or not.

Filename filter conditions and File filter conditions are logically treated the same whilst Directory name conditions are treated separately.

As can be seen at the bottom of the screen above, a great deal of logic flexibility is provided to the user on how the filter condition is to be evaluated.

The And/Or radio button selection, between the Directory and File/Filename conditions, defines how the Directory and File/Filename conditions are to be logically joined. Thus, if the 'And' radio button is selected then the Directory name AND the File/Filename conditions must both be true in order for a file to be considered found.

However, it should be remembered that each side of the filter (Directory Name and File/Filename) is first determined independently before the join filter (And/Or) is determined.

The And/Or joining condition can also be applied to the Email matching conditions although it should be noted that these are not affected by the Directory/File And/Or conditions. If no Email Folder/Email Item conditions are defined then the Email matching conditions are ignored.

This provides the user with a great deal of capability in determining which file(s) to target in a search.

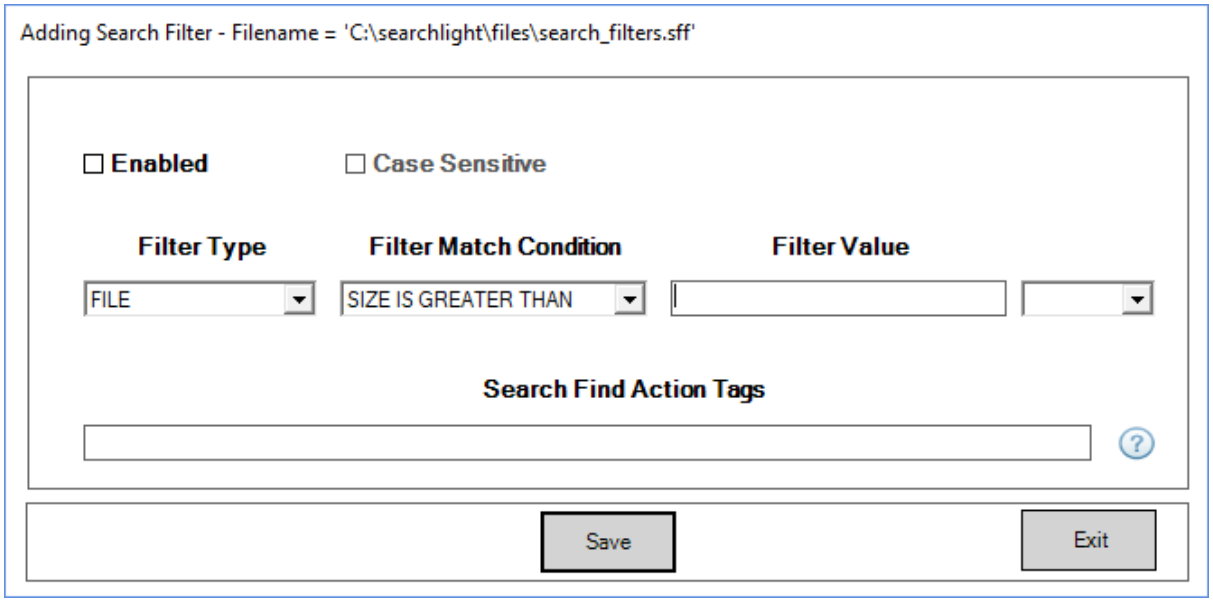
10.1 Maintaining Search Filters

Searchlight allows an unlimited number of Search Filter files to be created although only one Search Filter file can be used for a specific search.

When the menu options 'Files -> Search Filters -> Edit Search Filters' are selected, the user is prompted to select a Search Filters file.

Once a Search Filters file has been selected, Search Filters can be added, edited and deleted by selecting the appropriate button from the bottom of the 'Maintaining Saved Search Filters' dialog screen.

On selecting 'Add' from the 'Maintaining Saved Search Filters' dialog the following screen is displayed:



This 'Add' screen is similar to the one displayed when editing, deleting or viewing a Search Filter.

The 'Filter Type' dropdown listbox allows the following three types of filters to be defined:

- File

  The 'File' filter allows the following filter types to be applied to files:

  - Size is greater than - Bytes, Kb or MB
  - Size is less than - Bytes, Kb or MB
  - Created before - Date
  - Created after - Date
  - Modified before - Date
  - Modified after - Date
  - Created within ? days - Number of days
  - Modified within ? days - Number of days

  Case sensitivity, for the above filters, is disabled.

- Filename

The 'Filename' filter allows the following filter types to be applied to filenames:

- Starts with - Text value
- Does not start with - Text value
- Contains - Text value
- Does not contain - Text value
- Ends with - Text value
- Does not end with - Text value
- Equals - Text value
- Does not equal - Text value

Case sensitivity, for the above filters, is enabled.

- Directory Name

  The 'Directory Name' filter allows the following filter types to be applied to directory names:

  - Starts with - Text value
  - Does not start with - Text value
  - Contains - Text value
  - Does not contain - Text value
  - Ends with - Text value
  - Does not end with - Text value
  - Equals - Text value
  - Does not equal - Text value

  Case sensitivity, for the above filters, is enabled.

- Email Folder

  The 'Email Folder' filter allows the following filter types to be applied to any folder name found within a PST file:

  - Starts with - Text value
  - Does not start with - Text value
  - Contains - Text value
  - Does not contain - Text value
  - Ends with - Text value
  - Does not end with - Text value
  - Equals - Text value
  - Does not equal - Text value

  Case sensitivity, for the above filters, is enabled.

- Email Item

  The 'Email Item' filter allows the following filter types to be applied to an Email item found with a PST file:

  - Subject Equals - Text value
  - Subject Contains - Text value
  - Text Contains (Email Body) - Text value
  - Sent From (Email Address) - Text value
  - Sent By (Name) - Text value
  - Received Before - Date
  - Received After - Date

  Case sensitivity, for the above filters, is enabled.

Each filter can be enabled or disabled by checking or unchecking the 'Enabled' checkbox respectively. If a filter is disabled, by the 'Enabled' checkbox NOT being checked, then the filter is not loaded and not applied to a search.

The Search Filtering mechanism is extremely powerful and provides an easy and effective way of targeting specific files, or directories, according to a rich combination of filter types.

10.2 Filter Search Example:

If there was a requirement to only search files/directories, in accordance with the following criteria:

Filenames ending in '.log'

Files created before 15 April 2017

Files modified after 16 October 2016

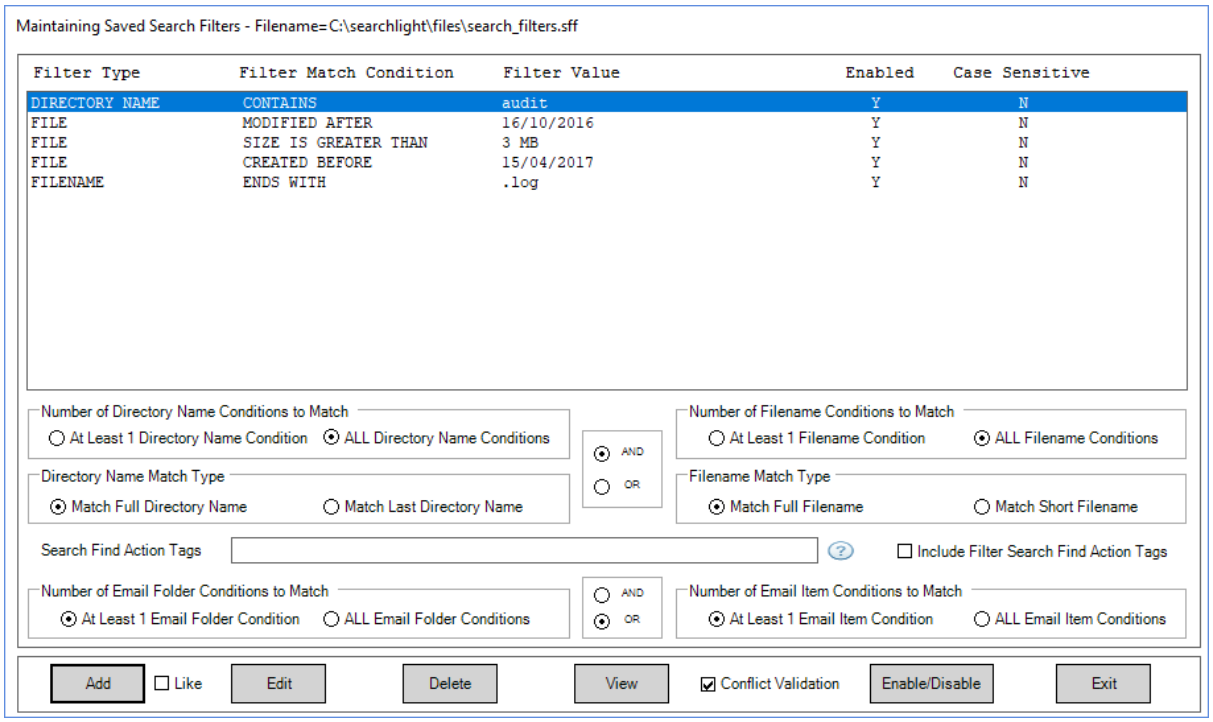
Files larger than 3Mb

Files that exist within a directory that has the text 'audit' in its name

then the following Search Filters could be defined (for example in a file called 'search_filters.sff'):

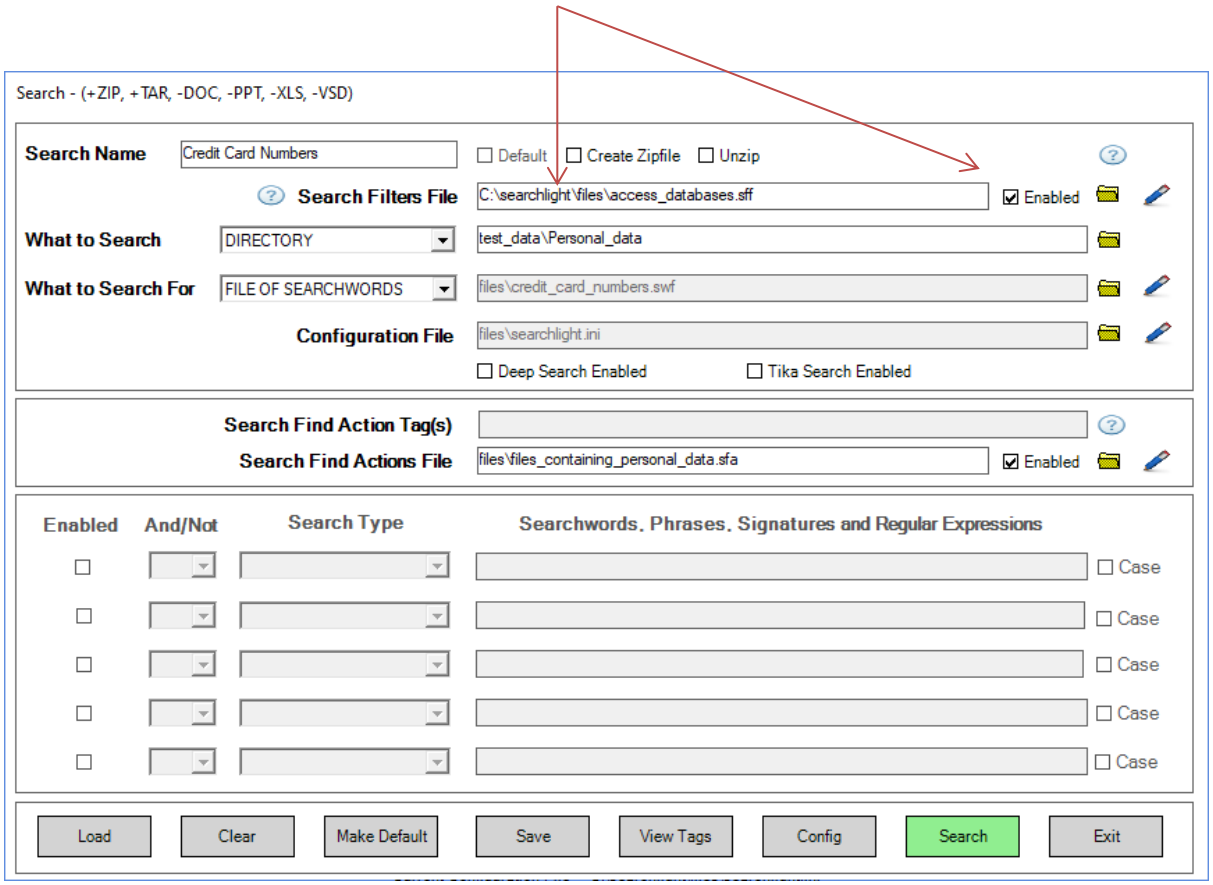
Filtering Requirement	Filter Type	Filter Match Condition	Filter Value
Filenames ending with '.log'	Filename	ENDS WITH	.log
Files created before 15/04/17	File	CREATED BEFORE	15/04/2017
Files modified after 16/10/16	File	MODIFIED AFTER	16/10/2016
Files larger than 3Mb	File	SIZE IS GREATER THAN	3Mb
Directory names that contain the word 'audit'	Directory Name	CONTAINS	audit

Once the search filters above are added to the 'search_filters.sff' file, the 'Maintaining Saved Search Filters' screen would look as follows:



10.3 Filtering Filenames in Combination with a File Content Search

File filtering can easily be applied to a file content search by enabling the search filter checkbox and defining a search filters file as shown below:



The above search definition will perform as follows:

- The directory 'test_data\personal_data', and all of its subdirectories, will be searched. When a full path name is not provided, the path is relative to the start directory (usually where searchlight.exe resides).

- All of the enabled filters in the file 'c:\searchlight\files\access_databases.sff' will be applied to the search

- The enabled Searchwords in the file 'files\credit_card_searchwords.swf' will be searched for

- On finding a searchword, the Search Find Actions, defined in the file 'files_containing_personal_data.sfa', will be actioned.

10.4 Filtering Filenames without a Content Search

Searchlight's powerful filtering mechanism can be used to find files that conform to a complex set of conditions, as already described, without the files' contents being searched.

This type of filename filter search is instigated by selecting the 'FILENAMES (FILTERED SEARCH)' from the 'What to Search For' dropdown on the main search screen.

On selecting this option, the system prompts for a File Filters file to be selected and it is the filters contained within this file that are used to filter the filenames during the search.

As mentioned, this type of search does NOT search the contents of files just their names and is therefore extremely fast.

When used in conjunction with the 'Create Zipfile' option, the 'filenames (filtered)' search allows a specific group of files to be targeted and packaged up into a zip file. For instance, all PDF files, older than a specified date or larger than a specified size, could be identified and neatly collected into a single zipfile.

Furthermore, if the zipfile configuration is appropriately configured, via the 'Zipfile Configuration' option on the Utility menu, then a script file could also be automatically created allowing a user action, such as copy or delete, to be applied to all of the identified PDF files.

11. Search Find Actions (Only available in Professional/Enterprise Editions)

One of the more specialised features of Searchlight is known as 'Search Find Actions' and is often referred to as SFAs.

SFAs, are stored in files of type '*.sfa', and are a set of selected actions that can be optionally performed by Searchlight, on a file that has been found as part of a search.

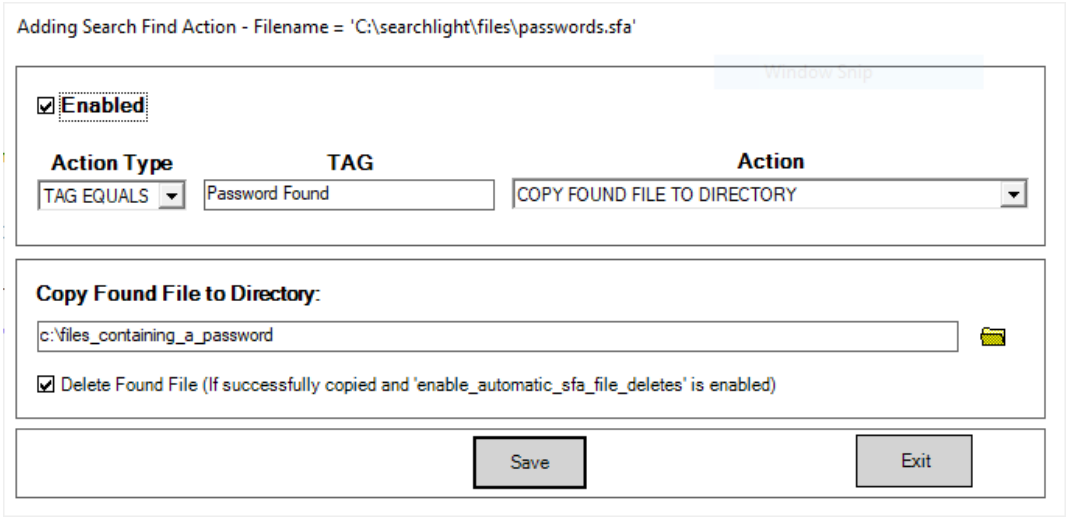
Each Searchword definition has a field 'Search Find Action Tags:' associated with it. This field can contain none, one or multiple Search Find Action Tags separated by a comma. The 'Search Find Action Tags:' field, defined as part of the Searchword definition, acts as a link between the Searchword and any Search Find Action to be executed. When a file is found as part of a search, the Searchword's Search Find Action Tag(s) are written to the results file which is then read by the Search Find Actions Engine and any Search Find Action Tags associated with a found file are processed accordingly.

The Search Find Action definition itself can only be one of the following two types:

TAG EQUALS and DO.

- TAG EQUALS

  This tells the SFA Processing Engine that if a search result contains a Search Find Action Tag equal to the one defined, in the SFA definition, then the associated 'Search Find Action', defined within the definition, must be performed. Thus, as an example:



The screen above shows a single Search Find Action definition instructing the SFA Processing Engine to do the following:

> If the results of a search contains a Search Find Action Tag equal to 'Password Found' (no case sensitivity) then the found file will be copied to the directory 'c:\files_containing_a_password'. Optionally, as in this example, the found file may be deleted but the deletion depends upon the conditions as described below.

A found file will only be automatically deleted if the 'Delete Found File' checkbox is checked and the configuration key 'enable_automatic_sfa_file_deletes' is set to 'yes'.

If the 'Delete Found File' checkbox is checked but the configuration key 'enable_automatic_sfa_file_deletes' is set to 'no' then the file will not be automatically deleted. However, in this case, Searchlight will write a MSDOS command to delete the file to a batch file named: '<install_directory>\scripts\<timestamp>_delete_files.bat'.

This is the default behaviour and gives the user an option to review file deletes before they are executed.

A user must check the 'Delete Found File' checkbox AND explicitly set the configuration key 'enable_automatic_sfa_file_deletes' to 'yes' in order to enable the automatic deletion of found files by the SFA Engine.

Search results may contain multiple Search Find Action Tags and therefore multiple actions can be performed upon a single found file.

- DO

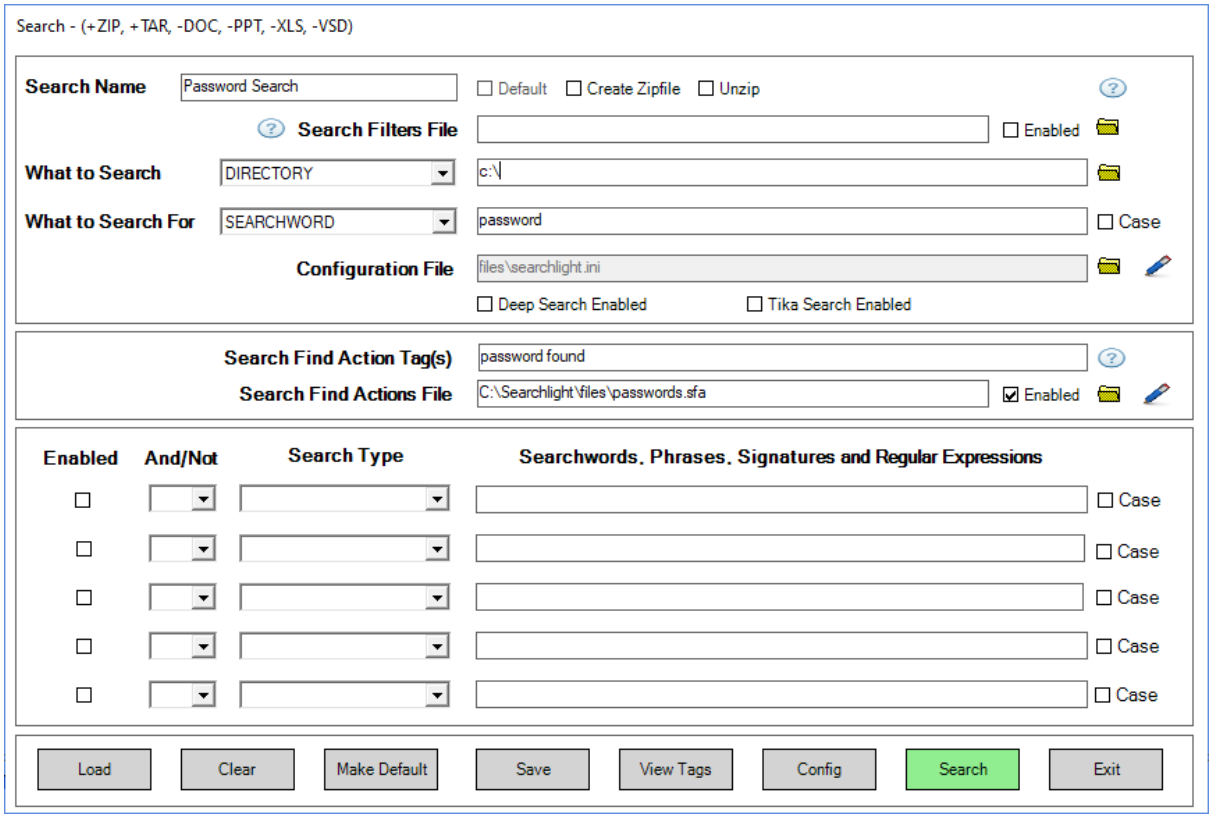
  The SFA Processing Engine will perform all of the 'DO' actions, defined in the search's associated Search Find Actions file (*.sfa), upon any file that is found as part of a search. Thus, the 'DO' action type does not require any TAG to be set or matched, it simply requires a file to be found. This command is particularly useful when combined with the Search Filters that can also have a Search Find Action definition associated with them.

  Thus, for example, a filter could be set to only find filenames that end in '.pdf' and a 'DO' Action Type could be defined instructing the SFA Processing Engine to copy any found file, matching the filter, to a directory called 'c:\pdf_files'.

  The combining of filters with Search Find Actions is very powerful and is very efficient because the Search Engine will not attempt to process the contents of a file, it will simply tag the file as being found if it passes the filter conditions.

11.1 Search Screen

Search Find Actions can be enabled via the 'Search Screen' as shown below:



The Search Screen above shows the following:

- The c:\ drive specified as the directory to be searched
- A Searchword/Phrase of 'password'
- A single Search Find Action Tag of 'password found'
- A Search Find Actions File of 'c:\searchlight\files\passwords.sfa' being associated with the search (and enabled).

Thus, when this search is performed, if the string 'password' (no case sensitivity specified) is found, then the search result will be tagged with the SFA tag 'password found'. Then, on completion of the search, the SFA Processing Engine will perform the

actions contained within the search's associated SFA file 'c:\searchlight\files\passwords.sfa'.

Only one Search Find Actions file (*.sfa), containing none, 1 or more Search Find Action definitions, can be associated with a particular 'Saved Search'.

The Search Find Action Tags provide a link between a found file and one or more associated Search Find Actions defined in the Search Find Actions file associated with a saved search definition.

If the Search Find Action Tags field contains multiple Search Find Action Tags then they must each be separated with a comma.

The saved search definition above will:

- Instruct Searchlight to search the files in the directory 'c:\', and all of its subdirectories, for the string 'password' (without any case sensitivity).
- If a file is found, to contain the string 'password', then the Search Find Action Tag 'password found' will be written to the results file.
- On completion of a search, if Search Find Actions are enabled, the Search Find Actions Processing Engine will process the results file as follows:
  - All of the found filenames, along with their associated Search Find Action Tag(s), if any, are extracted from the results file.
  - The Search Find Actions Engine then attempts to match the Search Find Action Tags, extracted from the results file, to a Search Find Action tag defined within the search's associated Search Find Actions file which, in this example, is 'c:\searchlight\files\passwords.sfa'.

- If a Search Find Action tag from the results file matches a Search Find Action tag in the search's associated Search Find Actions file then Search Find Action associated with Search Find Action Tag is executed.

12. Search Tags

Search Tags are a unique feature of Searchlight and provide a very efficient and simplistic way of encapsulating multiple search criteria into a single search definition.

Many search programs offer 'regular expressions' to achieve the same functionality provided by Search Tags. However, regular expressions, for many people, are difficult to use and, as such, they tend to be only used by those that are more technically minded or can take the time to learn the complicated syntax that they require.

A 'Search Tag', in Searchlight terms, is a numerical value that is prefixed and suffixed by the defined configuration values 'complex_searchword_prefix' and 'complex_searchword_suffix' which, by default, are set to the values '<<' and '>>' respectively (e.g. <<1>>).

12.1 Search Tag Example

Assume that the requirement is to search for the following phrase (and combination of other phrases as described below):

"A red Ford Escort with two doors"

Also assume that:

"red" could be replaced with any of the following:

black, green, yellow, blue, grey, silver and white

"Ford Escort" could be replaced with any of the following:

Honda Civic, Toyota, Land Rover, Nissan, Kia, Volvo

"two doors" could be replaced with "four doors"

If the above search criteria was expanded out to include every combination then 112 search phrases would have to be defined.

However, by defining the following 3 Search Tag definitions:

Search Tag ID	Search Tag Definitions
1	red,black,green,yellow,blue,grey,silver,white
2	Ford Escort,Honda Civic,Toyota,Land Rover,Nissan,Kia,Volvo
3	two doors,four doors

the following Search phrase could be used in order to match any of the 112 possible search phrases:

A <<1>> <<2>> with <<3>>

Search Tags provide an extremely efficient and powerful mechanism that can dramatically reduce the number search definitions required and are a unique feature of Searchlight.

All Search Tag definitions are stored in the file 'searchtags.stf' which, by default, is stored in the '/files' directory.

It should be noted that Search Tag definitions can be part of the main searchword text box, in the search screen top panel, as well as within any of the compound searchword text boxes in the search screen's middle panel and can appear an unlimited number of times.

13. Searchlight Trial Version

The 'Trial Version' of Searchlight enables remote searches to be simulated without having to install any slservers (Windows Services) on remote machines.

By default, the Trial version of Searchlight Professional is installed with five preconfigured slservices installed under the directory 'slservices'.

It should be noted that 'Slservices' are locally running processes that simulate the remote running of 'Slservers' which are Windows Services installed on remote machines that communicate with Searchlight.

13.1 Starting Slservices

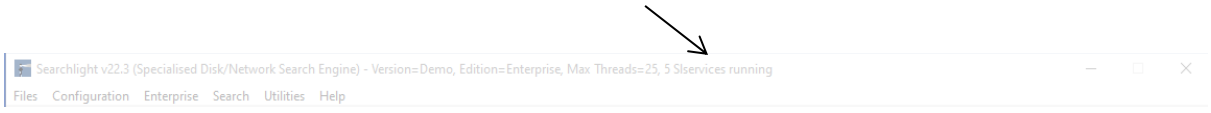
Slservices must be started before any simulated remote (Enterprise/Profile) searches can be invoked.

Slservices are started by selecting the following menu options:

Utilities -> Slservices -> Start All Slservices

Each Slservice runs with its own unique configuration as an individual process called 'Slservice'.

The number of Slservices actually running is displayed as part of the Searchlight window banner:

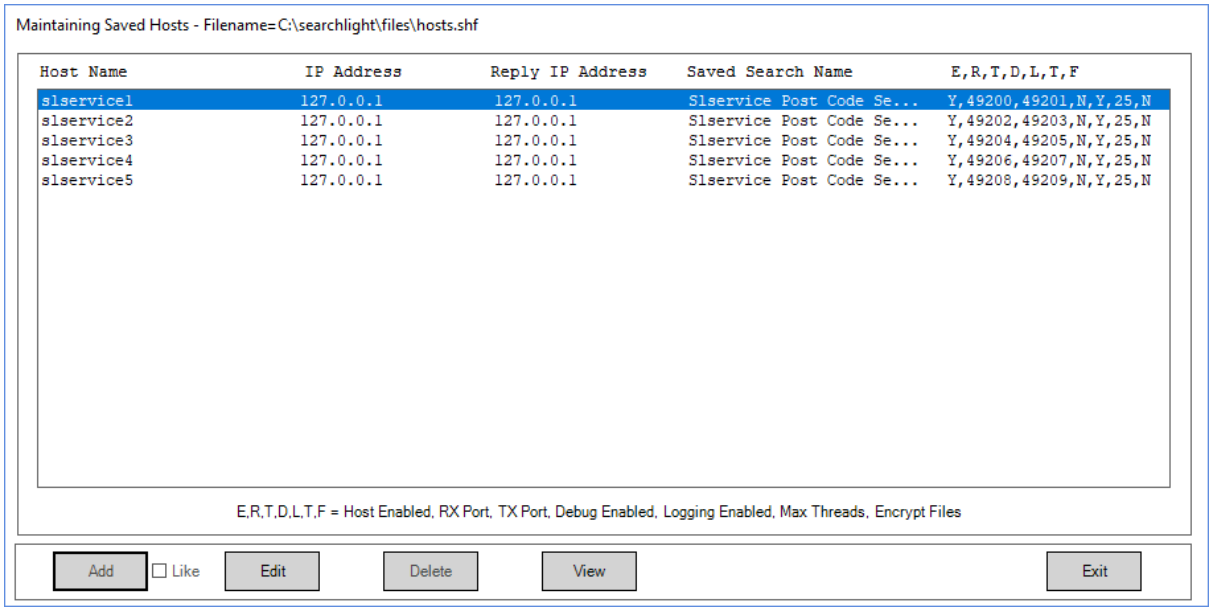


13.2 Slservice Configuration

Each Slservice has its own unique configuration which can be accessed by selecting the following menu options:

Enterprise -> Hosts -> Edit Hosts

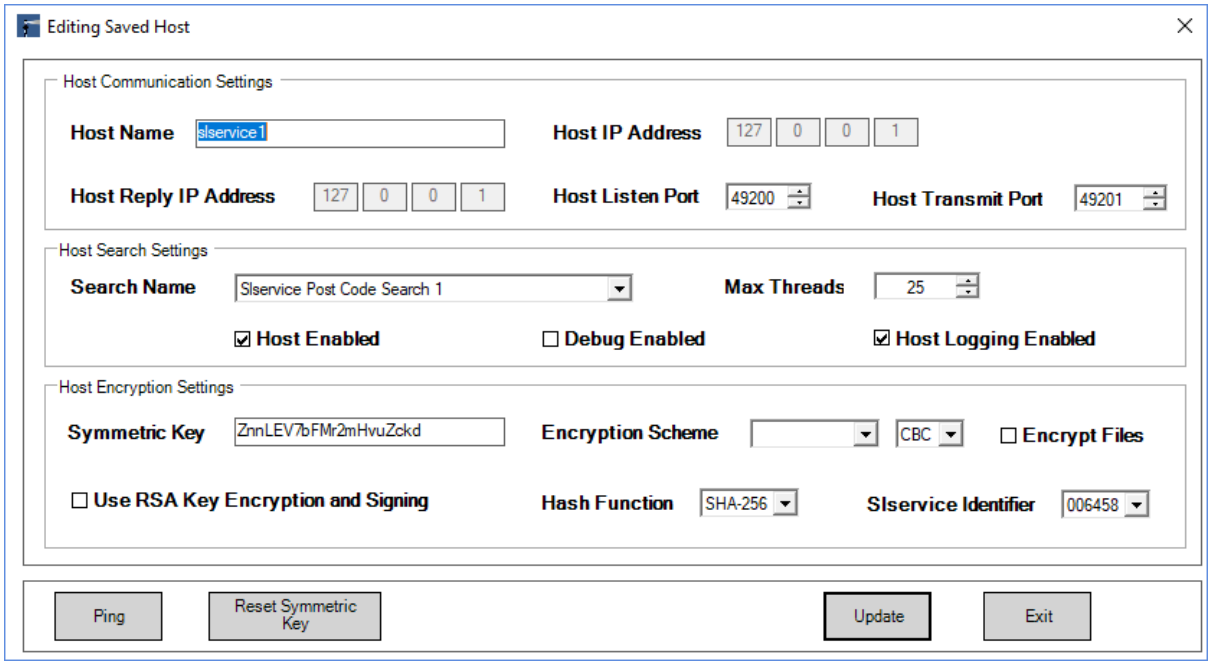
On selecting the above menu options the following screen is displayed whereupon one of the Slservices can be selected for maintenance:



It can be seen that each Slservice is configured to transmit and receive on the loopback IP address 127.0.0.1 and that each Slservice listens and transmits on a unique pair of port numbers.

The 'Add' and 'Delete' buttons and the 'Like' checkbox are all disabled as for demonstration purposes the number of Slservices is fixed.

On selecting one of the Slservices from the screen above the following screen is displayed:



The IP addresses are disabled as the Trial version always uses the loopback IP address of 127.0.0.1.

Although the IP address is restricted to the loopback address, every other option is enabled and operates exactly the same as if the Slservice was communicating to a remote machine with a specific non loopback IP address (simulating an slserver).

For more information about how an Enterprise search is invoked, select the option 'Profile Search' from the Help menu.

14. Searchlight Directories

The following is a list of the directories used by Searchlight:

14.1 files

This is where Searchlight stores the majority of its files including:

- searchlight.lic - Licence file
- searchlight.ini - Initialisation file
- searchlight_admin.ini - Admin initialisation file

...

and many other files as described in the 'Searchlight Files' help file.

14.2 log_archives

This is, by default, where Searchlight automatically copies log files, if log archiving is enabled as described in the 'Log Archiving' help file.

14.3 public_keys

This is where Searchlight stores the public keys of any slservers that it is licensed to communicate with.

It should be noted that Custom Computer Program's Ltd public key 'ccl_public_key.xml' is stored in the 'files' directory.

14.4 test_data

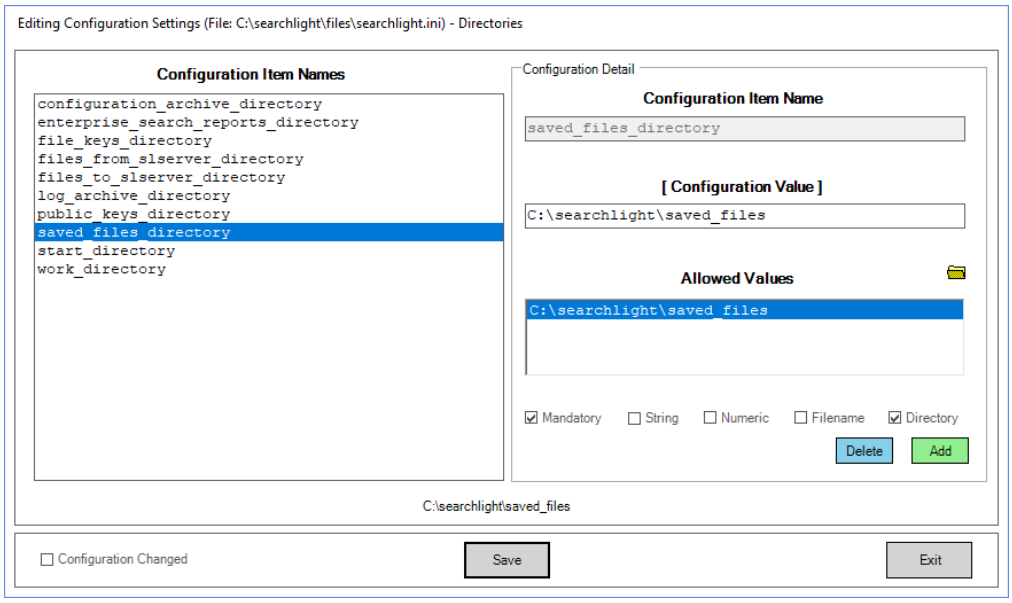
This directory contains a number of publicly available files containing Post Code data. These files are automatically installed and can be used to test Searchlight's search capability. When Searchlight is installed, a number of pre-set search definitions are configured to search the data stored in this directory.

14.5 saved_files

On completion of a search, Searchlight allows the results files to be saved and this is, by default, the directory that Searchlight presents.

This directory is defined by the configuration option 'saved_files_directory' and can be amended by following the menu options: (whereupon the screen below will be displayed):

Configuration -> Edit Current Configuration -> Directories



14.6 enterprise_search_reports

This directory is where Searchlight writes the log files that it receives back from slservers that participated in an Enterprise/Profile or Host Search.

This directory is defined by the configuration option:

'enterprise_search_reports_directory'

The 'enterprise_search_reports_directory' configuration setting is edited in the same way as that described for editing the 'saved_files_directory' configuration item above.

14.7 work

This directory is where Searchlight writes all of its work files when performing a search. At the start of each search, Searchlight deletes the 'work' directory, and all of its sub-directories, and then recreates it.

The 'work' directory should only ever contain the following three sub-directories:

- found_files
- found_work_files
- temp

If, when attempting to delete the 'work' directory, Searchlight detects any file or directory, other than the three listed above, then the search is halted and manual intervention is required to delete/move the offending file/directory.

This is a safety mechanism build into Searchlight to ensure that when the 'work' directory is deleted only recognised files/directories are deleted.

15. Searchlight Files

Searchlight uses the following simple text files:

15.1 ccl_public_key.xml

This is Custom Computer Program's Limited public key file and it is used to validate the licence file and '*.ssi' files.

15.2 hostgroups.shg

This file stores an unlimited number of host groups. A host group can consist of any number of hosts and allows hosts to be selectively grouped together and subsequently searched as a group when executing a profile search.

15.3 hosts.shf

This file contains the definitions of all of the hosts that are typically remotely searched.

15.4 multisearches.msf

This file contains multi search definitions where multiple searches can be grouped together and executed serially as a single search.

15.5 savedsearches.ssf

This file is the default saved search file and contains an unlimited number of search definitions. Searchlight allows multiple saved search files to be created and used.

15.6 savedsearchprofiles.ssp

This file contains an unlimited number of the profile search definitions which define the parameters of a remote search.

15.7 search_filters.sff

Search Filter files contain an unlimited number of Search Filters which can be individually enabled or disabled as required. In addition, Searchlight allows multiple search filter files to be created although only one Search Filter file can be associated with any one running search.

15.8 search_timestamps.txt

This file is used to record the timestamps of individual messages that are received from slservers. The purpose of this file is to ensure that a duplicate message can be detected and thus help prevent a malicious program attempting to replay a message that has already been sent. This mechanism is also used by the slserver programs to detect any duplicate message that they may receive from Searchlight.

15.9 searchlight.bin

This file is used to record the details, as described below, about the flow of messages between Searchlight and its associated slservers:

Timestamps of messages transmitted

IP addresses of the message source and destination

The 'searchlight_identifier' value (as defined within Searchlight's licence file)

The 'slserver_identifier' value (as defined within a valid '*.ssi' file)

15.10 searchlight.ini

This file contains Searchlight's configuration that is read at start-up and overrides any default value that is set internally within Searchlight.

15.11 searchlight.lic

This is a simple text file that defines Searchlight's licence.

15.12 searchlight_999999_private_key.xml

This is the private key file that is unique and specific to the installed copy of Searchlight. This file is used to sign messages that are sent to remote slservers and to sign log files when configured to do so.

15.13 searchlight_999999_public_key.xml

This is the public key file that is unique and specific to the installed copy of Searchlight and is the counterpart to the private key file described above. This file is typically installed with each slserver process so that it can validate files signed using Searchlight's private key file.

15.14 searchlight_999999_searchlightlog.srf

This file is produced each time Searchlight runs, overwriting the previous copy, and is the search results log file. If there is a requirement to save results files, rather than them being overwritten each time Searchlight runs, then 'Log Archiving' should be enabled.

15.15 searchlight_admin.ini

This is the configuration file that is read after the 'searchlight.ini' configuration file is read and overwrites any configuration item read from the 'searchlight.ini' with the same name. This is a security mechanism and provides administrators with a mechanism for ensuring that certain configuration values are not changed (assuming that the 'searchlight_admin.ini' file is appropriately secured using Windows permissions).

15.16 searchtags.stf

This file stores all of Searchlight's Search Tag definitions.

15.17 searchwords.swf

This type of file is used to store multiple searchword definitions and is used when a list of multiple searchwords are to be searched as part of a single search. There can be an unlimited number of these files created and each one of these files can contain an unlimited number of searchword definitions.

15.18 slservices_999999.ssi

This file contains one or more 'slserver_identifiers' and is used to associate an installation of Searchlight with one or more slservers. This file is signed using Custom Computer Program's private key and should not be edited.

15.19 tika_excluded_search_file_types.ftf

Searchlight has the ability to use the Apache Open Source program Tika in order to extract text from literally hundreds of file types. However, by default, if Tika file processing is enabled, Tika will attempt to extract text from every file type which is not always desired.

For instance, it wouldn't make any sense for Tika to try and extract the text from an executable program (*.exe). Thus, by placing the '.exe' file extension in the 'tika_excluded_search_file_types.ftf' file, Searchlight will ensure that Tika does not attempt to extract text from such files.

File extensions must be entered one per line.

16. Test Data

Searchlight is a powerful search tool providing many advanced and unique features such as Advanced File Filtering, Search Find Actions and Found File Utilities, all working in collaboration with a loosely coupled set of highly configurable configuration files.

However, it should be noted that the Search Find Actions feature is only available in the Professional Edition.

When Searchlight is installed, the following two configuration archive directories are installed:

<drive>:\searchlight\configuration_archives\files_00000000000000

<drive>:\searchlight\configuration_archives\test_data_00000000000000

At any time, Searchlight's active configuration can be saved into the configuration archive by selecting the following menu options:

Configuration -> 'Save Current Configuration'

Conversely, Searchlight's configuration can be loaded from a saved configuration fileset by selecting the menu option Configuration -> Load Archive Configuration

Archived configurations can be deleted by selecting the following menu options:

Configuration -> 'Delete Archive Configuration'

However, it should be noted that the following saved archives cannot be deleted:

<drive>:\searchlight\configuration_archives\files_00000000000000

<drive>:\searchlight\configuration_archives\test_data_00000000000000

16.1 Search Find Action Files (*.sfa)

These files are configured to demonstrate the very powerful 'Search Find Actions' functionality (see the Help File 'Search Find Actions'). For instance, the 'file_sort.sfa' file shows how files that match a certain rule (such as having a file extension of '.mdb') can be automatically copied to a nominated directory with optional deletion of the found file.

- classified_files.sfa
- file_sort.sfa
- files_containing_personal_data.sfa
- places_of_interest.sfa
- test.sfa

16.2 Search Filter Files (*.sff)

These files are configured to define 1 or more filter rules in order to filter out specified file types during a search.

For example, the 'access_databases.sff' file has the following filename filters defined:

- Filename ends with '.mdb'
- Filename ends with '.mdbx'
- Filename ends with 'accdb'

In addition, the filename matching rule has been defined as 'Match One Filename Condition' which means that a filename will pass the filter if any of the rules match (this is an OR condition).

- access_databases.sff
- excel_spreadsheets.sff
- file_sort.sff
- pdf_files.sff
- powerpoint_files.sff

- search_filters.sff
- visio_files.sff
- word_documents.sff

16.3 Saved Searches File (*.ssf)

These files contain saved search definitions.

- savedsearches.ssf - Contains all of the pre-set saved search definitions.

16.4 Searchword Files (*.swf)

These files define one or more Searchwords/Phrases/Byte Sequences that are used to match the content of a file. These examples show how a number of searchword types, such as Regular Expressions, Lists of Searchwords and Searchword Tags, can be utilised to identify files.

- bank_account_numbers.swf
- classifications.swf
- credit_card_searchwords.swf
- dates_of_birth.swf
- email_addresses.swf
- places_of_interest.swf
- searchwords.swf (default)
- strap_files.swf
- telephone_numbers.swf

17. Test Data

The Searchlight directory '<drive>:\searchlight\test_data', installed by default, is preconfigured with the following subdirectories and files:

17.1 binary_data

- National_postcodes_10.txt - Used to demonstrate the Binary Search

This file is not a binary file in the true sense. It is actually a plain text file which has been used to demonstrate the binary searching capability.

17.2 classified_files

This directory is populated with pseudo classified files when the Search Find Action file 'classified_files.sfa' is executed/enabled as part of the 'Classified files' saved search. In order to simulate a classified file, classifications, such as Restricted, Confidential and Secret, have been embedded in a number of National Postcode data files.

17.3 file_types

These directories are used by the Search Find Action definitions, defined as part of the Advanced Pre-Set configuration as previously described.

- access_files
- bitmap_files
- csv_files
- excel_files
- jpeg_files
- pdf_files
- png_files
- powerpoint_files
- text_files
- tif_files
- visio_files
- word_files

- zip_files

For instance, the saved search 'File Sort' has been configured, via a Search Find Action, to copy any found jpeg files to the following directory:

<drive>:\searchlight\test_data\file_types\jpeg_files'

Thus, the above directory must pre-exist in order for the Search Find Action to work.

It should be noted that the 'File Sort' saved search, defined within the saved searches file 'savedsearches.ssf', searches the directory '<drive>:\searchlight\test_data\test_files' and does the following:

- Filters each file against the filters defined in the file 'file_sort.sff'.
- Filters each file against the Search Find Action definitions in the file 'file_sort.sfa'.

In this case, the file filtering is applied at two levels. The first level of filtering is applied by the 'file_sort.sff' file and the second level is applied as part of the Search Find Action definitions. In this case, the first level of filtering could be disabled but it has been included in order to show that two levels of file filtering can be applied.

17.4 files_containing_personal_data

This directory is populated by the following saved search definitions:

- Bank Account Numbers
- Credit Card Numbers
- Dates of Birth
- Email Addresses
- Telephone Numbers

17.5 personal_data

- National_postcodes_[01..25].txt

These files are a copy of the post code data files held in directory '<drive>:\searchlight\test_data\post_code_data'.

The following files, in the 'personal_data' directory, have been modified to include some fictitious personal data within them, such as credit card numbers, telephone numbers and dates of birth etc.

- National_postcodes_05.txt
- National_postcodes_07.txt
- National_postcodes_10.txt
- National_postcodes_15.txt
- National_postcodes_20.txt
- National_postcodes_25.txt

17.6 places_of_interest

- Birmingham
- Bolsover
- Huntingdon
- Peterborough

These directories are populated by the Search Find Actions defined in the saved search definition 'Places of Interest'.

17.7 post_code_data

- National_postcodes_[01..25].txt

This directory is searched by the following saved searches in order to demonstrate a number of different saved search definitions.

- Classified Files

- List of Searchwords
- Password
- Places of Interest
- Post Code Search 1
- Post Code Search 2
- Regular Expression Search
- Strap Files
- Tag Search

17.8 test_files

This directory contains a number of text files, all based on the National Postcode data files, with their file extensions changed.

It is not possible to provide genuine copies of all of the file types listed below so in order to demonstrate Searchlight's very powerful filtering and 'Search Find Actions' capability, the National Postcode data files have simply been copied, into this directory, and had their file extensions modified to simulate different file types.

This capability is demonstrated by the 'File Sort' saved search and demonstrates how filenames can be searched, moved, or copied into a specified directory.

- National_postcodes_01.ppt
- National_postcodes_02.pptx

- National_postcodes_03.doc
- National_postcodes_04.docx

- National_postcodes_05.xls
- National_postcodes_06.xlsx

- National_postcodes_07.pdf
- National_postcodes_08.pdf

- National_postcodes_09.jpg

- National_postcodes_10.jpeg

- National_postcodes_11.accdb
- National_postcodes_12.mdb

- National_postcodes_13.csv
- National_postcodes_14.csv

- National_postcodes_15.vsdx
- National_postcodes_16.vsd

- National_postcodes_17.zip
- National_postcodes_18.zip

- National_postcodes_19.tif
- National_postcodes_20.bmp
- National_postcodes_21.png

- National_postcodes_22.txt
- National_postcodes_23.txt
- National_postcodes_24.txt
- National_postcodes_25.txt

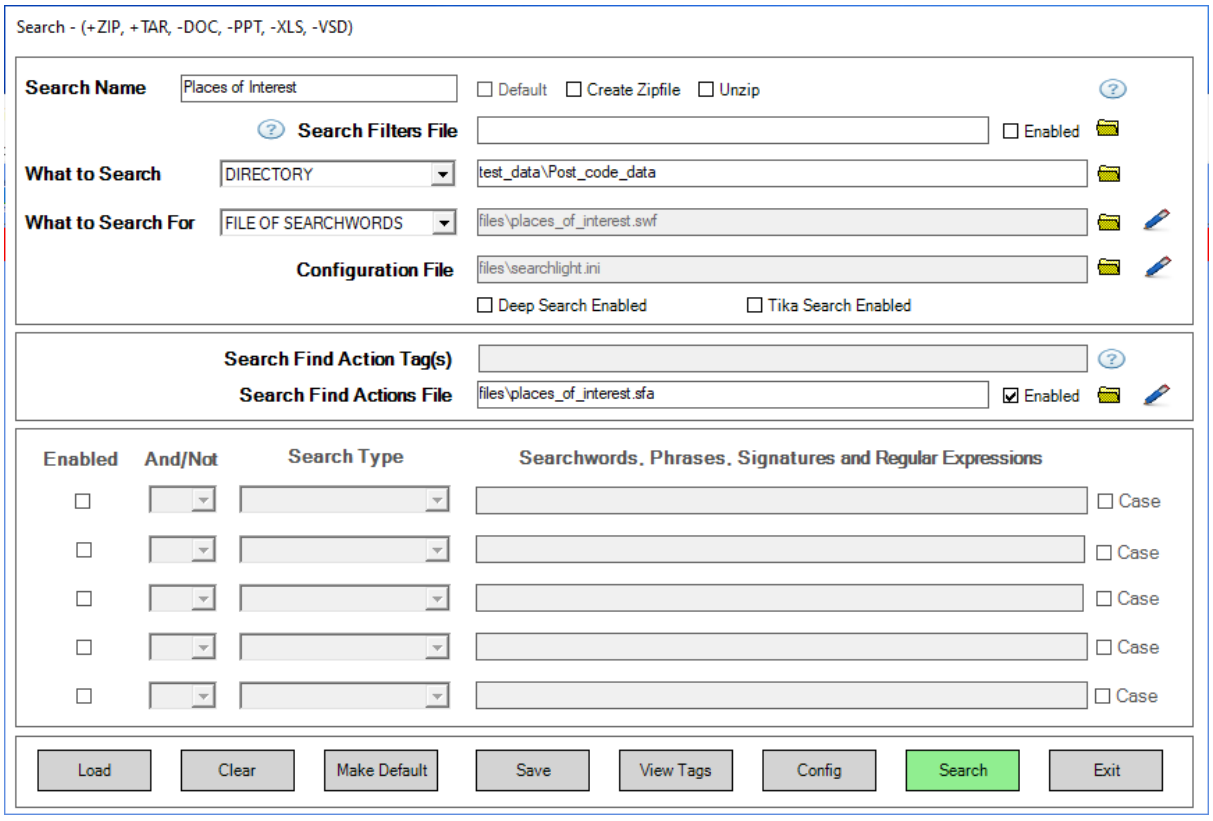
18. Configuration of a Saved Search

The following is an explanation of how the saved search definition 'Places of Interest' is used to identify certain files and then to automatically copy those found files to a nominated directory.

It should be noted that this example utilises Searchlight's Search Find Action functionality which is only available with the 'Professional' edition.

In the Standard edition, the checkbox 'Found File Utility Enabled' and the fields: 'Search Find Action Tag(s)' and 'Search Find Actions File' are not visible.

The 'Places of Interest' saved search definition is as follows:



The above screen shows the following:

- The directory ‘test_data\Post_code_data’ is to be searched.

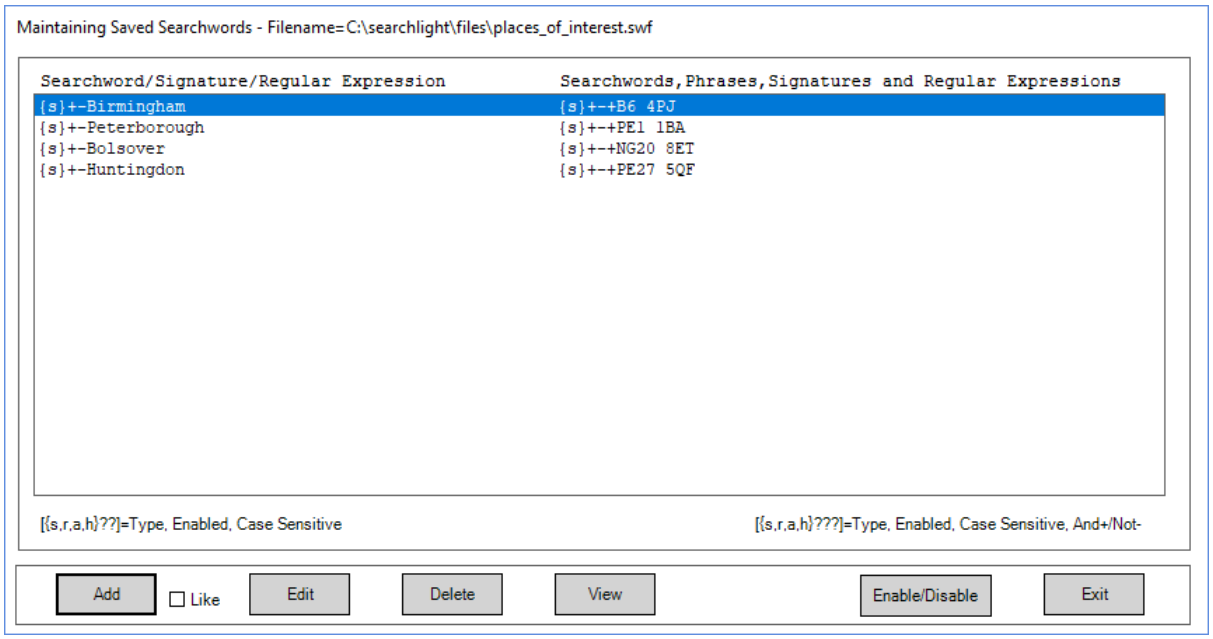
- Files will be searched for any searchword defined in the searchwords file ‘files\places_of_interest.swf’.

- The Search Find Actions, defined within the file ‘files\places_of_interest.sfa’, will be actioned against each file that is found.

If a search is for a single Searchword then the Search Find Action tag(s), defined on the Saved Search screen, are used to match against any enabled Search Find Actions. However, when a ‘File of Searchwords’ is used, as in the example shown above, then the compound Searchwords panel is disabled because any compound conditions used will be those defined for each Searchword loaded from the ‘File of Searchwords’ file.

It should be noted that when a single Searchword search is defined, the compound Searchwords panel is enabled allowing compound search conditions to be defined.

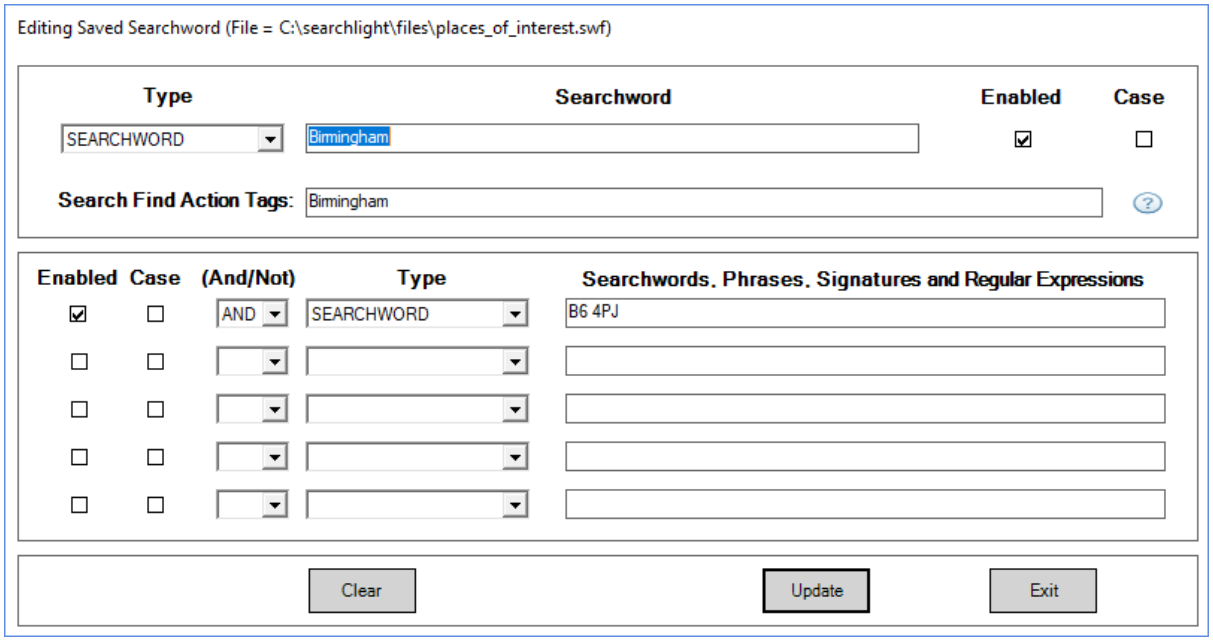
`The Searchwords file (places_of_interest.swf) is defined as follows:



This file shows 4 searchword definitions (Birmingham, Peterborough, Bolsover and Huntingdon) with each one defined as being Enabled, a Searchword and NOT case sensitive (as shown by the {s}+- indicators alongside each of the definitions).

It can also be seen that each of the searchwords has a compound condition associated with it. These have purposefully been added in order to restrict the number of finds as the Post Code data contains a lot of entries for the four place names being searched.

On selecting the 'Birmingham' searchword the following screen is displayed:



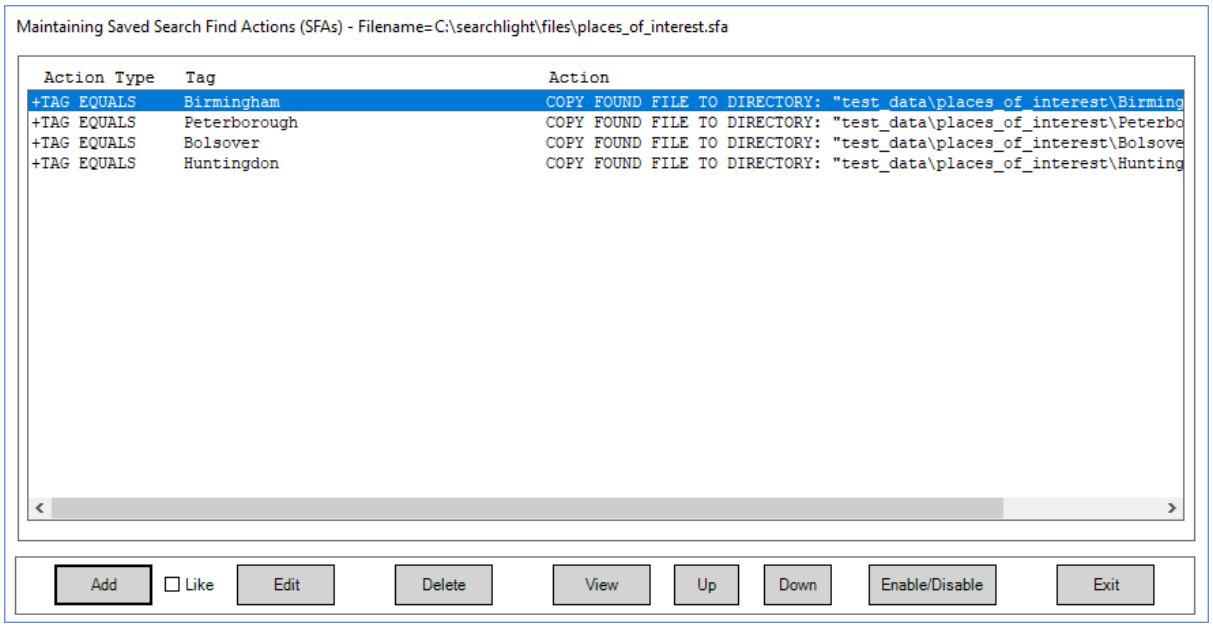
This screen shows that a file will be considered to be found if it contains the NOT case sensitive Searchword 'Birmingham' AND the NOT case sensitive Searchword 'B6 4PJ'.

Important...

It should be noted that the 'Search Find Action Tags:' field contains 'Birmingham' and it is the value of this field/tag that provides the link between any found file and any enabled Search Find Actions.

Thus, any file found to match a defined search condition will be potentially acted upon by the Search Find Actions, defined within the Search Find Action's file associated with the saved search, as shown below.

The Search Find Actions file (places_of_interest.sfa), associated with the saved search definition 'Places of Interest', is defined as follows:

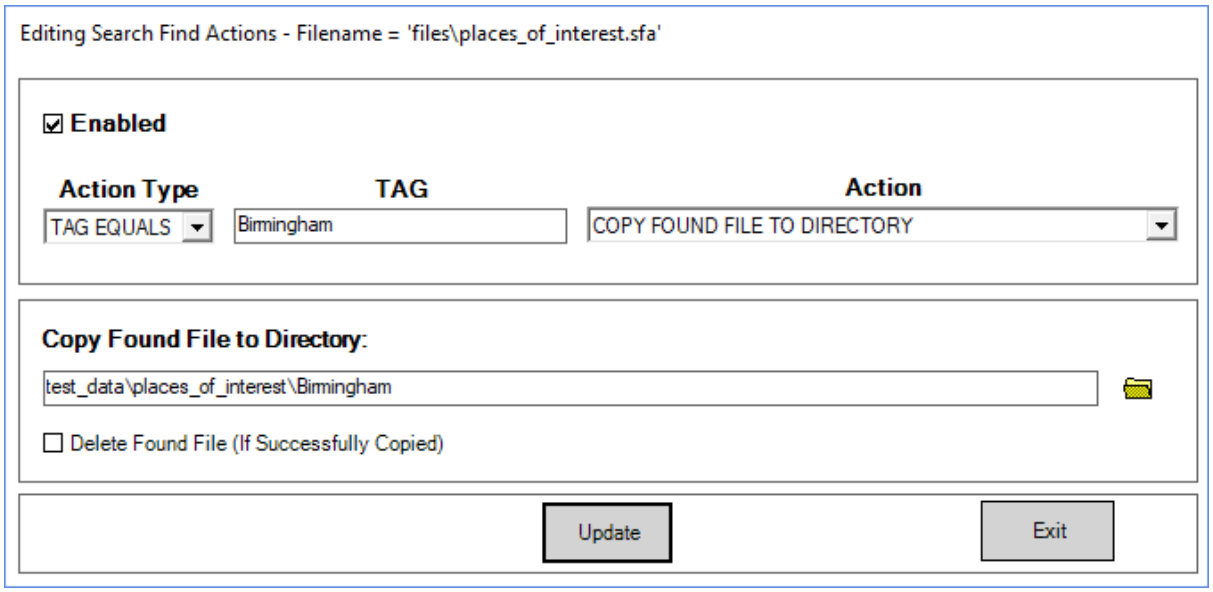


Thus, it can be seen that a 'Search Find Action' has been defined for each of the tags: 'Birmingham', 'Peterborough', 'Bolsover' and 'Huntingdon'.

It should be remembered that, as shown on the previous screen, the tag 'Birmingham' will be associated with any file that is found to contain the Searchword 'Birmingham' AND the Searchword 'B6 4PJ'. Thus, it is this tag 'Birmingham', associated with the Searchword definition that links to the Search Find Action Tag shown above.

On the completion of a search, the Search Find Actions processing engine will perform the actions, enabled within the Saved Search's associated Search Find Actions file, that have a matching tag (Birmingham in this example).

On double clicking the highlighted line on the Search Find Actions definition screen, as shown above, the following screen is displayed:



This screen defines the Search Find Action to be performed when a search result contains the tag 'Birmingham'. Thus, in this case, the action dictates that the found file is to be copied to the directory 'c:\searchlight\test_data\places_of_interest\Birmingham' (which must already exist) and the found file is NOT to be deleted.

Thus, the Search Find Action tag, defined on the Searchword definition screen, provides a direct link to one or more Search Find Actions with a matching tag.

If a search is for a single Searchword, as shown in the screen below, then the Search Find Action tag(s) defined on the Saved Search screen are used, to match against any enabled Search Find Actions.

It should also be noted that when a single Searchword is used for a search then the compound Searchwords panel is enabled allowing compound search conditions to be applied to the single Searchword defined. However, when a 'File of Searchwords' is used, then the compound Searchwords panel is disabled because any compound

conditions used will be the those defined for each Searchword loaded from the ‘File of Searchwords’ file.

19. Tika File Processing

Tika is an Apache Open Source program that can extract plain text from numerous different file types.

Searchlight performs a binary search over any file it encounters and when a file is stored as plain text, Searchlight can easily find targeted words and phrases. However, many programs, such as Microsoft Office, store their data in an obfuscated format and, as such, Searchlight is not able to find its targeted words and phrases in such files.

The Tika program has been specifically developed by the Open Source community to extract plain text from many different file types and it is this functionality that Searchlight exploits.

When appropriately configured, Searchlight utilises the Tika program as follows:

- Searchlight invokes the Tika program and passes it a filename
- Tika extracts the plain text from the file and writes it to a named disk file
- Searchlight searches the extracted plain text file produced by Tika

Thus, Tika is simply extracting the plain text from a file which Searchlight subsequently searches using its own proprietary search algorithm.

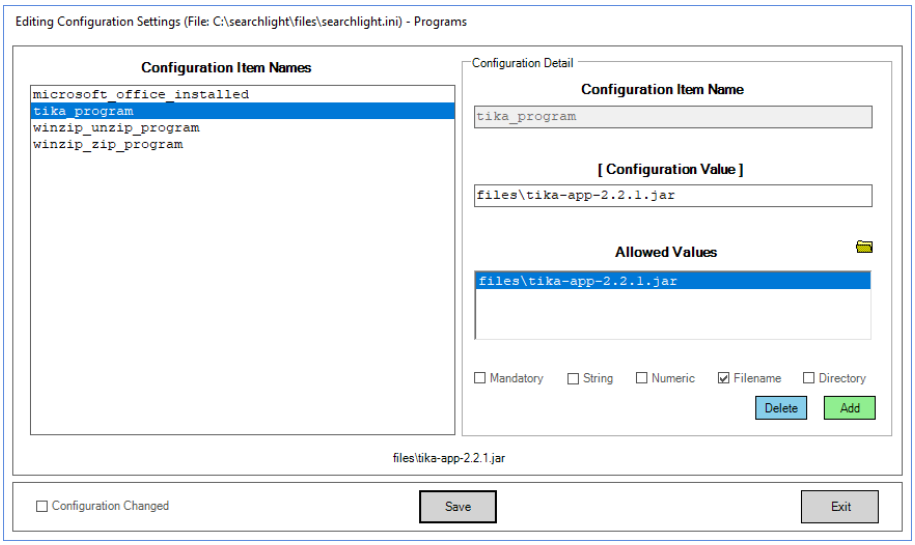
19.1 Installing Tika in Searchlight

Tika is installed in Searchlight as follows:

- Search the internet and download the runnable Jar file version of Tika
- Copy the downloaded Tika Jar file into Searchlight's files directory (...\files\)
- Select the following Searchlight menu options:

  Configuration -> Edit Current Configuration -> Programs

- On selecting the above menu options, the dialog screen below is displayed whereupon:
  - Select the 'tika Program' configuration item (as highlighted)
  - Update the Tika program name in the 'Configuration Value' box to reflect the version of the Tika program that has just been copied into the files directory and select the update button. Then select the exit button and save changes.

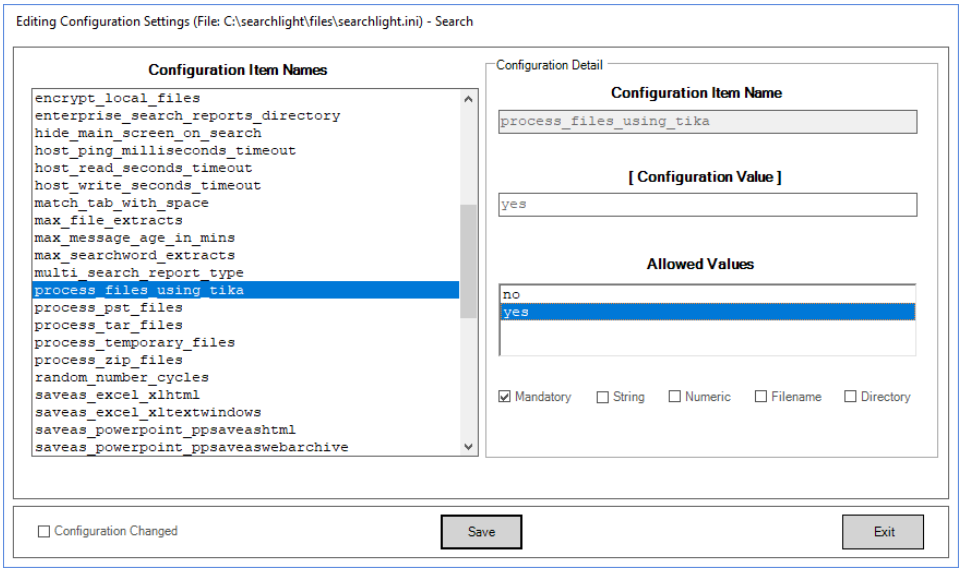


The Tika program has now been integrated into Searchlight but in order for Tika to actually process files, the configuration must be updated as follows:

- Select the following menu options:

  Configuration -> Edit Current Configuration -> Search

- Select the configuration value 'process_files_using_tika' and select 'yes' from the 'Allowed Values' list (this value is defaulted to 'no').



  - Select the 'Update' button then the 'Exit' button and save changes when asked.

If a searchword is found by Tika, then the suffix '(Tika-D)' is appended to the end of the filename, reported in the results file. When the configuration value of 'enable_tika_io_streaming' is set to 'yes' then the suffix '(Tika-M)' is appended instead.

A small performance advantage may be gained by setting the configuration value of ‘enable_tika_io_streaming’ to ‘yes’ as this forces Tika to process the file within memory otherwise Tika processes the file on disk.