



Search32

The Fastest 32 Bits Search Tool For Windows 95
Copyright © Anet Inc., 1996

Contents

Concept and Designated Purpose

Working with Search32.exe

Interface Overview

Hot Keys

Check Boxes from Options Tab

Configuration File Search32.cfg

Index Files or Indexes

Operations on Indexes

Search Queries

Search Query Syntax

Distribution Kit and Requirements

About

Concept and Designated Purpose

Welcome to [Search32](#), the fastest 32 bit search tool for Windows 95 environment!

Easy to use!

[Search32](#) tool is designed to perform word-based search queries on large groups of text (ANSI) files with a total size of tens, or even hundreds, of Megabytes.

Standard search programs, such as Windows 95's [Find](#) tool, perform their searches in a direct manner, scanning an entire disk for matching files. This takes a noticeable time, especially when a large amount of information is scanned.

[Search32](#) tool uses another approach to retrieve the information sought. Search operations are based on a set of specially created ancillary index files, rather than on the original text (ANSI) files. Of course, this technique requires additional disk space. Index file size is usually 50 - 60 % of original text files total size. This method allows [Search32](#) to perform the search queries amazingly fast. This approach is especially useful when it is necessary to perform numerous search queries based on the same index.

Search Queries

A search query is a word expression composed in accordance to search query syntax. Execution of any search query uses a currently loaded index. So, to perform any search query you must create at least one index. See 'Operations on indexes', how to create an index on a group of text (ANSI) files.

To perform a search query:

- 1) Make **Search** tab the current tab.
- 2) Select (load) from the **Index being used** combo box, the index you want. An empty index list (in which case the search is disabled) indicates that there is no index in the index folder. If no index is available or the existing indexes don't cover your needs, create a new one. See 'Operations on indexes' to get help on this.
- 3) Enter a proper search expression in **Enter search expression** combo box or choose one from the combo box list. See search query syntax topic for help with correct search expressions.
- 4) Execute the query by pressing **OK** button.

Once the query has been successfully executed you see the results:
both matching files and entries lists (shown in **Files found** and **Entries found** list boxes), total matching files found, total matching entries found

'Entry' means a source line that contains a word sought. To view more than one particular line of source text you can run a program associated with corresponding file found. Just double click the file name from **Files found** list box. Click right mouse button to run a viewer associated. Click right mouse button with **Shift** key pressed to change attached viewer by another one. (Windows 95 **QuikView** can be successfully used as a right-click file viewer.)



If indexed files (do not confuse with index files) have been changed since the corresponding index has been last updated, it is necessary to update the index. See 'Operations on indexes' to get help on this.

Index Files

Index files (an index) is a set of special ancillary files which allow performance of a search query amazingly fast. No search query is possible if no index exists. Before you can carry out any search, you must create at least one index. See 'Operation on indexes' topic to see how to create, update and remove indexes. A valid index always consists of 8 index files, having the same generic name and the following extensions: [HSH](#), [WRD](#), [WTB](#), [FLS](#), [FTB](#), [ENT](#), [STS](#) and [INF](#). These files contain information about all entries of any particular word from the origin indexed text (ANSI) files and provide fast access and data to retrieval.



- 1) Store all your indexes in a particular folder, making it an index folder for [Search32](#) tool, so that all the indexes can be easily accessed from the [Index being used](#) and the [Index name](#) combo boxes in the [Search](#) and [Indexes](#) tabs respectively.
- 2) Perform all operations on indexes only via Search32 interface.
- 3) Try to work with many indexes of moderate size rather than with one huge index. This minimizes the time required to create, update or load an index and, to a lesser degree, perform search queries. Besides, you can more easily free disk space by removing any index if it is no longer needed.

Search Query Syntax Examples

A search query expression has the following syntax:

[^]word1 <op> [^]word2 <op> [^]word3...

where word1, word2, word3 and so on are the words sought, optional symbol '^' stands for a boolean NOT operation and required <op> symbol can be either '&' or '|' signs that stand for boolean AND and OR operations respectively.

Word's Symbols:

The following symbols are detectable as valid word symbols: _, a - z, A - Z, 0 - 9 and * . Upper and lower-case letters are the same, search queries are CASE INSENSITIVE. Asterisk sign represents an arbitrary non-zero number of valid word symbols and can be placed at the beginning and/or at the end of the word. For example, *map, bit* and *tma* all match word 'bitmap'.

Boolean Operations:

If you wish for any queried words to get all matching files and, the word's entries in them, provided that EACH file must contain ALL the words queried, you should apply AND operation. For example:

word1 & word2 - means to show all files and, entries in them, where both words are presented

If you wish for any queried words to get all matching files and, the word's entries in them, provided that EACH file must contain ANY ONE (or more) of the words queried, you should apply OR operation. For example:

word1 | word2 - means to show all files and, entries in them, where either word1 or word2 is presented

If you wish to get all files not containing any word queried, you should apply NOT operation for that word. For example:

^word - means to show all files not containing word.

Using parentheses:

The precedences of AND and OR operations are the same, a search query expression is parsed from left to right, so word1 | word2 & word3 is not the same as word2 & word3 | word1. To ensure the right precedences use parentheses. For example:

(^(word1 & word2) | word3) & word4

Examples of improper search expressions:

^^word - two signs 'NOT' should be delimited by parentheses: ^(^word)

((word1 | word2) & word3 - numbers of left and right parentheses are not equal.

a & verylongwordwithlengthmorethan32symbols - very short and very long words to search.

Operations on Indexes

[Search32](#) interface provides the following operations on indexes: creating, updating, removing and switching between existing indexes.

To create a new index:

- 1) Make [Indexes](#) tab the current tab.
- 2) Fill in the [Index name](#) combo box with generic name of the index you want to create. Location for the new index is defined by current [index folder](#).
- 3) [Browse for files to index](#) or fill in the [Location of files to index](#) text box with a line using the following syntax:

`pathname1 | pathname2 | ...`,

where `pathname1`, `pathname2` and so on are full pathnames for files to index, delimited from each other with pipe symbol. Wildcards symbols '*' and '?' are available. Example:

`c:\mytxt files*.txt | c:\mypas files*.pas`

Whether or not subfolders (if any) will be processed depends on [Include Subfolders](#) check box from the [Index Options](#) frame.

- 4) Customize parameters from the [Index Options](#) frame.
- 5) Press the [Create](#) button. Keep watch on the status bar at the bottom of the window while indexing. Note that the index creation procedure takes a variable time depending on total size of indexed files as well as performance of your computer.



Be sure to not include files twice in the list of text files to index. For example, if any files `*.pas` locate in `txt` directory along with other `*.txt` files, these files will be included twice in the index:

`c:\txt*.* | c:*.pas`

assuming that [Include Subfolders](#) is checked.

To update an existing index:

- 1) Make [Indexes](#) tab the current tab.
- 2) Select an index you want to update from the [Index name](#) combo box.
- 3) Modify (if necessary) the list of indexed files for that index in [Location of files to index](#) text box. Use [Browse](#) button instead of editing this text box manually.
- 4) Press [Update](#) button and watch the status bar at the bottom of the window.



Text files that have been previously indexed but are not included in the new list are removed from the updated index. Old versions of the same files are replaced with new ones. The same versions remain untouched in updated

index.

To remove an existing index:

- 1) Make [Indexes](#) tab the current tab.
- 2) From the [Index name](#) combo box select the index you want to remove.
- 3) Press [Remove](#) button.

To switch between existing indexes:

- 1) Select (load) any index you want from either [Index being used](#) combo box in [Search](#) tab or [Index name](#) combo box in [Indexes](#) tab. When using the [Index name](#) combo box you are able to view [Notes](#) text box and [Index Statistics](#) frame. Both refer to the currently selected index.

Index folder is shown by [Index folder](#) locked text box at the top right corner of [Indexes](#) tab. It specifies the folder where all indexes you are currently working with are located. To change index folder press the [Change](#) command button and choose any folder you want using [Browse for index folder](#) dialog box.

Interface Overview

[Search32](#) tool's interface allows a user to perform both word-based search queries and operations on indexes. These correspond to two main dialog boxes: [Search](#) and [Indexes](#) tabs respectively. Go to the Options tab to customize [Search32](#) start behavior and hot keys combination.

While in the [Search](#) tab you can:

- 1) Select (load) an index you want from the list of indexes currently available.
- 2) Enter and execute a new word search query or one from the list of current search session queries.
- 3) View the results of an executed query: a list of matching files found and a list of matching source lines found.
- 4) Run the program associated with any particular matching file found to view more than one line of source text.
- 5) Attach and run any file viewer to quickly view files found.
- 6) Switch on the [Indexes](#) tab.

While in the [Indexes](#) tab you can:

- 1) Select (load) an index you want from the list of currently available indexes.
- 2) Create a new index for a group of text (ANSI) files.
- 3) Update an existing index.
- 4) Remove an existing index.
- 5) Change current index folder.
- 6) View notes, statistics and options for the currently selected index.
- 7) Switch on the [Search](#) tab.

All previously executed search queries are accumulated and become available during the current search session from the [Index being used](#) combo box.

Configuration file Search32.cfg

Configuration file [Search32.cfg](#) is automatically re-written each time you exit [Search32](#) tool. It is located in the same folder as [Search32.exe](#).

[Search32.cfg](#) stores the following parameters:

- 1) start parameters and hot keys combination from the Options tab;
- 2) an index folder;
- 3) the index name you last worked with;
- 4) full path name for right-click file viewer attached;
- 5) a list of file types used in [Add Files to Be Indexed](#) dialog box;
- 6) registration number and password values;

When [Search32](#) begins running, it tries to load the index specified in [Search32.cfg](#). It is assumed that this index is located in the index folder from the same [Search32.cfg](#). If [Search32.cfg](#) doesn't exist or contains improper parameters, the folder containing [Search32.exe](#) becomes an index folder. In this case the executable will try to load any index from the folder.

Browse

Pressing [Browse](#) button opens [Add Files To Be Indexed](#) dialog box. It provides the ability to define the list of files to be indexed instead of filling [Location of files to index](#) text box manually.

There are two ways to determine the list of files to be indexed: through the use of [Add Folder](#) and [Add Files](#) command buttons respectively. The first means to add all files of the currently selected type (see [Files types to select:](#) combo box) from the currently selected folder (see [Folders:](#) list box). The second means to add only the selected files from [Files](#) list box. Multiple selections are available. Use standard Windows technique to select them. All your actions are reflected in the Result text box at the top of the dialog box. You can freely edit this text box using syntax that is exactly the same as [Location of files to index](#) text box. See ['Operations on indexes'](#) topic to get help on this.



- 1) Whether or not subfolder' s contents (if any) will be included in the list of indexed files depends only on [Include Subfolder](#) check box.
- 2) It is not recommended to use [*.* \(All files\)](#) file's type for creating the list of indexed files, because along with the files of text (ANSI) format, the files of improper formats, e.g. executables, help files and so on, implicitly come into the list. The program does not check indexed files for their formats.

Index Options

This frame contains the following parameters:

Include Subfolder check box:

Defines whether or not subfolders (if any) were included as locations for indexed files.

Language combo box:

Now supports [English](#) and [English + Russian](#) ANSI character codings. If you precisely know, that you are working with English texts only, you should choose [English](#) coding while creating an index. This eliminates the inclusion in the index of any "words", consisting of improper characters that sometimes are encountered in text files. On the contrary, choose [English + Russian](#) coding, when you need to index files, consisting of characters with both English and Russian ANSI codings.

Maximum number of unique words text box:

Specifies the maximum number of unique words the system can process while indexing. When creating a new index it is reasonable to set this important value with a safety margin, otherwise you may get an **unique words overflow** error while indexing. The default value is 150000. This is sufficient for indexing large volumes, but sometimes a greater value is required. On the other hand, do not increase this value unless necessary, since it increases total index size.

Maximum word length and Minimum word length text boxes:

Specify a maximum and a minimum word length while indexing or for a currently selected index. Words with greater or lesser length are ignored while indexing and searching. You cannot enter a maximum greater than 32 or a minimum less than 2.

Last updated label:

Displays date and time when currently selected index was last updated.



Note, you can change all editable parameters from the [Index Options](#) frame only when creating a new index (not when updating an existing one).

Use [Notes](#) text box to enter any user comments for a particular index while creating or updating it.

Distribution Kit and Requirements

[Search32](#) distribution kit consists of two floppy disks.

Disk1:

ctl3d32.dl_
keyhook.dl_
mfc40.dl1
msvcrt20.dl_
msvcrt40.dl_
olepro32.dl_
setup.exe
setup.lst
setup132.ex_
srch32_b.dl_
st4unst.ex_
stkit432.dl_
tabctl32.oc_
vb40032.dl_
ven2232.ol_

Disk2:

comctl32.oc_
comdlg32.oc_
mfc40.dl2
search32.ex_
search 32.hl_
threed32.oc_

Requirements:

[Search32](#) tool is designed for Windows 95 operating system. It is not compatible with previous versions of Windows.

About

This product is copyrighted by:

© Anet Inc., 1996

Contact information:

Anet Inc.

1070 Quartermaster Canyon

San Ramon, CA 94583

U.S.A.

Phone: 800 - 434 - 3439

510 - 735 - 1173

FAX: 510 - 735 - 7035

E-Mail: 70651.1125@compuserve.com

Implementation notes:

This product has been made on the base of [Srch32_b.dll](#) dynamic link library which is a powerful engine for indexing and text retrieval operations. It includes a set of unique algorithms that specially have been arranged as comprehensive and easy to use tool. Its interface is designed as a number of functions that can be easily built into any 32-bit codes to provide speedy search capabilities. **Anet** company is ready to help you with your needs in up-to-date text retrieval technologies. Regarding this subject, please contact us by E-mail:

102740.104@compuserve.com

Multilanguage support:

Now only English and Russian ANSI character sets are supported. Regarding the support of other languages, please contact us by E-mail:

102740.104@compuserve.com

Search Queries Examples

Sign \Leftrightarrow in the following examples means equivalence. Remember that all search queries are case insensitive.

Query expression:

Query action:

car		to find all files and, entries of car in them, provided that each file must contain at least one entry of car
^car		to find all files not containing word car
car & lorry		to find all files and, entries of car and lorry in them, provided that each file must contain at least one entry of car and one entry of lorry
car lorry		to find all files and, entries of car and lorry in them, provided that each file must contain at least one entry of car or one entry of lorry
car & ^lorry		to find all files and, entries of car in them, provided that each file must contain at least one entry of car and no entry of lorry
^car & ^lorry	\Leftrightarrow	^(car lorry) to find all files provided that each file doesn't contain neither car nor lorry
^car ^lorry	\Leftrightarrow	^(car & lorry) to find all files provided that each file doesn't contain either car or lorry
bit*		to find all files and, entries of words beginning with bit... in them (e.g. bitmap , bitwise and so on), provided that each file must contain at least one matching entry
bit* *bit* *bit		to find all files and, entries of words in them, which begin with bit... (e.g. bitmap), have ...bit... in the middle (e.g. arbitrary) or end with ...bit (e.g. inhibit) provided that each file must contain at least

any one matching entry

Hot Keys

[Search32](#) has <Ctrl + 'letter key'> hot keys combination (to choose a proper 'letter key' go to the [Options tab](#) and select any one). If you have [Search32](#) "sleeping" on the taskbar, this gives you a handy way to find a word from within another program you are currently working with. Just select the word and press the hot keys combination. You will be amazingly quickly supplied with appropriate search results.

Check Boxes from Options Tab

Run at Windows start:

Specifies whether [Search32](#) will run at every Windows start.

Update all indexes at start:

Specifies whether [Search32](#) will update all the indexes whenever it starts. This will ensure that all relevant files modifications, deletions and creations since the time the indexes have been last updated, will be properly reflected in updated indexes.

Run minimized:

Specifies whether [Search32](#) will appear minimized on the taskbar at start.

