SearchEngine 20.20

by **Thorsten Doherr**
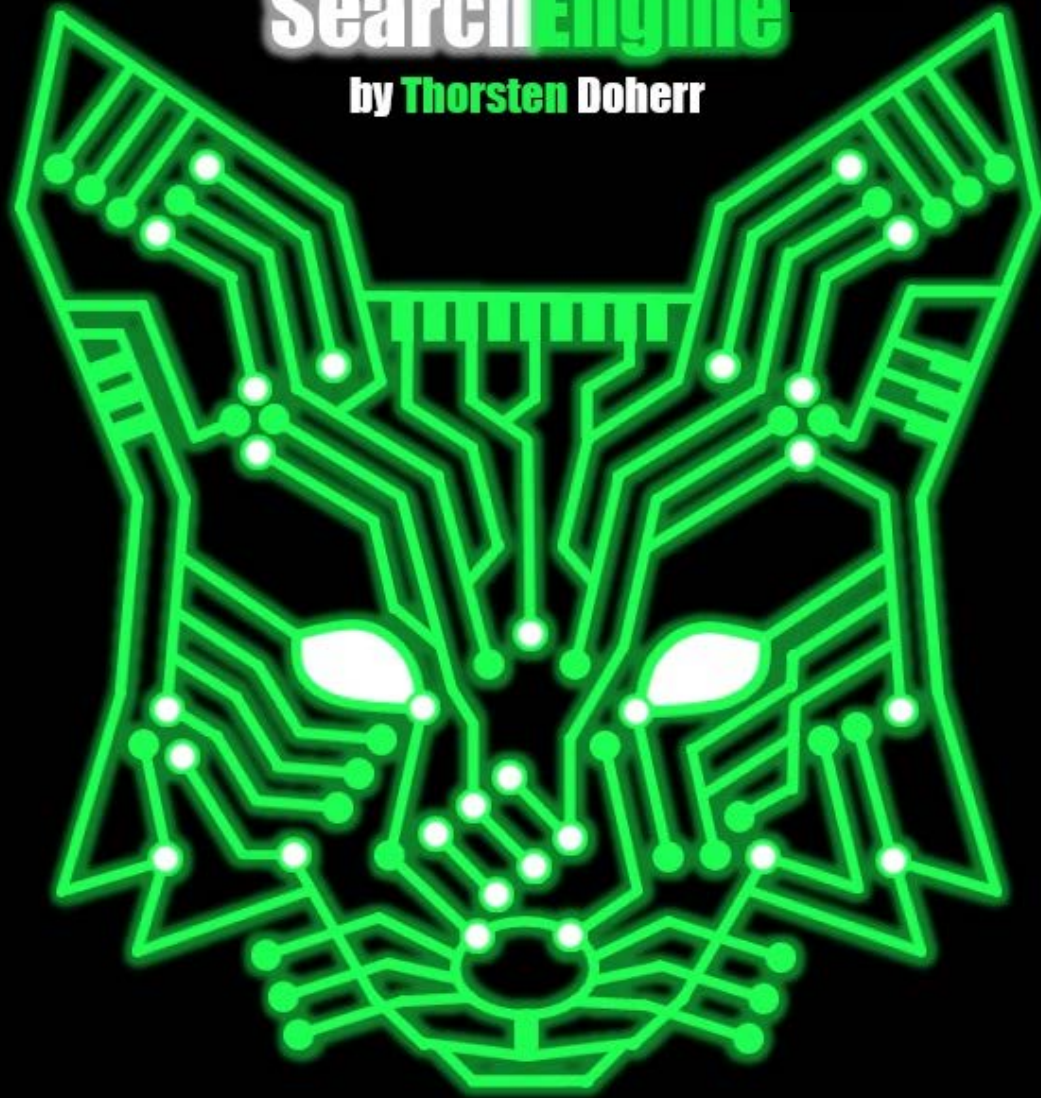
powered by **FoxPro** & **ParallelFox**

https://github.com/ThorstenDoherr/searchengine

SearchEngine installation and legend

Installation

- On the GitHub page click on [Code] →[Download ZIP] to get a zip container. The code directory contains the source code.
- Just copy the contents of the "SE" sub-directory from the zip container into an empty directory of your choosing. This directory is called SearchEngine directory.
- The directory "preparer" contains special country or content specific preparer. Copy the respective "SearchEngine.xml" file into your SearchEngine directory to activate them. Naturally, only one special preparer file can be active per SearchEngine.
- Because the SearchEngine itself will create many additional files, it is advised to keep it separate from the base, search and result tables in an exclusive directory (traditionally called "SE", residing at the level of the associated base table).
- As the base table constitutes the SearchEngine, every base table requires a separate SearchEngine directory. Different preparer/search type setups for the same base table also require separate directories.
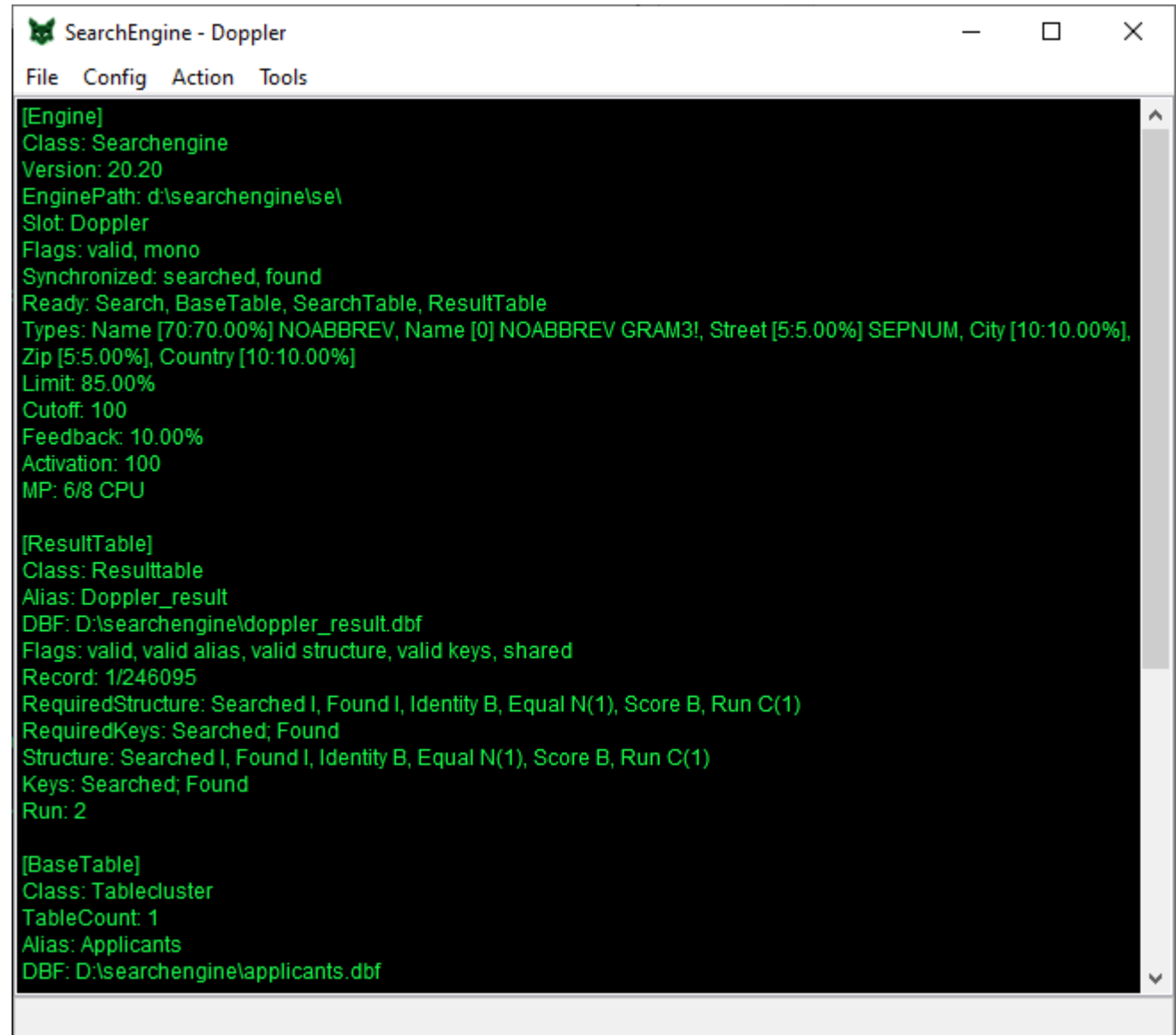- Result tables should reside close to the search table they are linked to.

Legend

- Basic information
- Best practice suggested by the developer
- Potential source for misconduct

SearchEngine main window

Main window

- Main window shows the so called structure string representing all settings.
- The structure string is also the save format of the SE.
- The string can be marked and copied to the clipboard, e.g. in case of a support request.

SearchEngine - Doppler ▫ □ ✕

File   Config   Action   Tools

```
[Engine]
Class: Searchengine
Version: 20.20
EnginePath: d:\searchengine\se\
Slot: Doppler
Flags: valid, mono
Synchronized: searched, found
Ready: Search, BaseTable, SearchTable, ResultTable
Types: Name [70:70.00%] NOABBREV, Name [0] NOABBREV GRAM3!, Street [5:5.00%] SEPNUM, City [10:10.00%],
Zip [5:5.00%], Country [10:10.00%]
Limit: 85.00%
Cutoff: 100
Feedback: 10.00%
Activation: 100
MP: 6/8 CPU

[ResultTable]
Class: Resulttable
Alias: Doppler_result
DBF: D:\searchengine\doppler_result.dbf
Flags: valid, valid alias, valid structure, valid keys, shared
Record: 1/246095
RequiredStructure: Searched I, Found I, Identity B, Equal N(1), Score B, Run C(1)
RequiredKeys: Searched; Found
Structure: Searched I, Found I, Identity B, Equal N(1), Score B, Run C(1)
Keys: Searched; Found
Run: 2

[BaseTable]
Class: Tablecluster
TableCount: 1
Alias: Applicants
DBF: D:\searchengine\applicants.dbf
```
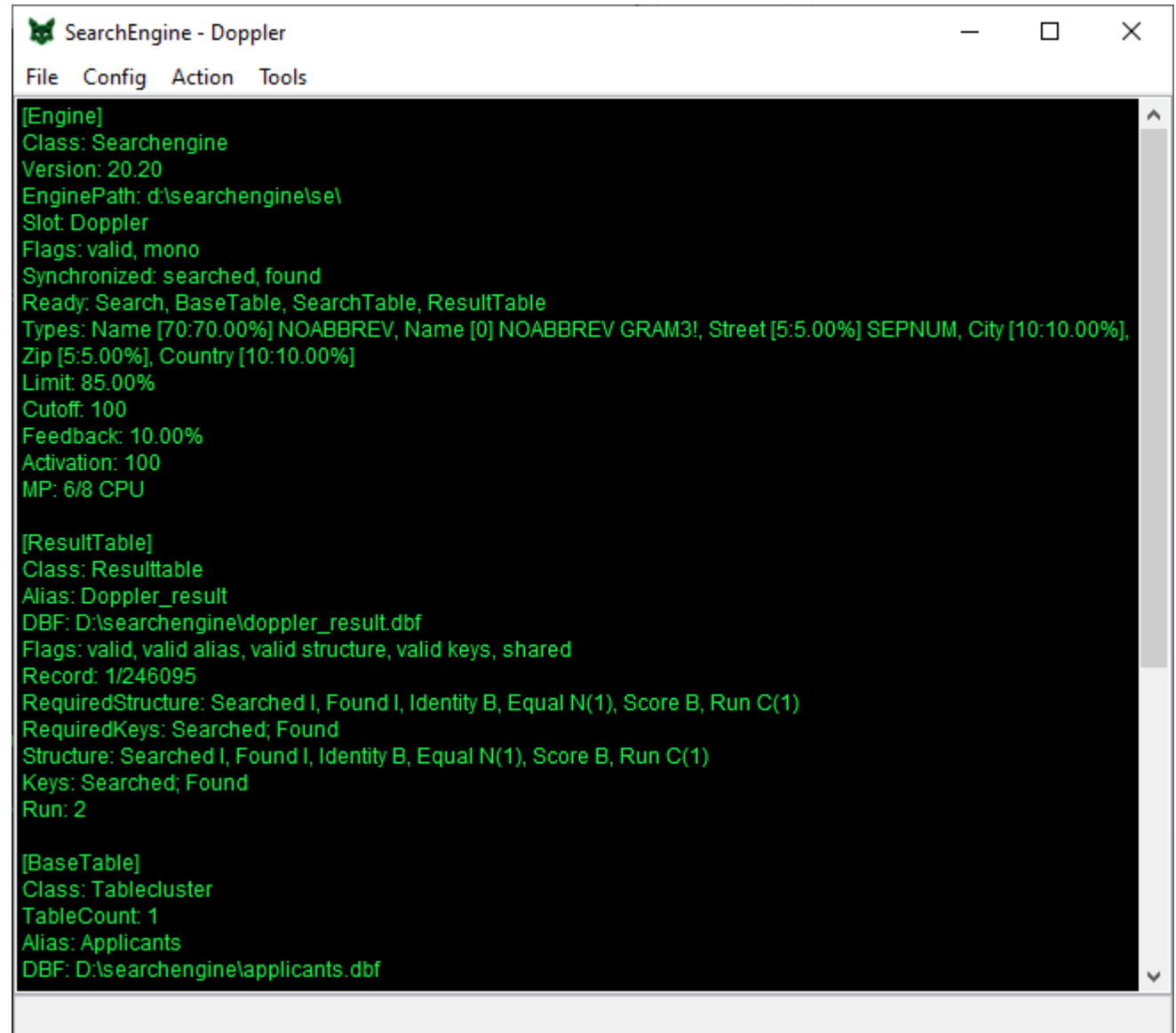
SearchEngine menu structure

Menu [suggested order]
- File
  - Save settings [8]
  - Load Settings [9]
  - Export
    - Export [10]
    - Extended Export [11]
    - Grouped Export [12]
    - Result Export
    - Meta Export
  - Exit
- Config
  - File Locations [1]
  - Join SearchFields [3]
  - SearchTypes [4]
  - Settings [5]
  - Preferences
- Action
  - Search [6]
  - Research
  - Refine
  - Create [2]
  - Recreate
  - Expand
  - Mirror
- Tools
  - QuickSearch
  - ResultChecker [7]
  - Notes
  - Browser

SearchEngine - Doppler — □ ✕

File   Config   Action   Tools

```
[Engine]
Class: Searchengine
Version: 20.20
EnginePath: d:\searchengine\se\
Slot: Doppler
Flags: valid, mono
Synchronized: searched, found
Ready: Search, BaseTable, SearchTable, ResultTable
Types: Name [70:70.00%] NOABBREV, Name [0] NOABBREV GRAM3!, Street [5:5.00%] SEPNUM, City [10:10.00%],
Zip [5:5.00%], Country [10:10.00%]
Limit: 85.00%
Cutoff: 100
Feedback: 10.00%
Activation: 100
MP: 6/8 CPU


[ResultTable]
Class: Resulttable
Alias: Doppler_result
DBF: D:\searchengine\doppler_result.dbf
Flags: valid, valid alias, valid structure, valid keys, shared
Record: 1/246095
RequiredStructure: Searched I, Found I, Identity B, Equal N(1), Score B, Run C(1)
RequiredKeys: Searched; Found
Structure: Searched I, Found I, Identity B, Equal N(1), Score B, Run C(1)
Keys: Searched; Found
Run: 2


[BaseTable]
Class: Tablecluster
TableCount: 1
Alias: Applicants
DBF: D:\searchengine\applicants.dbf
```
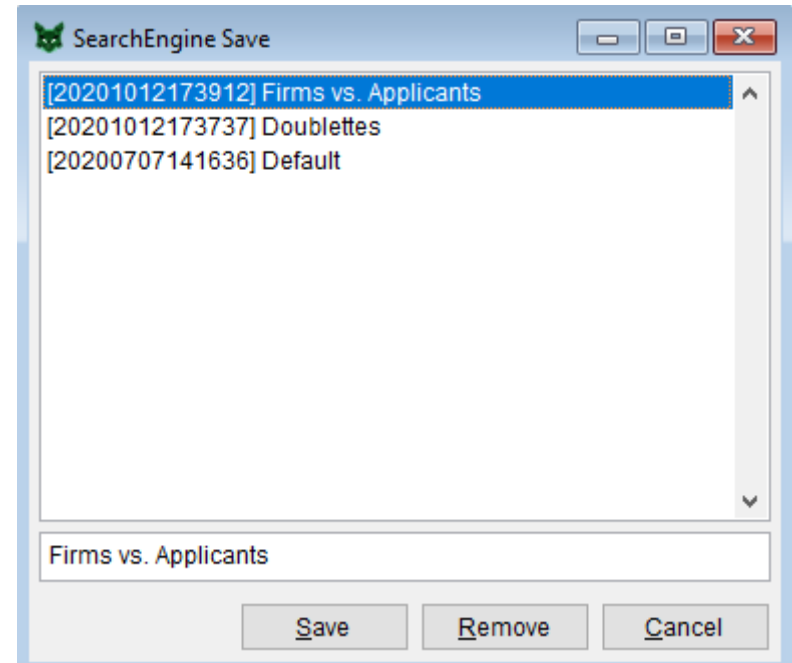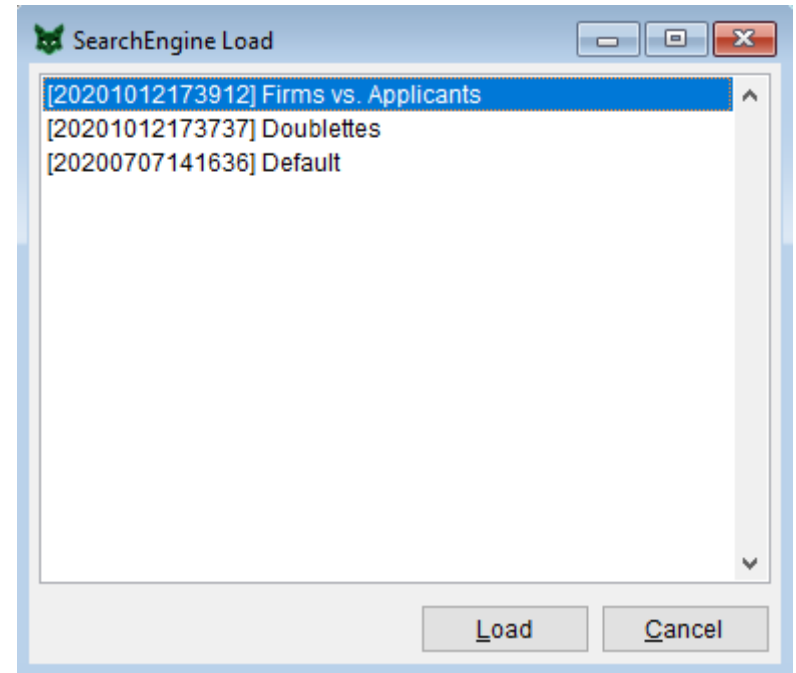
SearchEngine ▶File ▶Save

Save
- Saves the current SearchEngine structure string (all settings) into a named save slot.
- On startup the most recently saved slot will be used to instantiate the SearchEngine.
- Save slots not required anymore can be selected and then removed.
- The "Default" slot can never be removed because the SearchEngine references it for the search type structure → the "create", "recreate" and "expand" functions always refresh the "Default" slot.
- It is easy to accidentally overwrite a save slot, therefore it is best practice to first create a fresh save slot for every new project before making changes → the new slot, having the most recent timestamp, is used by default.

SearchEngine Save

[20201012173912] Firms vs. Applicants
[20201012173737] Doublettes
[20200707141636] Default

Firms vs. Applicants

Save   Remove   Cancel

SearchEngine ▶ File ▶ Load

Load
- Loads the structure string of the selected slot to instantiate a new SearchEngine.
- On startup the most recently saved slot will be used to instantiate the SearchEngine.
- If you plan a new search project, which is similar to an existing one, load the similar project and save it immediately under a new name to save time and avoid accidental overwriting.
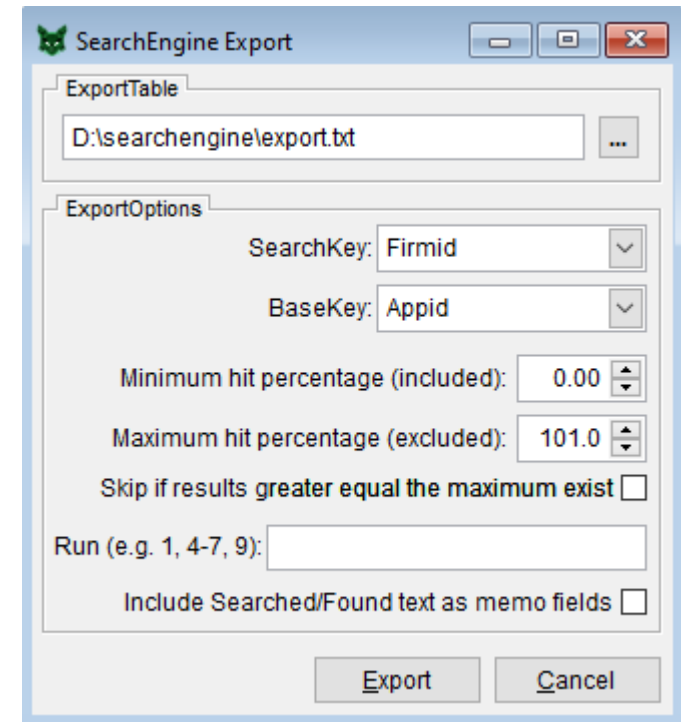

SearchEngine Load

[20201012173912] Firms vs. Applicants
[20201012173737] Doublettes
[20200707141636] Default

Load   Cancel

Export
- The format contains all information contained in the result table.
- The record numbers of the result table can be replaced by primary (unique) keys of the associated base and search table records.
- By specifying the extension ".txt", the export table will be a tab delimited text file.
- The option "SearchKey" specifies the unique key of the search table.
- The option "BaseKey" specifies the unique key of the base table.
- Choosing RECORD as "BaseKey" or "SearchKey" references the record number of the corresponding table instead of key fields → the tables do not need unique keys.
- If key fields are used, they have to be unique. If you are using a search field, e.g. firm name or person name, something is wrong.
- The range of exported results can be selected by the option "Lowest hit percentage" and "Highest hit percentage" (the later will be excluded from the selection) in regard of the candidate identity.
- The option "Skip if results greater equal the maximum exists" excludes candidates within the respective range for search records having already candidates in a higher range.
- The "Run" option allows for selecting specific search runs similar to the pages selection syntax for printing, i.e. 1-3, 7, 8 would select run 1, 2, 3, 7, 8.

- "Include Searched/Found text…" appends the contents of the respective search and base table fields as long text fields separated by pipes "|". This function is deprecated (see Extended Export).

Export
- Run selection is an efficient way to evaluate search strategies
- By selecting ranges and skipping lower alternative candidates, the results can be separated into confidence groups, i.e. 101-100, 100-95, 95-90, 90-80 and so on, where every search key will only be reported within the group that belongs to its highest ranked candidates.
- The identity and score are reported with a higher precision than in the other export formats.
- This function provides a simple, space efficient format for further handling by external tools or methods.
- It is applicable if manual checking (eyeballing) is not intended.
- Including the search fields can be appropriate if subsequent string comparison/manipulation with external programs is intended.
- Do not use the option "Skip if results greater equal…" if the base table contains unidentified duplicate entries, like variants of a patent assignee. This is the case if the base table does not have a proper focus (see SearchEngine presentation).
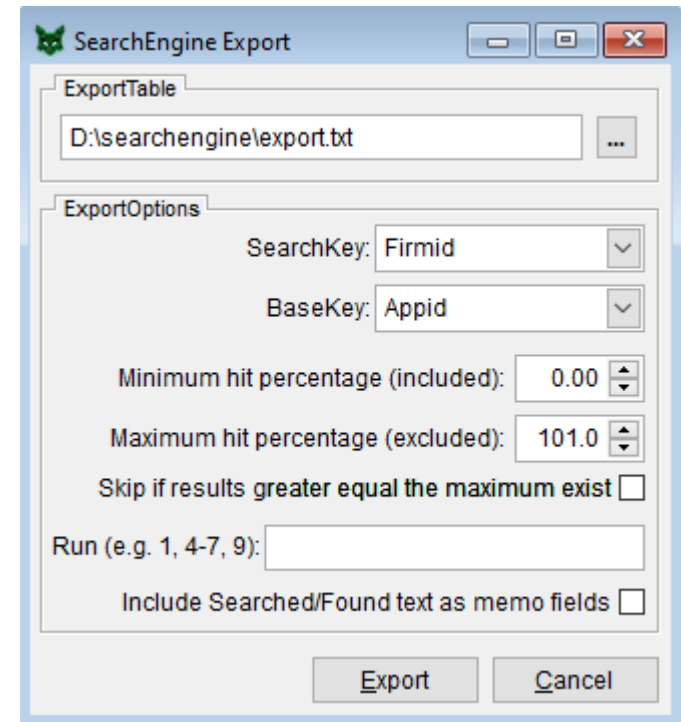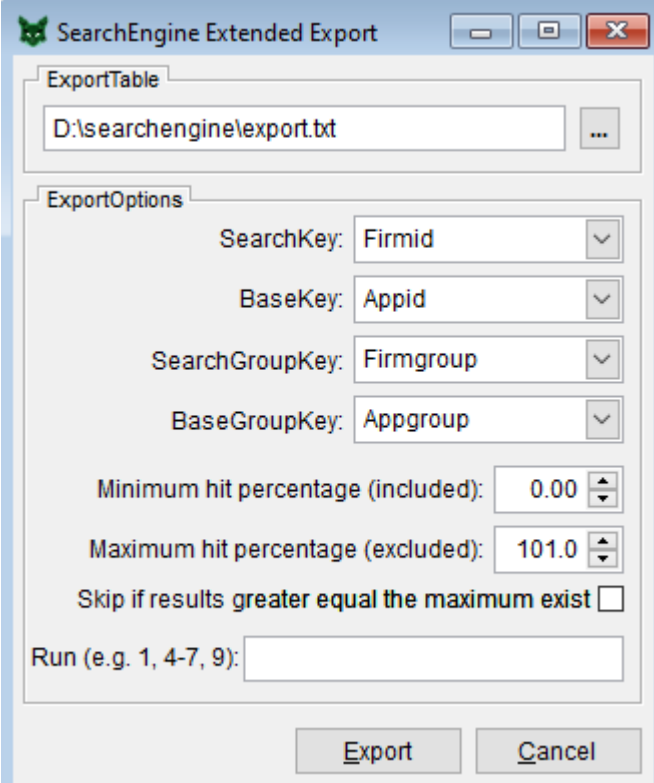
searched	found	identity	equal	score	run
1	4	100	0	5.332523637693407	1
1	2	99.568468	0	5.332523637693407	1
1	3	99.568468	0	5.332523637693407	1
1	80415	90	0	5.332523637693407	1
1	27269	90	0	5.332523637693407	1
1	27267	90	0	5.332523637693407	1
1	27270	90	0	5.332523637693407	1
1	27271	90	0	5.332523637693407	1
1	27268	90	0	5.332523637693407	1
1	27272	89.568468	0	5.332523637693407	1
2	2	99.90073099999999	0	5.506045795888449	1
2	3	99.90073099999999	0	5.506045795888449	1
2	4	97.183803	0	5.506045795888449	1
2	80415	87.28307199999999	0	5.506045795888449	1
2	27269	87.28307199999999	0	5.506045795888449	1
2	27267	87.28307199999999	0	5.506045795888449	1
2	27270	87.28307199999999	0	5.506045795888449	1
2	27272	87.28307199999999	0	5.506045795888449	1
2	27271	87.28307199999999	0	5.506045795888449	1
2	27268	87.28307199999999	0	5.506045795888449	1
3	15	95	0	2.731191135223984	1
3	16	95	0	2.731191135223984	1
3	14	95	0	2.731191135223984	1
3	55181	90	0	2.731191135223984	1
3	13	90	0	2.731191135223984	1
3	55182	90	0	2.731191135223984	1
3	55180	90	0	2.731191135223984	1
3	71467	88.809524	0	2.731191135223984	1
3	71465	88.809524	0	2.731191135223984	1
3	1920	88.809524	0	2.731191135223984	1

Extended Export

- This format is suited for manual checking as the content for the linked search fields of the base table and the search table will get reported in a clear layout.
- The format additionally contains all information provided by the result table.
- By specifying the extension ".txt", the export table will be a tab delimited text file.
- The options "SearchKey", "BaseKey", the range selection and run filtering are equivalent to the (simple) Export dialog.
- The range selection is a good way to reduce and structure the output without loosing substantial information.
- "SearchGroupKey" designates an entity key within the search table, which has the same value for a group of records referencing a specific entity, i.e. different historic variants of a firm name.
- "BaseGroupKey" does the same as "SearchGroupKey" for the base table.
- Using one or both of these group keys instructs the SearchEngine to aggregate the candidates with a prior on quality on the respective entity key as the actual "key of interest".
- It is not required to specify unique keys for SearchKey and BaseKey as the record number is always "unique", but sometimes it is nice to have them reported instead of record numbers.
- The "group" field in the export format reports the "SearchGroupKey" while the candidates are distributed among the associated "SearchKeys" or record numbers by best fit.
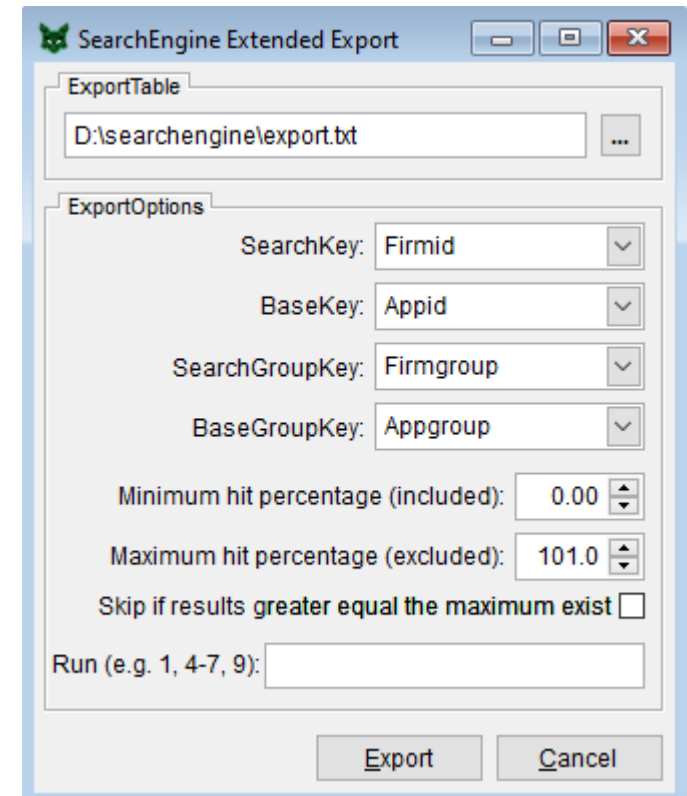
SearchEngine Extended Export

ExportTable

D:\searchengine\export.txt

ExportOptions

SearchKey: Firmid
BaseKey: Appid
SearchGroupKey: Firmgroup
BaseGroupKey: Appgroup
Minimum hit percentage (included): 0.00
Maximum hit percentage (excluded): 101.0
Skip if results greater equal the maximum exist ☐
Run (e.g. 1, 4-7, 9):

Export   Cancel

- GroupKeys in rectangular brackets (further down the lists) group only but will not be reported. This is useful in conjunction with Meta Export.
- If reliable group keys exists, they reduce the output by focusing on the actual keys of interest.

Extended Export

- Range selection greatly reduces the effort of eyeballing especially for high quality groups, which can directly be used for preliminary analysis. Range selection with range exclusivity ("Skip results…") brings candidates of the same quality level together, which eases the burden on eyeballing by leveling the shortcomings of the candidates within each range.
- The sorting of ExtendedExport tables by candidate size (descending) and score (ascending) brings the worst results with the lowest identification potential on top while the best can be found at the bottom.
- Look at the bottom of ExtendedExport tables for still valid results even in low quality ranges. If completeness is not required, go upwards until the diminishing returns aren't worth the effort anymore.
- Even though unchecked ranges respectively records cannot be considered quality matches, they still can be excluded from potential control groups.
- The "equal" column is reserved for manual marking of valid matches by comparing the top line of every block displaying the search term with the following candidate lines.
- Use the top line of a block to designate the default value for the whole block: 1 = match, 9 = no match. Mark all exceptions of the default assessment of a block with the opposing value (default = 9, exception = 1, or other way round).

SearchEngine Extended Export

ExportTable

D:\searchengine\export.txt   [...]

ExportOptions

SearchKey: Firmid

BaseKey: Appid

SearchGroupKey: Firmgroup

BaseGroupKey: Appgroup

Minimum hit percentage (included): 0.00

Maximum hit percentage (excluded): 101.0

Skip if results greater equal the maximum exist ☐

Run (e.g. 1, 4-7, 9):

Export   Cancel

- Think about other efficient ways to organize the manual checking like only marking non-matches in high quality ranges and only marking matches in low quality ranges or using other codes to signal uncertainty.

GROUP	SEARCHED	FOUND	IDENTITY	EQUAL	FIRM	ADDRESS	CITY	ZIP	COUNTRY	SCORE	CNT	RUN
3861	3862				Hitachi, Ltd.	12th Floor, Marunouchi Center Building, 6-1, Marun	Tokyo	100-8220	JP	6.11	16	
3861	3862	36165	95.03		Hitachi, Ltd.	6-6, Marunouchi 1-chome Chiyoda-ku	Tokyo	100-8220	JP	6.11	16	1
3861	3862	50034	90.10		Kabushiki Kaisha Hitachi Seisakusho(Hitachi, Ltd.)	6-6, Marunouchi 1-chome	Chiyoda-ku,Tokyo	100-8280	JP	6.11	16	1
3861	3862	43819	90.08		Hitachi, Ltd.	6 Kanda Surugadai 4-chome	Chiyoda-ku, Tokyo	100-8010	JP	6.11	16	1
3861	3862	45426	90.08		Hitachi, Ltd.	6 Kanda Surugadai 4-chome Chiyoda-ku	Tokyo	100-8010	JP	6.11	16	1
3861	3862	50255	90.08		HITACHI CONSTRUCTION MACHINERY CO., LTD.	6-2, Ohtemachi 2-chome Chiyoda-ku	Tokyo	100-0004	JP	6.11	16	1
3861	3862	86988	90.08		Hitachi Consumer Electronics Co., Ltd.	2-1, Otemachi 2-chome Chiyoda-ku	Tokyo	100-0004	JP	6.11	16	1
3861	3862	56291	90.02		Hitachi Building Systems Co., Ltd.	6, 1-chome, Kandanishiki-cho, Chiyoda-ku	Tokyo	101-0054	JP	6.11	16	1
3861	3862	2358	90.00		HITACHI CHEMICAL CO., LTD.	Shinjyuku-Mitsui Building,1-1, 2-chome Nishishinju	Shinjuku-ku,Tokyo	163	JP	6.11	16	1
3861	3861				Hitachi, Ltd.	6, Kanda Surugadai 4-chome Chiyoda-ku	Tokyo	101	JP	0.57	16	
3861	3861	36165	100.00		Hitachi, Ltd.	6, Kanda Surugadai 4-chome, Chiyoda-ku	Tokyo	101-8220	JP	0.57	16	1
3861	3861	45426	100.00		Hitachi, Ltd.	6 Kanda Surugadai 4-chome, Chiyoda-ku	Tokyo	101-8010	JP	0.57	16	1
3861	3861	14527	99.95		HITACHI TECHNO ENGINEERING CO., LTD.	3, Kanda Surugadai 4-chome,Chiyoda-ku	Tokyo	101	JP	0.57	16	1
3861	3861	16176	99.75		Hitachi, Ltd.	6 Kanda Surugadai 4-chome	Chiyoda-ku,Tokyo	101-0062	JP	0.57	16	1
3861	3861	43819	99.75		Hitachi, Ltd.	6, Kanda Surugadai 4-chome	Chiyoda-ku,Tokyo	101-8010	JP	0.57	16	1
3861	3861	56291	96.11		Hitachi Building Systems Co., Ltd.	6 Nishiki-cho 1-chome Kanda,Chiyoda-ku	Tokyo	101-0054	JP	0.57	16	1
3861	3861	86309	96.04		Hitachi Industrial Equipment Systems Co., Ltd.	3, Kanda Neribei-cho Chiyoda-ku	Tokyo	101-0022	JP	0.57	16	1
3861	3861	67698	95.28		Hitachi Plant Technologies, Ltd.	1-14, Uchikanda 1-chome Chiyoda-ku	Tokyo	101-0047	JP	0.57	16	1
4617	4617				Delphi Technologies, Inc.	Legal Staff, 1450 W. Long Lake Road, P.O. Box 5052	Troy	MI 48007-5052	US	3.89	11	
4617	4617	55132	100.00		Delphi Technologies, Inc.	Legal Staff,1450 W. Long Lake Road,P.O. Box 5052	Troy	MI 48007-5052	US	3.89	11	1
4617	4618				Delphi Technologies Inc.	Legal Staff- Mail Code:480-410-202, P.O. Box 5052	Troy	MI 48007-5052	US	3.97	11	
4617	4618	55132	99.44		Delphi Technologies, Inc.	Legal Staff Mail Code: 480-400-402 P.O. Box 5052	Troy	MI 48007-5052	US	3.97	11	1
4617	4618	83029	92.64		Delphi Technologies, Inc.	Legal Staff: M/C 480-410-202 5825 Delphi Drive	Troy	MI 48098-2815	US	3.97	11	1
4617	4619				Delphi Technologies, Inc.	Legal Staff - MC 480-414-420 1450 W. Long Lake Roa	Troy	MI 48007-5052	US	4.14	11	
4617	4619	55132	98.07		Delphi Technologies, Inc.	Legal Staff, Mail Code: 480-414-420, P.O. Box 5052	Troy	MI 48007-5052	US	4.14	11	1
4617	4619	62207	92.16		Delphi Technologies, Inc.	S725 Delphi Drive, M/C 483-400-603	Troy	MI 48007	US	4.14	11	1
4617	4619	45307	92.06		Delphi Technologies, Inc.	5725 Delphi Drive	Troy	MI 48007	US	4.14	11	1
4617	4619	85519	92.02		Delphi Technologies, Inc.		Troy, Michigan	48007	US	4.14	11	1
4617	4620				Delphi Technologies, Inc.	Post Office Box 5052	Troy	MI 48007-5052	US	3.32	11	
4617	4620	55132	99.52		Delphi Technologies, Inc.	P.O. Box 5052, M/C: 483-400-402	Troy	MI 48007-5052	US	3.32	11	1
4617	4620	45307	96.54		Delphi Technologies Inc.	5725 Delphi Drive, P.O. Box 5052	Troy,Michigan	48007	US	3.32	11	1

Grouped Export
- This format is only available for self-referential searches (disambiguation) and applies (nested) cascaded traversal.
- It reports the clusters defined by the cascades without any search metrics, because cluster membership is not based on hierarchy.
- By specifying the extension ".txt", the export table will be a tab delimited text file.
- The "BaseGroupKey" can be a unique key or an entity key, if applicable, implementing variant aggregation → in case of the latter the links will be aggregated to the maximum.
- For reasons of consistency, range selection is available but only the option "Minimum hit…" should be used, as separation in confidence groups would yield inconsistent results.
- Run selection should only be used to exclude unwanted search runs but not for quality assessment as search metrics get lost during the clustering.
- The option "No singles" reduces the output to clusters with at least two members foregoing single unit clusters.
- The option "No search fields" suppresses the output of the contents of the search fields if scrutinizing of the results is not required.
- Output: the "group" variable in the export designates the cluster id, which is always the smallest "member" id.

SearchEngine Grouped Export

ExportTable
D:\searchengine\group.txt

ExportOptions
BaseGroupKey: Appgroup
Minimum hit percentage (included): 0.00
Maximum hit percentage (excluded): 101.00
☐ Skip if results greater equal the maximum exist
Run (e.g. 1, 4-7, 9):
☐ No search fields   ☐ No singles

NestedCascadedTraversal
E.g.: min >= 70 and p >= 75 or min >= 90 @ 0; min >= 80 @ 11
min: minimum identity of a link
max: maximum identity of a link (see mirror)
s: score of a link   r: number of the run
p: score percentile of a link   number: artefact threshold

p >= 75 and min >= 60 or min >= 90 @ 0, min >= 90 @ 301; min >= 85 @ 4

Export   Cancel

- Output: the "subgroup" can be used to manually separate clusters by giving each subgroup a different identifier. New cluster ids can be constructed by concatenation, i.e. 67121_1, 67121_2, or multiplication, i.e. 67121001.

13

Nested Cascaded Traversal

- Syntax: *cascade* [; *cascade* ...]
- Syntax *cascade*: [*artefact*,] *rule @ cascade_limit* [, *rule @ cascade_limit* ...]
- Syntax *rule*: logical expression using "min", "s" (score), "p" (score percentile) "r" (run) and rarely "max"
- Due to the fact that search results are not commutative, a linkage between two entities has a maximum (max) and a minimum (min) identity. Given the recommendation of high thresholds the minimum can be zero while the maximum is above the threshold. By mirroring the missing results (see function Mirror) it is possible to obtain an identity for the cases blow the threshold. Use this in conjunction with the score percentile to define more complex rules if necessary.
- Syntax *artefact:* if this number is exceeded, the link is considered an artefact and connections will be trimmed.
- A Semicolon separates nested cascades.
- Nesting is advised if heterogeneous data structure in regard of variation per entity is suspected.
- If nesting is applied, the first cascade always should have a cascade limit of zero → direct activation of the rule.
- There is no limit to the nesting of cascades but implementing more than two barely provides any benefits.
- The first cascade should define reliable clusters by strict rules with a low risk for false positives. The second cascade defines the arbitrary level of tolerated intransitivity.

- Artefact thresholds should be applied after examination of the basic export (see Extended Export), which is also helpful to assess limits when using the absolute score "s".

GROUP	MEMBER	SUBGROUP	NAME	STREET	CITY	ZIP	COUNTRY	CNT
42122	42122		Panavision, Inc.	6219 De Soto Avenue	Woodland Hills	CA 91367	US	5
42122	42123		Panavision Inc.	6219 De Soto Avenue	Woodland Hills	CA 91367	US	5
42122	42124		Panavision Federal Systems, Llc	6219 De Soto Avenue	Woodland Hills	CA 91367	US	5
42122	47555		Panavision Inc.A Corporation of the State of Delaware,U.S.A.	6219 De Soto Avenue	Woodland Hills, California	91367,U.S.A.	US	5
42122	47556		Panavision Inc.A Corporation of the State of Delaware,U.S.A.	6219 De Soto Avenue	Woodland Hills, California	91367	US	5
43951	43951		Liebherr-Werk Biberach GmbH	Hans-Liebherr-Strasse 45	Biberach an der Riss	88400	DE	5
43951	43952		Liebherr-Werk Biberach GmbH	Hans-Liebherr-Strasse 45	Biberach	88400	DE	5
43951	43953		Liebherr-Components Biberach GmbH	Hans-Liebherr-Strasse 45	Biberach/Riß	88400	DE	5
43951	43954		Liebherr-Werk Biberach GmbH	Memminger Straße 120 Postfach 1663	Biberach/Riß	88400	DE	5
43951	43955		Liebherr-Components Biberach GmbH	Hans-Liebherr-Strasse 45	Biberach an der Riß	88400	DE	5
45375	45375		NOVASEP	Site EIFFEL,Boulevard de la Moselle,BP 50	Pompey	54340	FR	5
45375	45376		Novasep	Site EIFFEL, Boulevard de la Moselle, BP 50	Pompey	54340	FR	5
45375	45377		Groupe Novasep	Site Eiffel Boulevard de la Moselle	Pompey	54340	FR	5
45375	45378		Novasep	Site EIFFEL Boulevard de la Moselle	Pompey	54340	FR	5
45375	45379		Groupe Novasep SAS	82 Boulevard de la Moselle Site Eiffel BP50	Pompey	54340	FR	5
45716	45716		Karl Storz Imaging Inc.	University Business Center,175 Cremona Drive	Goleta	CA 93117	US	5
45716	45717		Karl Storz Development Corp.	175 Cremona Drive	Goleta	CA 93117	US	5
45716	45718		Karl Storz Development Corp.	175B Cremona Drive	Goleta	CA 93117	US	5
45716	45719		Karl Storz Imaging, Inc.	University Business Center 175 Cremona Drive	Goleta	CA 93117	US	5
45716	45720		Karl Storz Imaging Inc.	175 Cremona Drive	Goleta	CA 93117	US	5
45875	45875		Kabushiki Kaisha Takeuchi Seisakusho	9347, Ooaza Sakaki,Sakaki-machi	Hanishina-gun,Nagano-ken		JP	5
45875	45876		Takeuchi Mfg, Co., Ltd	9347 Sakaki Sakaki-machi Hanishina-gun	Nagano		JP	5
45875	45877		Takeuchi MFG.Co.,Ltd.	9637, Sakaki, Sakaki-machi	Hanishina-gun,Nagano	389-0601	JP	5
45875	45878		Takeuchi Mfg, Co., Ltd	9347 Oaza Sakaki Sakaki-machi	Hanishina-gun, Nagano	389-0601	JP	5
45875	45879		Takeuchi Mfg. Co. Ltd.	9347 Oaza Sakaki Sakakimachi Hanishina-gun	Nagano	389-0601	JP	5
843	843		SKF Industrial Trading & Development Company B.V.	Kelvinbaan 16 P.O. Box 50	Nieuwegein	NL-3430 AB	NL	4
843	844		S.K.F. INDUSTRIAL TRADING & DEVELOPMENT COMPANY B.V.	Kelvinbaan 16	Nieuwegein		NL	4
843	46769		SKF Engineering & Research Centre B.V.	Kelvinbaan 16	Nieuwegein	3439 MT	NL	4
843	46770		SKF BV	P.O. Box 2350	Nieuwegein	3430 DT	NL	4

SearchEngine ▶ File ▶ Export ▶ Result Export

Result Export

- This function creates a fully functional subset respectively sample of the result table. It can be used to remove unwanted runs, curtail the identities or to draw a random sample for further exports.
- The export format will always be ".dbf". Text format is not supported.
- The range selection and run filtering is equivalent to the (simple) Export dialog. The "Skip" option is not supported.
- With "Sampling", a random sample of the result table can be drawn. The entered number is considered a share if it is smaller than 1, e.g. 0.25. Alternatively, a percentage sign can be used, e.g. 25%. A number greater equal 1 designates the absolute number of cases to be drawn. A case consists of all candidates of a drawn searched entry.
- Gaps that are caused by filtered runs can be removed by compressing the run order.
- The exported result table can be manually assigned as the new result table in the "File Locations" dialog or directly by ticking the corresponding option on this dialog.
- The sampling function is intended to be used in conjunction with "Meta Export". It provides a training dataset for machine learning approaches to reduce the workload of manual checking. By temporarily replacing the original result table, an extended export file and the corresponding meta file can be created.

SearchEngine Result Export

ResultTable
D:\searchengine\resultsample.dbf [...]

ExportOptions
Minimum hit percentage (included): 0.00
Maximum hit percentage (excluded): 101.0
Run (e.g. 1, 4-7, 9):
Enter a sample size as count or share, e.g. 850 or 5%
Sampling: 10%
Remove gaps in run code ☐
Use as new ResultTable ☑

[Export] [Cancel]
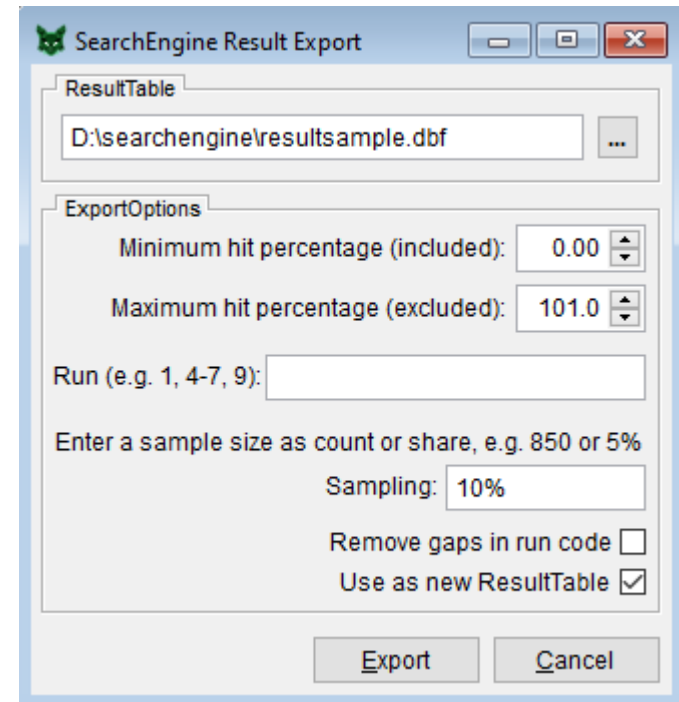
- If you have switched the result table with a sample, don't forget to revert to the original in the "File Location" dialog after the exports.

Meta Export

- This function exports meta information from the result table consisting of the harmonized occurrences of the matching words of the search term and the candidate (M), words that are exclusive to the search term (S) and surplus words in the found candidates (F). For every search type fields of the M, S and F category are created. The number of fields depends on the Meta specification (see below).
- The export table will be a tab delimited text file if the extension is ".txt".
- The range selection and run filtering is equivalent to the (simple) Export dialog. The "Skip" option is not supported.
- The Meta option specifies how many words of a specific search type will be exported. If nothing is specified, the default of 5 fields for every search type and every category will be used. A specification of "1-99 = 7; 2 = 0; 3 = 4; 4-6 = 1" would first set 7 fields for all search types, skip search type 2, sets search type 3 to 4 fields and types 4 to 6 to only one occurrence per category. The maximum of a range specification can exceed the actual number of search types to simplify the definition of defaults.
- The harmonized occurrences have a value range between 0 and 1. A zero means that the word was not used or has the highest occurrence, while a 1 represents a unique word:

$$harmocc = 1 - (occurs - 1)/maxocc$$

- By default all values are normalized because harmonization does not guarantee the utilization of the full range [0,1]. The normalization can be skipped with the last option.
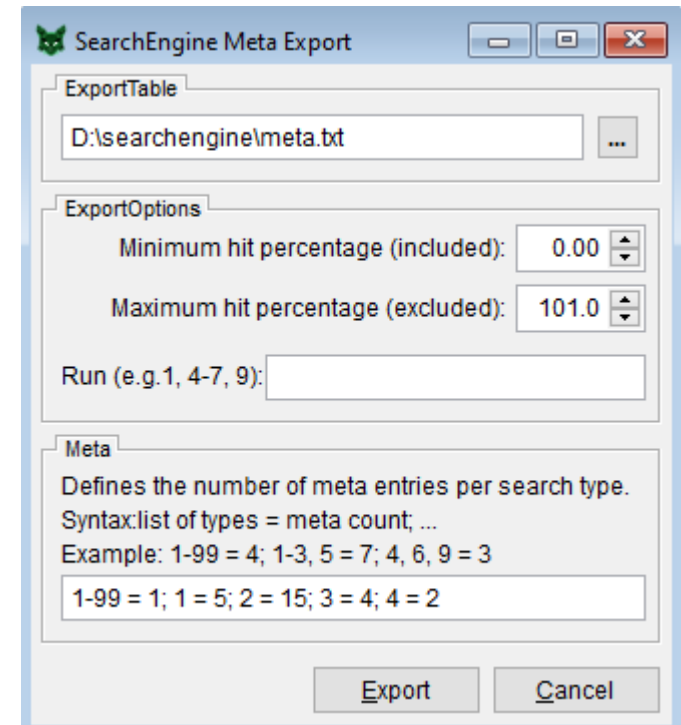

- For the F category two harmonized occurrences will be evaluated. One is based on a Registry of the search table, while the other uses the occurrence and maximum occurrence of the base Registry. The lower of both normalized values will be used. Because of the different bases in the F category, all occurrences are harmonized.

17

Meta Export

- Besides the occurrence fields, a run dummy vector (run1, run2…), the identity, the score, the number of candidates for a search term (cnt), the number of distinct identities (icnt) and the relative position within these identities (ipos) are reported. Furthermore, the "R" category specifies the overlap between non-destructive search types.
- With the "No normalization" option, the meta data can be exported in raw format to apply external normalization routines. The occurrence fields stay harmonized.
- The Registry of the SearchTable required for the "F" category will be created only once per SearchTable to speed up subsequent exports, i.e. in the case of samples (see Result Export). The files are located in the SearchTable directory with the postfixes "*_registry.dbf" and "*_registry.cdx".
- The Meta Export is intended to be used for machine learning (ML) approaches. Only a sample of the results needs to be manually scrutinized. The meta information of the sample can be used to train a probabilistic model, i.e. Neural Network, Logit, Random Forest etc. The model predicts the outcome of the matches for the whole data. The SearchEngine provides a method to block the data efficiently, while the final match is done by the ML approach.
- The directory SEML contains sample programs based on the STATA package "brain": https://github.com/ThorstenDoherr/brain
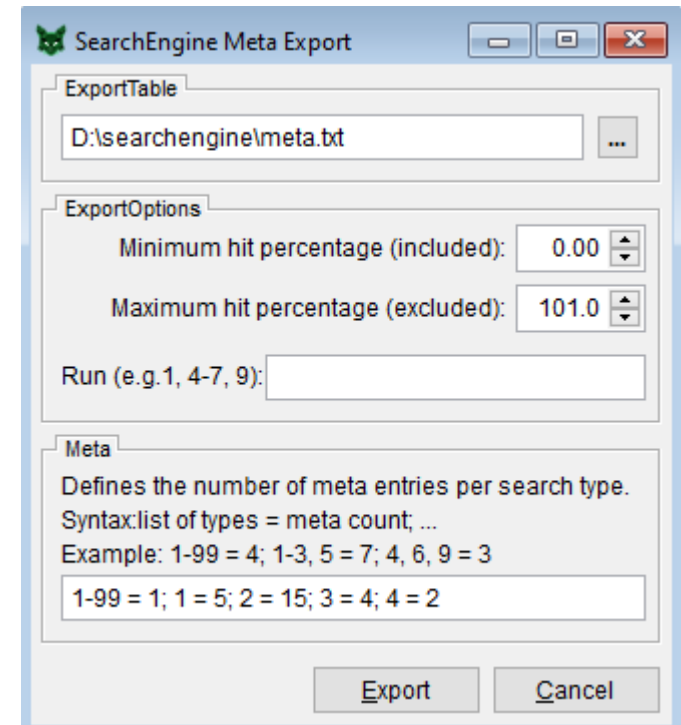
- The Result Export with the Sampling option can be used to draw a random selection as a temporary Result table to export the corresponding Meta information and Extended Export file as training data. The Meta information of the sample is compatible to the corresponding Meta information of the complete data.

SEARCHED	FOUND	IDENTITY	SCORE	CNT	ICNT	IPOS	RUN1	RUN2	RUN3	M1_1	M1_2	M1_3	M1_4	M1_5	M1_6	M1_7	F1_1	F1_2 ...
32	849	1	0.01987094	0.09090909	0.16666667	1	1	0	0	0.9991895	0.96920084	0	0	0	0	0	0	0
32	850	1	0.01987094	0.09090909	0.16666667	1	1	0	0	0.9991895	0.96920084	0	0	0	0	0	0	0
32	848	0.64238095	0.01987094	0.09090909	0.16666667	0.5	1	0	0	0.9991895	0	0	0	0	0	0	0.52358567	0
33	195	1	0.29223537	0	0	1	1	0	0	1	0.99991895	0.9998379	0.99230021	0	0	0	0	0
34	201	1	0.03023705	0.09090909	0.33333333	1	1	0	0	0.99959475	0.86124169	0.00008105	0	0	0	0	0	0
34	202	0.66666667	0.03023705	0.09090909	0.33333333	0.66666667	1	0	0	0.99959475	0.86124169	0.00008105	0	0	0	0	0	0
34	200	0.64812562	0.03023705	0.09090909	0.33333333	0.33333333	1	0	0	0.99959475	0	0	0	0	0	0	0.55722159	0
35	202	0.93783827	0.02834079	0.09090909	0.33333333	1	1	0	0	0.99959475	0.86124169	0.00008105	0	0	0	0	0	0
35	200	0.91929722	0.02834079	0.09090909	0.33333333	0.66666667	1	0	0	0.99959475	0	0	0	0	0	0	0.55722159	0
35	201	0.60450494	0.02834079	0.09090909	0.33333333	0.33333333	1	0	0	0.99959475	0.86124169	0.00008105	0	0	0	0	0	0
36	216	1	0.05374211	0.09090909	0.16666667	1	1	0	0	0.9998379	0.79615821	0	0	0	0	0	0	0
36	215	1	0.05374211	0.09090909	0.16666667	1	1	0	0	0.9998379	0.79615821	0	0	0	0	0	0	0
36	217	0.64435876	0.05374211	0.09090909	0.16666667	0.5	1	0	0	0.9998379	0.79615821	0	0	0	0	0	0	0
37	223	0.89384123	0.00880646	0.27272727	1	1	1	0	0	0.99756849	0.99148971	0.41449181	0	0	0	0	0	0
37	222	0.89341354	0.00880646	0.27272727	1	0.85714286	1	0	0	0.99756849	0.99148971	0.41449181	0	0	0	0	0	0
37	219	0.87941189	0.00880646	0.27272727	1	0.71428571	1	0	0	0.99756849	0.99148971	0.41449181	0	0	0	0	0	0
37	224	0.68109601	0.00880646	0.27272727	1	0.57142857	1	0	0	0.99756849	0.99148971	0.41449181	0	0	0	0	0	0
37	221	0.66666667	0.00880646	0.27272727	1	0.42857143	1	0	0	0.99756849	0.99148971	0.41449181	0	0	0	0	0	0
37	220	0.5605079	0.00880646	0.27272727	1	0.28571429	1	0	0	0.99756849	0.99148971	0.41449181	0	0	0	0	0	0
37	218	0.33333333	0.00880646	0.27272727	1	0.14285714	1	0	0	0.99756849	0.99148971	0.41449181	0	0	0	0	0	0
38	29955	1	0.0705355	0	0	1	1	0	0	0.9998379	0.97536067	0	0	0	0	0	0	0
39	235	1	0.02662092	0.04545455	0.16666667	1	1	0	0	0.99910845	0.99764954	0.99708219	0.99440752	0.98395202	0.97228076	0.97203534	0	0
39	234	0.60354459	0.02662092	0.04545455	0.16666667	0.5	1	0	0	0.99910845	0.99764954	0.99708219	0.99440752	0.98395202	0.97228076	0.97203534	0.99975685	0
40	65572	0.71631734	0.01277256	0.31818182	0.33333333	1	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0.99116551	0.9794132
40	245	0.71631734	0.01277256	0.31818182	0.33333333	1	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0	0
40	242	0.71631734	0.01277256	0.31818182	0.33333333	1	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0	0
40	243	0.71631734	0.01277256	0.31818182	0.33333333	1	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0	0
40	244	0.71631734	0.01277256	0.31818182	0.33333333	1	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0	0
40	248	0.71579356	0.01277256	0.31818182	0.33333333	0.66666667	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0	0
40	246	0.70862226	0.01277256	0.31818182	0.33333333	0.33333333	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0	0
40	247	0.70862226	0.01277256	0.31818182	0.33333333	0.33333333	1	0	0	0.99805479	0.00008105	0	0	0	0	0	0	0
41	3456	1	0.03239203	0.18181818	0.5	1	1	0	0	0.9996758	0.96806614	0.38490841	0.26762847	0	0	0	0	0
41	3454	0.97912973	0.03239203	0.18181818	0.5	0.75	1	0	0	0.9996758	0.96806614	0.38490841	0.26762847	0	0	0	0	0
41	3455	0.97912973	0.03239203	0.18181818	0.5	0.75	1	0	0	0.9996758	0.96806614	0.38490841	0.26762847	0	0	0	0	0
41	3457	0.97654314	0.03239203	0.18181818	0.5	0.5	1	0	0	0.9996758	0.96806614	0.38490841	0.26762847	0	0	0	0	0
41	3453	0.97629964	0.03239203	0.18181818	0.5	0.25	1	0	0	0.9996758	0.96806614	0.38490841	0.26762847	0	0	0	0	0
43	279	1	0.04121571	0.22727273	0.5	1	1	0	0	0.9990274	0.9900308	0.95720538	0.38490841	0.00008105	0	0	0	0
43	278	0.68206461	0.04121571	0.22727273	0.5	0.75	1	0	0	0.9990274	0.9900308	0.95720538	0.38490841	0.00008105	0	0	0	0
43	2461	7.0301E-05	0.04121571	0.22727273	0.5	0.5	1	0	0	0.9990274	0.9900308	0.95720538	0.38490841	0.00008105	0	0	0	0
43	2463	0	0.04121571	0.22727273	0.5	0.25	1	0	0	0.9990274	0.9900308	0.95720538	0.38490841	0.00008105	0	0	0	0
43	2464	0	0.04121571	0.22727273	0.5	0.25	1	0	0	0.9990274	0.9900308	0.95720538	0.38490841	0.00008105	0	0	0	0
43	2462	0	0.04121571	0.22727273	0.5	0.25	1	0	0	0.9990274	0.9900308	0.95720538	0.38490841	0.00008105	0	0	0	0
44	281	1	0.06677876	0.04545455	0	1	1	0	0	0.99975685	0.99910845	0.96920084	0	0	0	0	0	0
44	280	1	0.06677876	0.04545455	0	1	1	0	0	0.99975685	0.99910845	0.96920084	0	0	0	0	0	0
45	1350	0.66760029	0.03350184	0.09090909	0.33333333	1	1	0	0	0.99862214	0.99764954	0.99456962	0.99424542	0.99416437	0.98711298	0.98613926	0	0

BaseTable

- It is usually a large (largest) table or the table that encompasses the most noise.
- It constitutes the heuristic (registry) and should not be exchanged.
- Harmonization is not required but good practice.
- Should be free of duplicate records regarding the search fields.
- Accepts tab-delimited text files (*.txt), which will be transferred to Foxpro (*.dbf).
- Unique keys are not required if record numbers suffice as reference.

SearchTable

- Usually a smaller table (or with less noise).
- Can be exchanged.
- Besides these points: same as above.

ResultTable

- Container for the candidates.
- Should reside in the same directory as the SearchTable because it only contains candidates of one SearchTable.
- Is always a Foxpro table (*.dbf).
- For internal purposes only. Use export functions to access the search results.

SearchEngine Files

BaseTable
`D:\searchengine\applicants.dbf`

SearchTable
`D:\searchengine\firms.dbf`

ResultTable
`d:\searchengine\firms_result.dbf`

Import Options
- Decode unicode characters ☐
- No memo fields ☐
- Fast parsing ☐

OK   Cancel

Import Options

- Options for importing tab-delimited text files with header (*.txt).
- No memo fields: max field size of 254 chars (ticked) → Faster execution of all functions but loss of information.
- Decode single unicode characters: transformation of 2 byte chars to extended ASCII chars.
- Fast parsing: is faster but may lead to truncated fields.
- Unicode files are not supported. UTF-8 is supported.
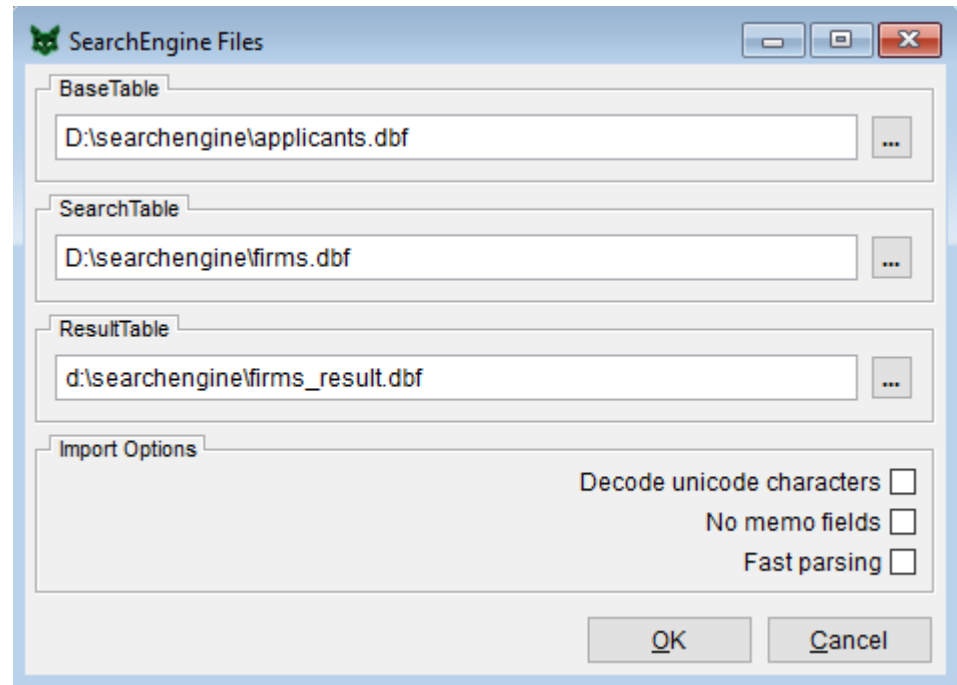
TableCluster
- Foxpro tables have a maximum size of 2GB.
- TableCluster are Foxpro tables with the same name template containing a sequence number, i.e. firms2017_1.dbf, firms2017_2.dbf, firms2017_3.dbf, …
- A sequence is always the rightmost separated number of a table name, starts always with 1 and has no leading zeros.
- All tables within a cluster have the same structure as they represent a virtual table that can be much larger than 2GB.

Tab-delimited text files (*.txt)
- Imported text files may compile into several cluster tables with a sequence number.
- As long as the imported Foxpro tables exists, they will be used instead of a repeated import. If a re-import is required, delete the corresponding Foxpro tables manually.
- Associated with a Foxpro table are all similar named files with the extensions: *.dbf, *.cdx, *.fpt, *.bak, *.fbk. The extensions *.bak and *.fbk are reserved for restoration after structural changes to a table and can be deleted without consequences.

SearchEngine Files

BaseTable

D:\searchengine\applicants.dbf

SearchTable

D:\searchengine\firms.dbf

ResultTable

d:\searchengine\firms_result.dbf

Import Options

Decode unicode characters ☐
No memo fields ☐
Fast parsing ☐

OK   Cancel

- The text file should only contain fields required for the search, i.e. unique identifier plus search fields.
- Make sure that the tab-delimited text files do not contain tabulators and other control characters in the data itself, for example by applying the following STATA code:

```
replace var = subinstr(var,char(9)," ",.) if strpos(var,char(9)) > 0
replace var = subinstr(var,char(13)," ",.) if strpos(var,char(13)) > 0
replace var = subinstr(var,char(10)," ",.) if strpos(var,char(10)) > 0
replace var = subinstr(var,`""'," ",.) if strpos(var,`""') > 0
replace var = subinstr(var,`""'," ",.) if strpos(var,`""') > 0
```

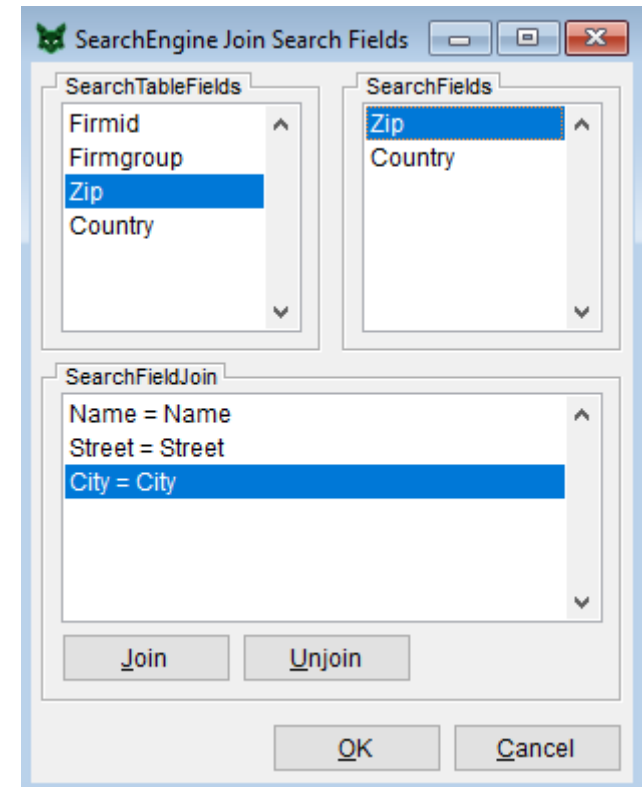# SearchEngine ▶ Config ▶ Join Search Fields

SearchTableFields
- Lists all fields of the search table.
- The context of the selected field should be fully contained in the context of the correspondingly selected search field, i.e. city → address field including street, zip and city.
- Lowered identities are the consequence, if the context of a search table field exceeds the context of its search field.
- If necessary, adjust fields of the search table according to the search fields by joining or separating contents

SearchFields
- Lists all base table fields used to define search types.
- A search type is just a specific way to harmonize a search field, therefore only the parent field can be linked but not the type.

SearchFieldJoin
- Lists all linked search and base table fields.
- Not every search field needs to be linked, as the search table may not have a matching field for every one, e.g. zip missing.
- "Join" links fields, "Unjoin" removes a link (removed fields are appended to the end of the respective list).
- The order of the links determines also the field order of some export formats and can be changed accordingly without further consequences for the search process.
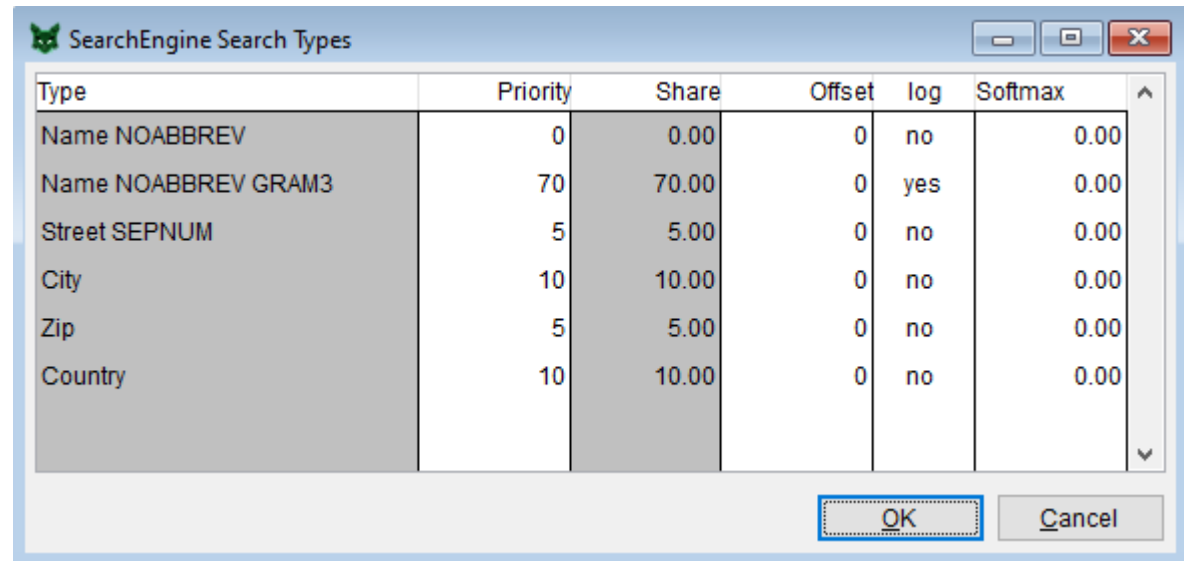
- Sometimes it is helpful to create empty search table fields to be able to link them before conducting an export. As a result, at least the base field contents will be reported.

SearchEngine ▶ Config ▶ Search Types

SearchTypes

- List of all search types with the current setting for the weight (share), offset, logarithmic and softmax smoothing.
- The priority will be redistributed to shares, which are the weights for the search types (chapters of the registry).
- The offset accepts negative values.
- Click in the "log" column switches log smoothing off/on (yes/no).

Type	Priority	Share	Offset	log	Softmax
Name NOABBREV	0	0.00	0	no	0.00
Name NOABBREV GRAM3	70	70.00	0	yes	0.00
Street SEPNUM	5	5.00	0	no	0.00
City	10	10.00	0	no	0.00
Zip	5	5.00	0	no	0.00
Country	10	10.00	0	no	0.00

SearchEngine Search Types — OK   Cancel

Search strategies

- The SearchEngine supports incremental search runs with different settings, which will be merged into the result table.
- Do not try to find everything with one go.
- Concentrate the weights on the identifying search field(s) (priority field), i.e. firm name, and play around with the weights for the remaining fields.
- Take the impact of the threshold for the candidate identities into account: a threshold equal to the weight of the priority field means that a perfect match for this field is good enough, but if the supporting fields also contribute to the result, the constrains for the priority field are relaxed.

- The first couple of runs should be based on non-destructive preparer by setting the weights of destructive types (gram, soundex, …) to zero (see screen capture).
- Follow the basic runs with destructive runs to catch the misspellings by switching the priorities.
- When using grams, conduct one run with and one without logarithmic smoothing (or softmax < 1).
- Every candidate in the result table is designated with a sequential run number to retrace the impact of different settings in the export files.
- See Presentation for more information about Softmax.

SearchEngine ▶ Config ▶ Settings

Limit and Cutoff
- Limit defines the threshold for the candidate identities.
- The cutoff allows to reduce excessive candidate lists by adjusting the threshold to the identity of the candidate at the cutoff position.
- Adjust the cutoff according to the amount of expected duplicates in the base data.
- Think about which weight combinations of search types would make it to the limit.

Feedback and Activation
- Feedback regulates how much the candidate identities resemble a Jaccard index.
- Only if the number of candidates reaches the activation limit, feedback will be applied.
- If both, cutoff and activation, are greater than zero, the feedback introduces variation into the candidate list to bring the candidates with the least amount of relevant noise on top. This is only used for temporary ranking and does not change the final candidate identities.

SearchFlags
- Relative Search redistributes the weights if search fields are missing in a search term.
- Darwinistic keeps only the candidates with the highest identities → clean base table required.

SearchEngine Settings

Threshold and Cutoff

Minimum required identity for candidates. Lower results will be excluded. — 85.00

The threshold will be adjusted to the candidate at the cutoff position (0 = no cutoff). — 20

Feedback and Activation

The percentage of the feedback effect on surplus words in candidates (0 = no feedback). — 10.00

Feedback will be activated when the number of candidates reaches this number. — 20

SearchFlags

Relative search (missing search fields will be ignored) ☐
Darwinistic search (only the best result survives) ☐
Ignorant to unknown words in searched records ☐
Zealous (ignores limit to get results) ☐

OK   Cancel

- Ignorant gives words not found in the registry an identification potential of zero instead of the average IP, potentially leading to matching with very weak terms like legal forms → overblown candidate lists.
- Zealous picks the next best candidates, if none would regularly reach the threshold → unexpected results.

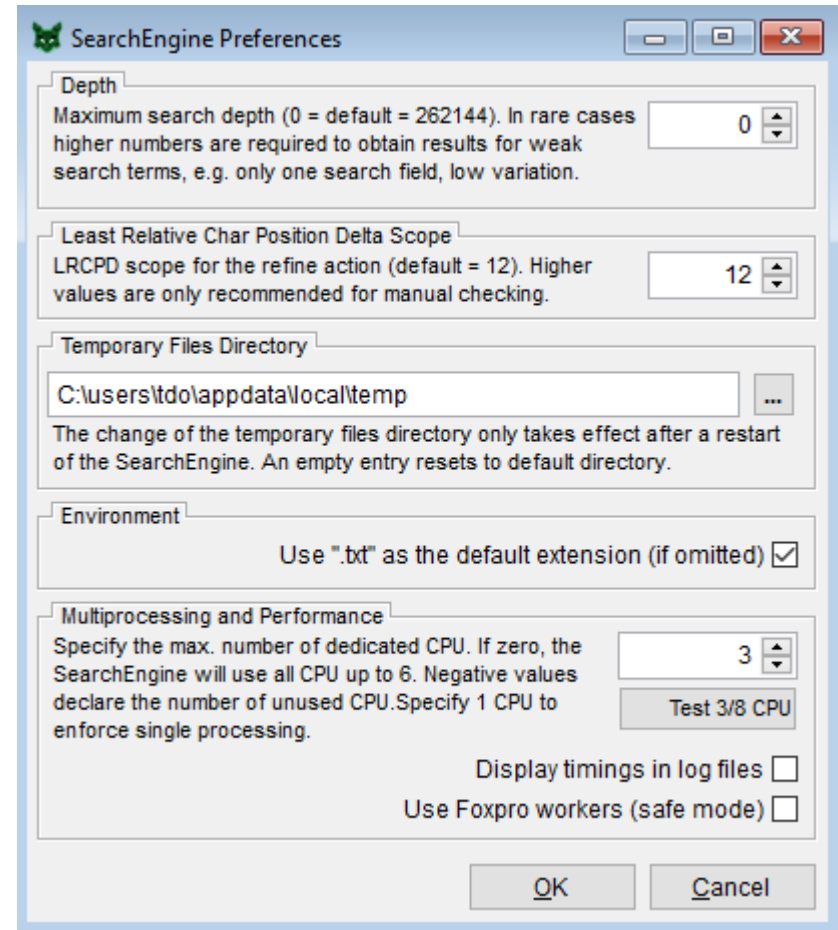# SearchEngine ▶ Config ▶ Preferences

Preferences
- Default values: depth = 0, LRCPD scope = 12

Depth
- Depth defines the size of an internal buffer of up to 8388608 entries.
- The default depth of 262144 is selected when no depth is explicitly specified (depth = 0), which is suitable for most applications.
- If there is no word in a search term with an occurrence lesser equal this number, the SearchEngine will not be able to retrieve candidates.
- Lower numbers speed up the search considerably but may lead to missed candidates. Higher numbers slow down the search but may be beneficial if weak search terms are expected, e.g. only one search field with low variation.

Least Relative Char Position Delta Scope
- The LRCPD algorithm overstates the similarity for long strings as the deltas decrease with the string size.
- The LRCPD scope defines the maximum distance the algorithm is looking for a matching character and adjusts the deltas accordingly.
- Larger scopes increase the tolerance of the string comparison algorithm.

SearchEngine Preferences

Depth
Maximum search depth (0 = default = 262144). In rare cases higher numbers are required to obtain results for weak search terms, e.g. only one search field, low variation.

`0`

Least Relative Char Position Delta Scope
LRCPD scope for the refine action (default = 12). Higher values are only recommended for manual checking.

`12`

Temporary Files Directory

`C:\users\tdo\appdata\local\temp`

The change of the temporary files directory only takes effect after a restart of the SearchEngine. An empty entry resets to default directory.

Environment
Use ".txt" as the default extension (if omitted) ☑

Multiprocessing and Performance
Specify the max. number of dedicated CPU. If zero, the SearchEngine will use all CPU up to 6. Negative values declare the number of unused CPU. Specify 1 CPU to enforce single processing.

`3`

Test 3/8 CPU

Display timings in log files ☐
Use Foxpro workers (safe mode) ☐

OK   Cancel

SearchEngine ▶ Config ▶ Preferences

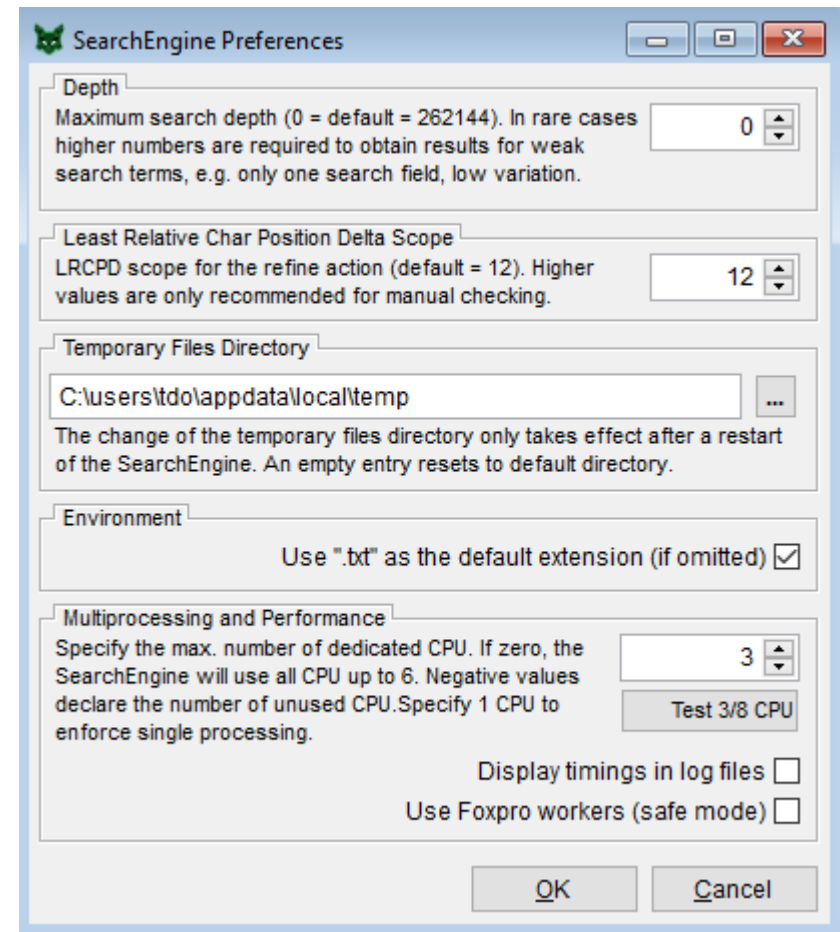**Temporary Files Directory**
- The location for the temporary work space can be changed in case of right issues or insufficient space at the default directory. The change will take effect after a restart of the SearchEngine.

Environment
- If the ".txt" option is checked, the SearchEngine will consider text files as the default format making imports and exports more convenient.

Multiprocessing
- A positive number determines how many CPUs (cores) will be dedicated to the SearchEngine process. A negative number determines how many CPUs stay free for other tasks. The default is zero, which reserves up to 6 CPUs to the SearchEngine.
- The efficiency of multiprocessing (MP) can be assessed by a benchmark test. MP has diminishing returns to performance caused by the file system.
- Benchmark timings can be displayed as comments in the log file "searchengine.log" or a log file assigned to a script (see Batch mode).
- The multiprocessing is based on workers, which show as "parallelfox.exe" in the task manager. These will be registered in the Windows registry on startup of the SearchEngine. If that is not possible because of missing admin rights, Foxpro workers are used which appear in the task bar. This behavior can be manually enforced. The disadvantage of Foxpro workers is that they can be accidentally closed.
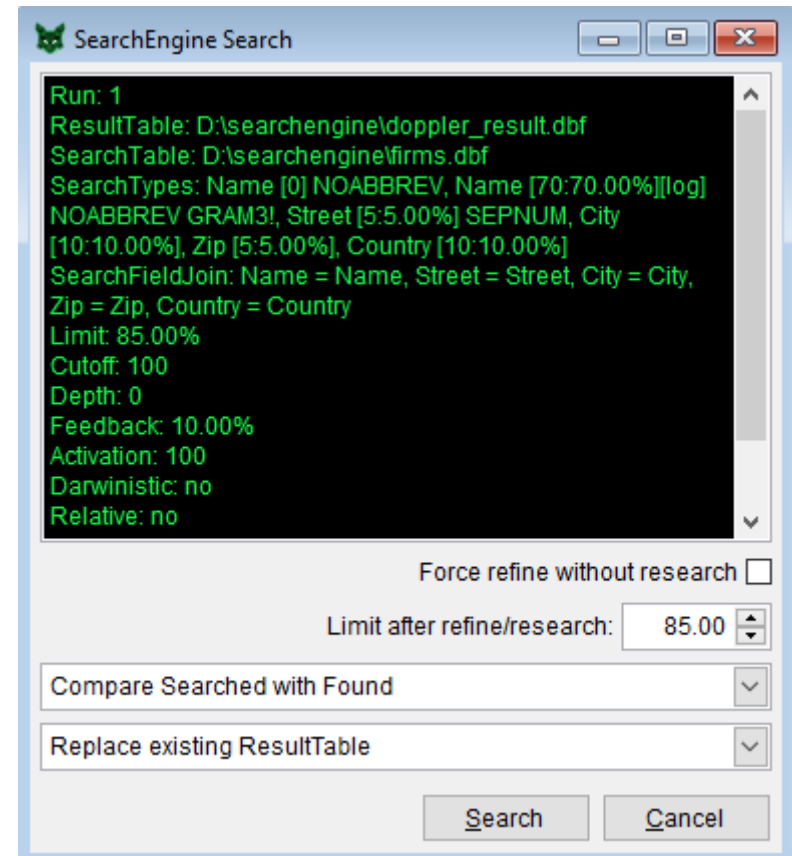
SearchEngine Preferences

Depth
Maximum search depth (0 = default = 262144). In rare cases higher numbers are required to obtain results for weak search terms, e.g. only one search field, low variation. `0`

Least Relative Char Position Delta Scope
LRCPD scope for the refine action (default = 12). Higher values are only recommended for manual checking. `12`

Temporary Files Directory
`C:\users\tdo\appdata\local\temp` `...`
The change of the temporary files directory only takes effect after a restart of the SearchEngine. An empty entry resets to default directory.

Environment
Use ".txt" as the default extension (if omitted) ☑

Multiprocessing and Performance
Specify the max. number of dedicated CPU. If zero, the SearchEngine will use all CPU up to 6. Negative values declare the number of unused CPU.Specify 1 CPU to enforce single processing. `3`
Test 3/8 CPU

Display timings in log files ☐
Use Foxpro workers (safe mode) ☐

OK Cancel

Search
- Shows information relevant for the search.
- "Force refine", "Limit after refine" and "Compare" options will be explained in Search [2].
- "Replace existing ResultTable" erases the existing results and starts a fresh result table → run counter reset to one.
- "Complete with results for unmatched search records" conducts the search only for search table records that have no candidates yet in the result table → filling the gaps.
- "Merge results with existing ResultTable" conducts a complete search and creates a union of existing and new results → enriching the results.
- "Continue a canceled search" looks for the candidate with the highest search record number for the current run and continues from there → after canceling a search midways.
- "Complete by replacing last run" deletes all results of the last run and "fills the gaps".
- "Merge by replacing last run" deletes all results of the last run and "enriches the results".
- Every result table has its own run counter which will be incremented with every search using the "Complete…" or "Merge…" option
- All candidates found during a match are marked with the corresponding run number.
- Some functions like Export or Research can be restricted to specific runs using the same syntax as the page selection in word, i.e. 1, 2, 4-7.

SearchEngine Search

```
Run: 1
ResultTable: D:\searchengine\doppler_result.dbf
SearchTable: D:\searchengine\firms.dbf
SearchTypes: Name [0] NOABBREV, Name [70:70.00%][log]
NOABBREV GRAM3!, Street [5:5.00%] SEPNUM, City
[10:10.00%], Zip [5:5.00%], Country [10:10.00%]
SearchFieldJoin: Name = Name, Street = Street, City = City,
Zip = Zip, Country = Country
Limit: 85.00%
Cutoff: 100
Depth: 0
Feedback: 10.00%
Activation: 100
Darwinistic: no
Relative: no
```

Force refine without research ☐

Limit after refine/research: 85.00

Compare Searched with Found

Replace existing ResultTable

Search   Cancel

- Use [ESC] to prematurely cancel the search to assess the quality of the matches and the search strategy before a long run → continue or replace by using the respective option.

Refine and Research

- When destructive preparer are involved (weight > 0) additional options become available.
- Similarity index of a string comparison of "destroyed" search fields replaces the corresponding components of the candidate identity based on the relative identification potential (rIP). Because it is not possible to subsequently disentangle the candidate identities, they will be set to zero followed by a "refine" step adding the string comparison component and a "research" step adding the rIP component according to the corresponding search type weights.
- With "Force refine without research" the refine step will be conducted for all fields making the research step obsolete.
- The refine process, like the main heuristic, is not commutative, therefore the direction of the comparison has to be specified:
  - "Compare Searched with Found" is the default direction and mimics the general SearchEngine behavior.
  - "Dynamic compare" compares in both directions and uses the lowest result → suitable for person names.
  - "Compare Found with Search" reverses the default direction → more noise in the base table.
- After refine and research a new limit for the candidate identities can be specified to separate the candidate selection from the final evaluation.

SearchEngine Search

```
Run: 1
ResultTable: D:\searchengine\doppler_result.dbf
SearchTable: D:\searchengine\firms.dbf
SearchTypes: Name [0] NOABBREV, Name [70:70.00%][log]
NOABBREV GRAM3!, Street [5:5.00%] SEPNUM, City
[10:10.00%], Zip [5:5.00%], Country [10:10.00%]
SearchFieldJoin: Name = Name, Street = Street, City = City,
Zip = Zip, Country = Country
Limit: 85.00%
Cutoff: 100
Depth: 0
Feedback: 10.00%
Activation: 100
Darwinistic: no
Relative: no
```

Force refine without research ☐

Limit after refine/research: 85.00

Compare Searched with Found

Replace existing ResultTable

Search   Cancel

- In most cases, the default settings regarding refine and research should be fine.

SearchEngine ▶ Action ▶ Research

Research
- This function can be used to harmonize the candidate identities of several search runs based on one general search parametrization. It recalculates the candidate identities for selected runs according to the current parameters like weights, offsets, logarithmic smoothing, feedback and the "relative" and "ignorant" settings → no new results will be generated.
- Runs can be selected using the same syntax as the page selection before printing, e.g. 1-3,5,7.
- "Always update identity", "Never update identity", "Maximize identity", "Minimize identity", "Increment identity" and "Average identity" define how the new identities interact with the existing identities.
- Always update score", "Never update score", "Maximize score" and "Minimize score" define how the new scores interact with the existing scores.
- Identities will be truncated at 100%.
- To replicate the refine and research steps of a search using destructive preparer choose the option "Only active SearchTypes…" to exclude search types containing destructive preparer and "Always update identity". A subsequent Refine step is required (see Refine).

SearchEngine Research

SearchTable: D:\searchengine\firms.dbf
ResultTable: D:\searchengine\doppler_result.dbf
SearchTypes: Name [70:70.00%][log] NOABBREV GRAM3!, Street [5:5.00%] SEPNUM, City [10:10.00%], Zip [5:5.00%], Country [10:10.00%]
SearchFieldJoin: Name = Name, Street = Street, City = City, Zip = Zip, Country = Country
ShareSum: 100.00%
Feedback: not used (activation > 0)
Relative: no
Ignorant: no

Run (e.g. 1, 4-7, 9):

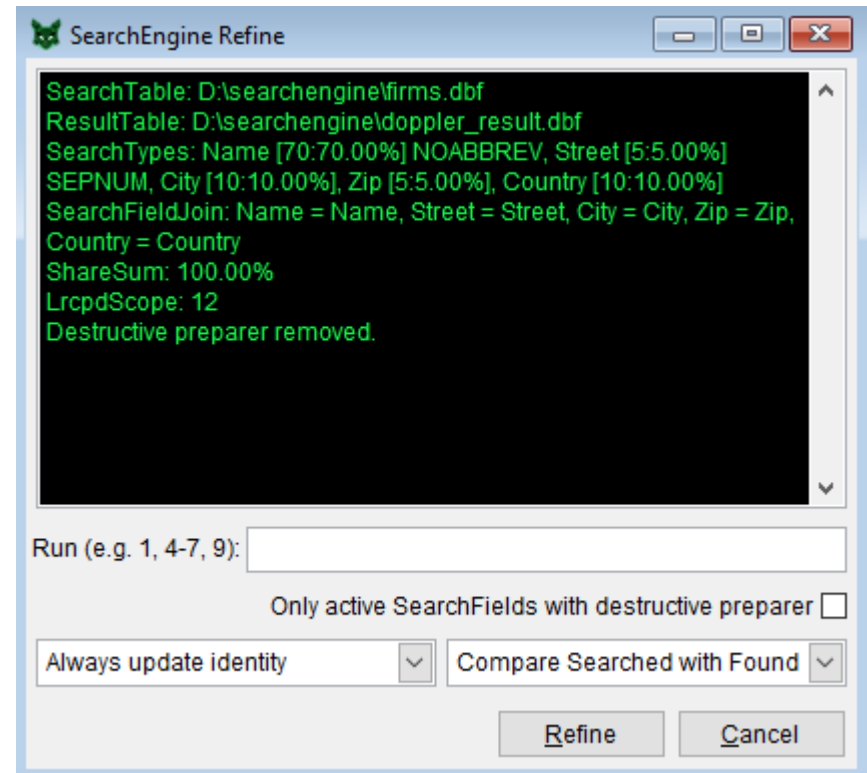Only active SearchTypes with non-destructive preparer ☐

Always update identity ⌄   Always update score ⌄

Research   Cancel

SearchEngine ▶ Action ▶ Refine

Research
- This function can be used to replace the frequency based heuristic of the candidate identities with a string comparison based similarity index. It recalculates the candidate identities for selected runs using the LRCPD string comparison method → no new results will be generated.
- Runs can be selected using the same syntax as the page selection before printing, e.g. 1-3,5,7.
- To guarantee a comparison close to the original data all destructive preparer will be deactivated.
- "Always update identity", "Never update identity", "Maximize identity", "Minimize identity", "Increment identity" and "Average identity" define how the new identities interact with the existing identities.
- "Compare Searched with Found", "Dynamic compare" and "Compare Searched with Found" specifies the direction of the non-commutative comparison.
- Identities will be truncated at 100%.
- To replicate the refine and research steps of a search using destructive preparer choose the option "Only active SearchFields…" to exclude search fields containing only non-destructive preparer and "Increment identity" to combine both identity components (see Research).

SearchEngine Refine

SearchTable: D:\searchengine\firms.dbf
ResultTable: D:\searchengine\doppler_result.dbf
SearchTypes: Name [70:70.00%] NOABBREV, Street [5:5.00%] SEPNUM, City [10:10.00%], Zip [5:5.00%], Country [10:10.00%]
SearchFieldJoin: Name = Name, Street = Street, City = City, Zip = Zip, Country = Country
ShareSum: 100.00%
LrcpdScope: 12
Destructive preparer removed.

Run (e.g. 1, 4-7, 9):

Only active SearchFields with destructive preparer ☐

Always update identity   Compare Searched with Found

Refine   Cancel

SearchEngine ▶ Action ▶ Create

BaseTableFields
- Lists all field names of the base table, ready to be declared as search field (arrow right).

SearchTypes
- Combination of a search field and a preparer is called search type.
- A search field can occur more than once with different preparer combination.
- Plan search strategies ahead: you can mix and merge multiple search runs on the same search table into one result table.
- It is advised to link n-gram preparer only to search fields that will get a high weight, because they slow the search down and should therefore be worth it.
- Not any preparer combination will work, but you will get no warning.
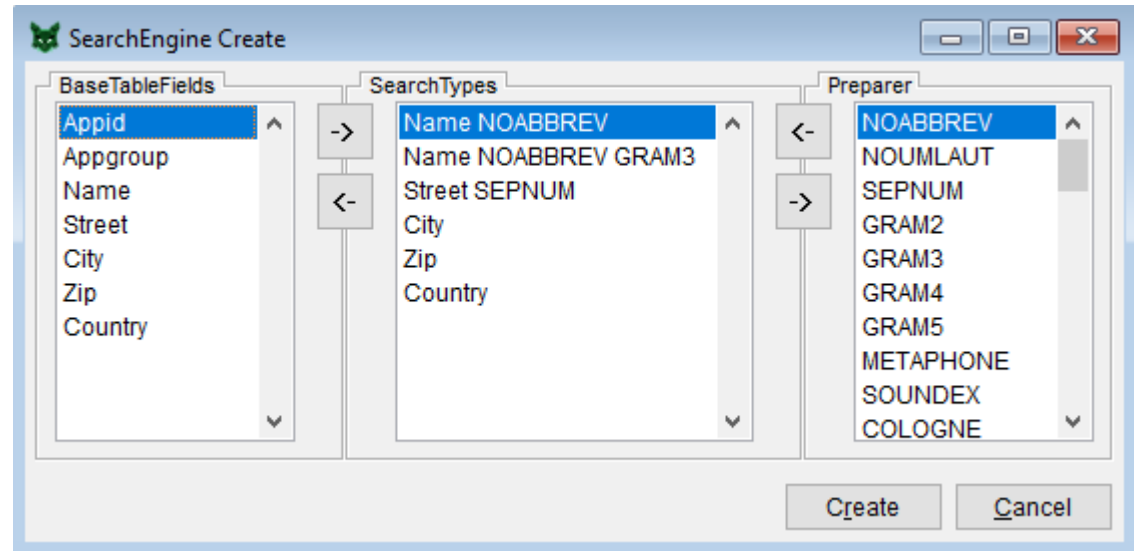
Preparer
- Basic harmonization (upper case and so on) always takes place.
- Left arrow to link it, right arrow to remove it from search type.
- NOABBREV: gather single letters into one word (B A S F → BASF)
- NOUMLAUT: change normal German umlaut transformation into a simplified one, typical for US data (Ä→AE →A).
- SEPNUM: separate numbers from letters (house numbers).
- GRAMn: implements the n-gram method (effective but inefficient).
- METAPHONE, SOUNDEX, COLOGNE: implements these methods.
- MAXLENGTHn: truncate all words to length n.
- MAXWORDSn: ignore all words after word n, e.g. selecting first name.
- SKIPWORDSn: ignore the leftmost n words, e.g. selecting last name.
- Red bullet points mark destructive preparer, green are non-destructive.

SearchEngine ▶ Action ▶ Recreate

Recreate
- Deletes the current registry and all associated tables to replace it with a new registry.
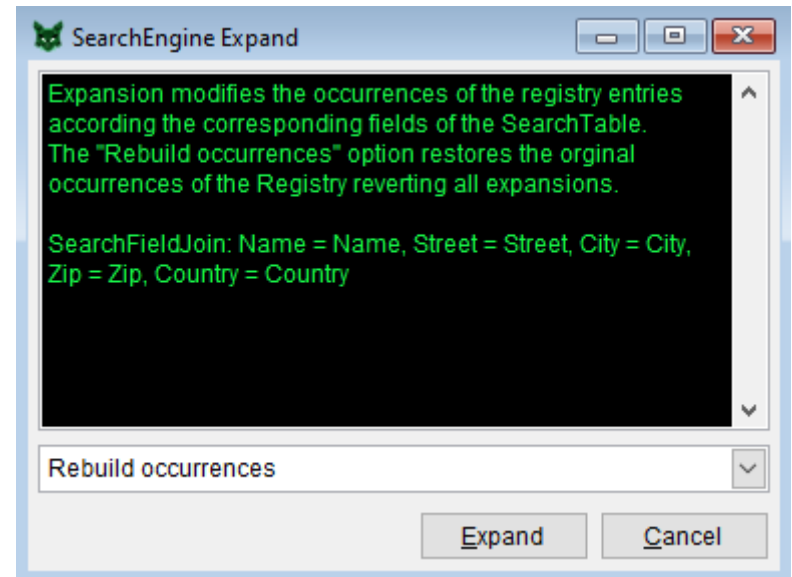- Shows the current search types.
- For details see Create function.
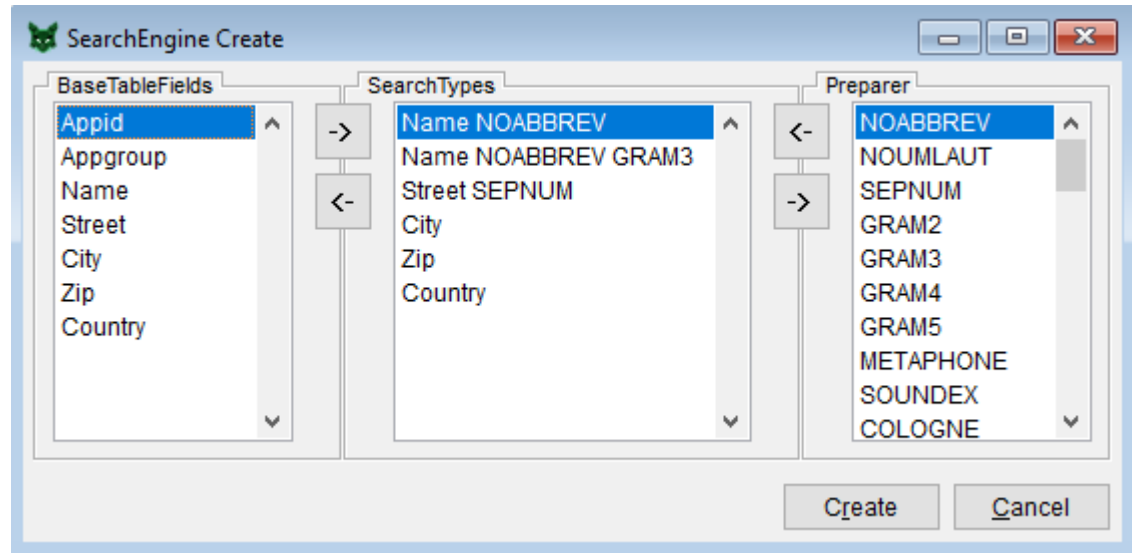
SearchEngine ▶ Action ▶ Expand

Expand
- This function creates a registry of the search table and merges it with base registry changing its occurrences.
- The following interactions between the occurrences of the base registry and the occurrences of the corresponding entries of the search table are available: replace, minimize, maximize, increment and average.
- The base registry can be restored to its original form by the rebuild option, which does not require a specified search table.
- No new registry entries will be created.
- All expansions are permanent and affect further searches with different search tables until rebuilt.
- Can be used to harmonize the registry in regard of systematic differences like legal forms or usage of appreviations, i.e. "University" vs "Univ", which otherwise would distort the heuristic (maximize, increment).
- If the search table has a different focus (i.e. wine producing firms) than the base table (i.e. patent assignees) the search may loose its focus (all options but minimize) because the specific words are getting a low identification potential ("wine" becomes a filler word) → candidates outside the focus will be preferred.
- Given the experimental character of expansions, it is advised to examine candidates retrieved after an expansion using the run option of the export function.

SearchEngine Expand

Expansion modifies the occurrences of the registry entries according the corresponding fields of the SearchTable. The "Rebuild occurrences" option restores the orginal occurrences of the Registry reverting all expansions.

SearchFieldJoin: Name = Name, Street = Street, City = City, Zip = Zip, Country = Country
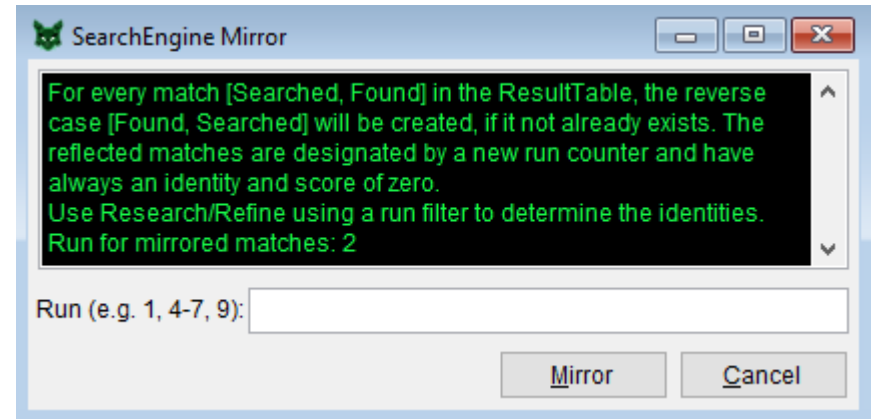
Rebuild occurrences

Expand   Cancel

- Rebuild – undo all expansions.
- Replace – imposes the distribution of the search table.
- Minimize – accentuates the words of the search table with a low occurrence.
- Maximize – levels systematic differences between search and base table.
- Increment – levels systematic differences between search and base table.
- Average – effect depends on the size difference of the search table to the base table.

SearchEngine ▶ Action ▶ Mirror

Mirror
- This function is only available for self-referential searches (disambiguation). It mirrors matches [Searched, Found] that have no reverse case [Found, Searched] because its identity was below the threshold during the search.
- The generated matches have always an identity and score of zero and get a new run counter, which is reported in the dialog.
- The mirror process can be restricted to specific runs by using the same syntax as the page selection before printing, e.g. 1-3,5,7.
- The identities of the generated matches can be defined by using the Research and/or Refine function on the specified run.
- Although not a requirement for cascaded traversal (see GroupedExport), mirroring allows to assess the similarity below the limit without needlessly inflating the result table with hopeless cases.
- By mirroring, very large result tables (close to 2GB or 79 Million records) can exceed system limits.

SearchEngine Mirror

For every match [Searched, Found] in the ResultTable, the reverse case [Found, Searched] will be created, if it not already exists. The reflected matches are designated by a new run counter and have always an identity and score of zero.
Use Research/Refine using a run filter to determine the identities.
Run for mirrored matches: 2

Run (e.g. 1, 4-7, 9):

Mirror   Cancel
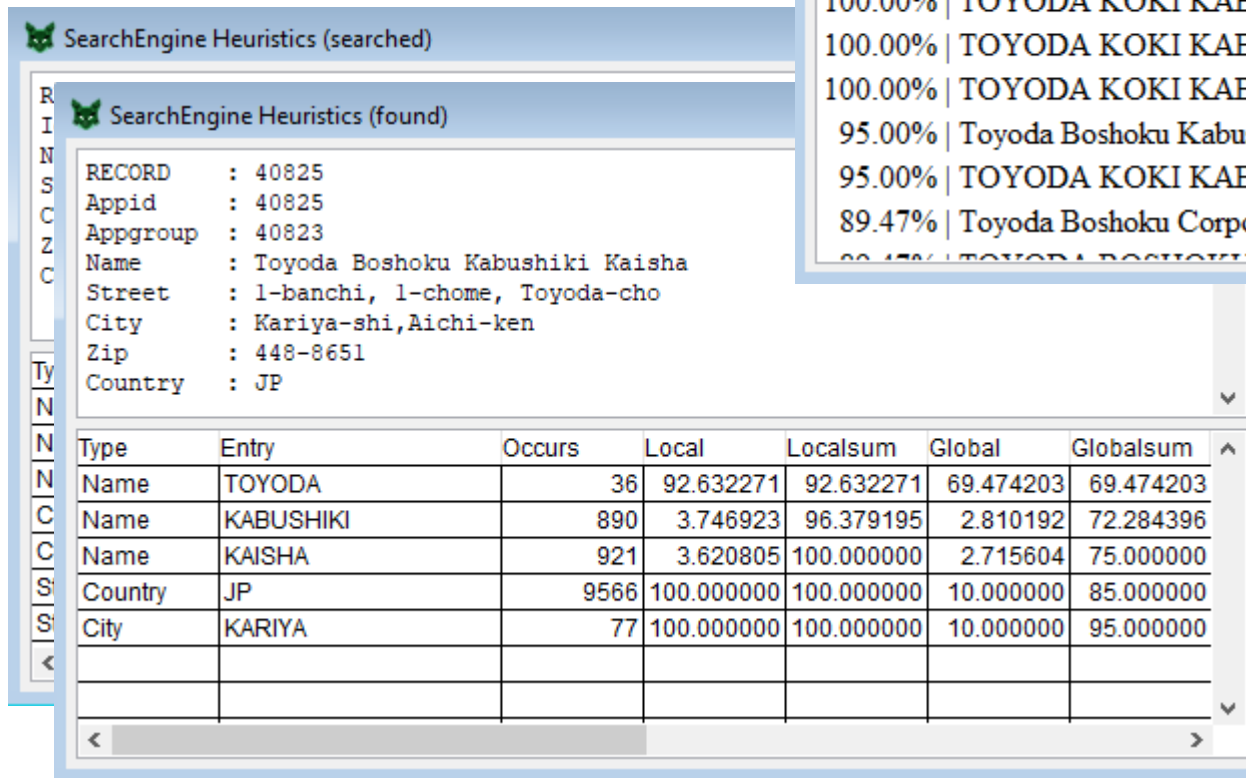
- Self-referential searches have the tendency to inflate the result table → try cutoff/activation (see Search) to reduce the impact of large clusters (1000 similar entries generate 1000*1000 matches) without loosing too much information as the truncated candidate lists are highly interconnected.

SearchEngine ▶ Tools ▶ Quick Search
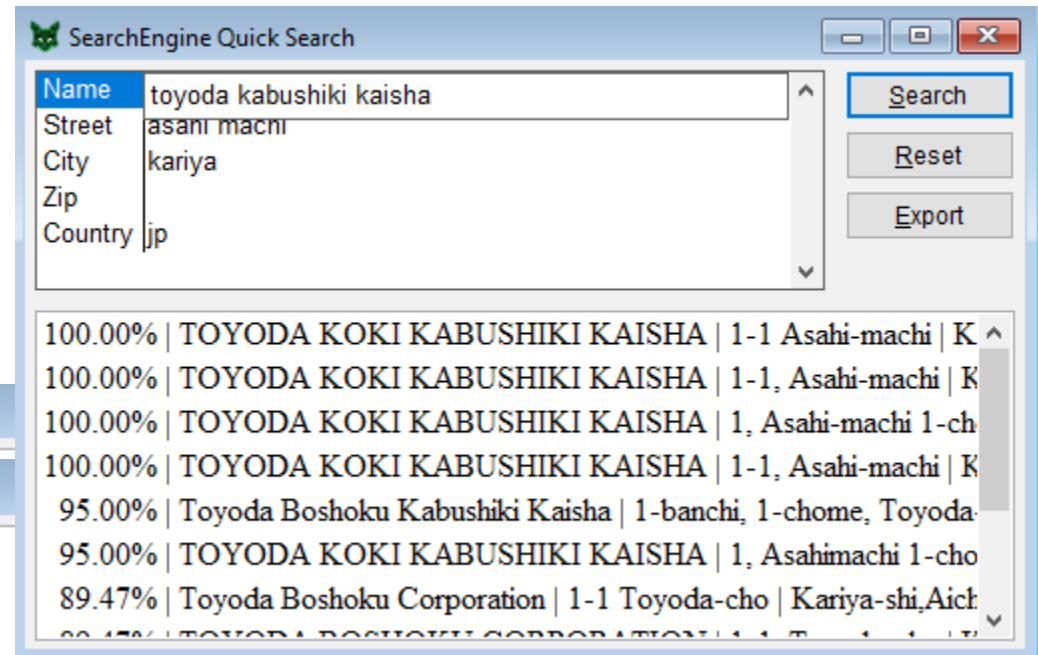
Quick Search
- Entered fields will be searched according to the current settings (no refine step).
- Candidate list can be exported (see Extended Export).
- Right-clicking on a candidate opens windows displaying the heuristics for the search term and the active candidate.

SearchEngine Quick Search

Name	toyoda kabushiki kaisha
Street	asani machi
City	kariya
Zip	
Country	jp

[Search] [Reset] [Export]

```
100.00% | TOYODA KOKI KABUSHIKI KAISHA | 1-1 Asahi-machi | K
100.00% | TOYODA KOKI KABUSHIKI KAISHA | 1-1, Asahi-machi | K
100.00% | TOYODA KOKI KABUSHIKI KAISHA | 1, Asahi-machi 1-ch
100.00% | TOYODA KOKI KABUSHIKI KAISHA | 1-1, Asahi-machi | K
 95.00% | Toyoda Boshoku Kabushiki Kaisha | 1-banchi, 1-chome, Toyoda-
 95.00% | TOYODA KOKI KABUSHIKI KAISHA | 1, Asahimachi 1-cho
 89.47% | Toyoda Boshoku Corporation | 1-1 Toyoda-cho | Kariya-shi,Aich
```

SearchEngine Heuristics (searched)

SearchEngine Heuristics (found)

```
RECORD    : 40825
Appid     : 40825
Appgroup  : 40823
Name      : Toyoda Boshoku Kabushiki Kaisha
Street    : 1-banchi, 1-chome, Toyoda-cho
City      : Kariya-shi,Aichi-ken
Zip       : 448-8651
Country   : JP
```

Type	Entry	Occurs	Local	Localsum	Global	Globalsum
Name	TOYODA	36	92.632271	92.632271	69.474203	69.474203
Name	KABUSHIKI	890	3.746923	96.379195	2.810192	72.284396
Name	KAISHA	921	3.620805	100.000000	2.715604	75.000000
Country	JP	9566	100.000000	100.000000	10.000000	85.000000
City	KARIYA	77	100.000000	100.000000	10.000000	95.000000

Heuristic
- Shows the fields of the selected record and the words of active search types along with the respective heuristic information like occurrence (incl. offset, log, softmax), local and global share of the rIP.

SearchEngine ▶ Tools ▶ Result Checker

Result Checker

- Displays the contents of the result table enriched with the contents of the joined search fields of the base (found) and the search (searched) table (separated by "|").
- This is the first station to assess the quality of a search .
- Right-clicking on the search term or an found entry opens the respective heuristic window (see Quick Search).
- The heuristic is based on the current settings and not on the parametrization of the search run
- The Result Checker can also be used for manual checking by marking valid candidates. This function is depreciated by Extended Export.

Searched

- Shows the record number of the search table and the associated search term.
- "<" or ">" moves to the next/previous search table record with candidates.
- "<<" or ">>" moves to the next/previous record with uncertain candidates.
- "<<<" or ">>>" moves to the next/previous record with unchecked candidates.
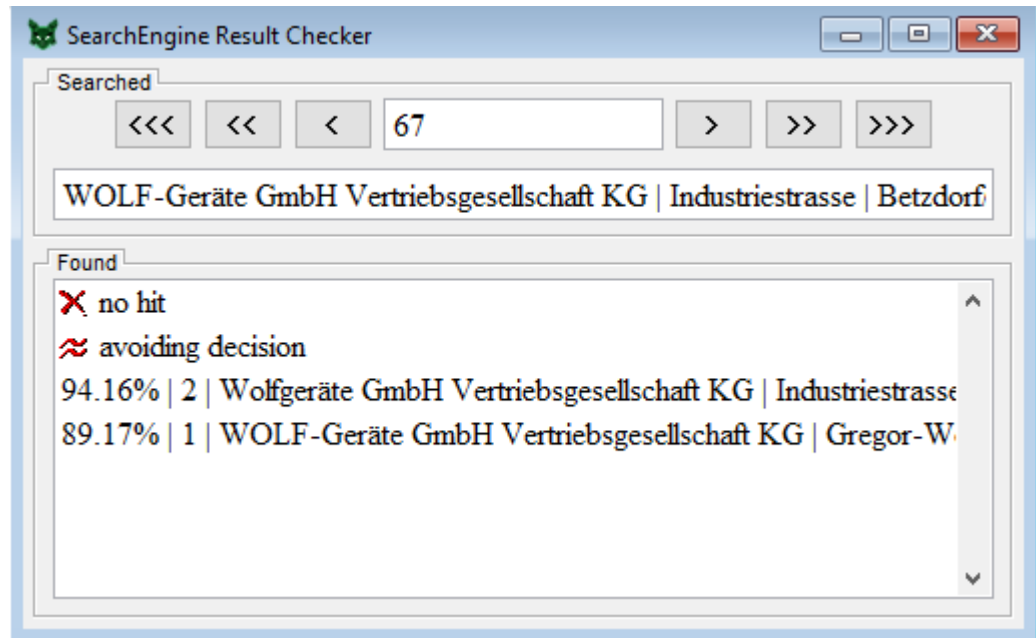
Found

- Shows the candidate identity, the run number and content of the search fields in the base table.
- Use simple clicking, SHIFT-clicking or CTRL-clicking to switch candidates between selected (match) and unselected (no match).
- CTRL-clicking on "avoiding decision" marks the assessment as uncertain.
- Clicking on "no hit" removes all selections and declares all candidates as non-matching.
- The checking functionality is depreciated because it is less efficient than working with Extended Export files.

SearchEngine ▶ Tools ▶ Notes
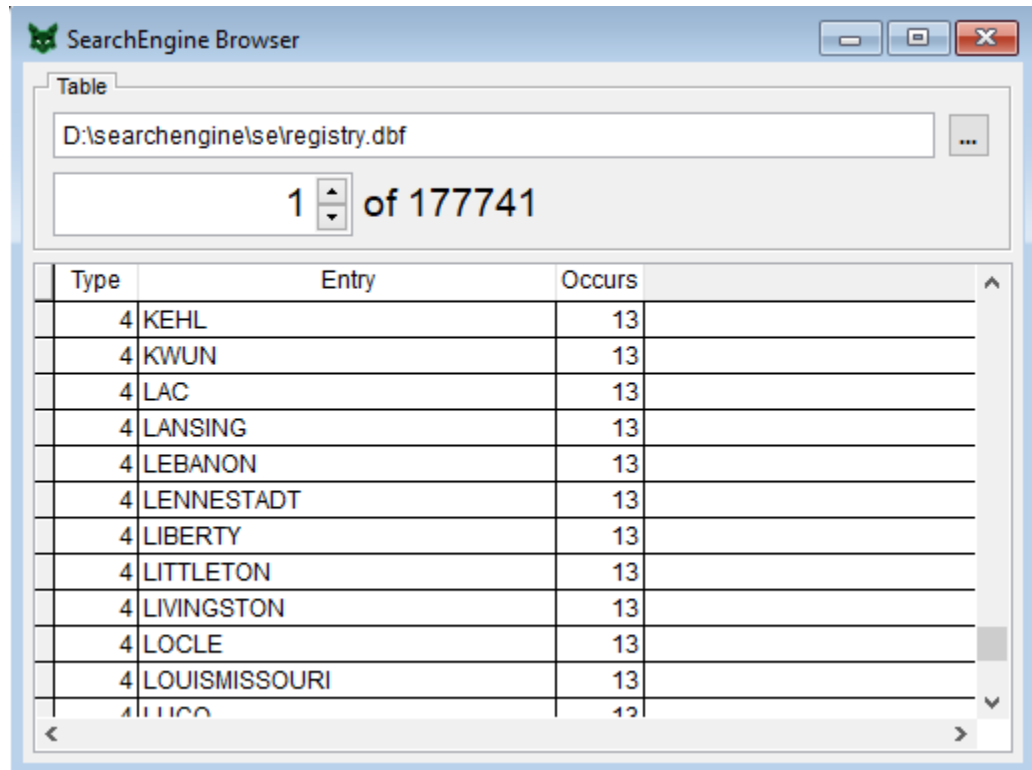
Notes
- Free text can be appended to the SearchEngine structure string following the tag "[info]".
- Every save slot of the SearchEngine has its own notes section.
- Can be used to make notes about the data, search strategies and other stuff related (and unrelated) to the current SearchEngine.
- The file "searchengine.log" in the SearchEngine directory reports all activities of the SearchEngine making the main reason for notes obsolete.

SearchEngine Notes

The SearchEngine needs to be recreated after removing html tags from the data.

Ok   Cancel
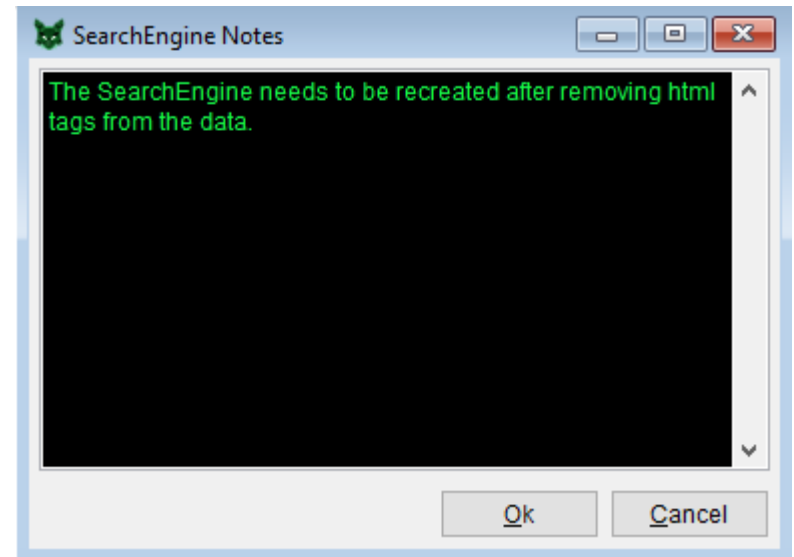
SearchEngine ▶ Tools ▶ Browser

Browser
- Opens a browser window to any selected Foxpro table (*.dbf).
- The browser window can be navigated by entering a record number, skipping records or by scrolling.
- Can be used to take a peek into the internal SearchEngine tables like registry.dbf or control.dbf.
- Allows to scrutinize imported search and base tables, originating from tab-delimited text files, for invalid characters or messed up field separation.

SearchEngine Browser

Table

D:\searchengine\se\registry.dbf

1 of 177741

Type	Entry	Occurs	
4	KEHL	13	
4	KWUN	13	
4	LAC	13	
4	LANSING	13	
4	LEBANON	13	
4	LENNESTADT	13	
4	LIBERTY	13	
4	LITTLETON	13	
4	LIVINGSTON	13	
4	LOCLE	13	
4	LOUISMISSOURI	13	
4	LUCO	13	

Batch mode and searchengine.log

- All actions performed using the GUI of the SearchEngine are logged into the file searchengine.log in the SearchEngine directory.

- The entries are identical to the commands of the SearchEngine batch mode to facilitate easy setup of batch scripts.

- Command line parameter of SearchEngine.exe:
  > searchengine.exe *batch_file* [*output_file* [append]] [[*para1=content*] [*para2=content*] …]
    - Output generated from the script is written into the optional output file.
    - If option "append" is specified, the existing output file will be complemented.
    - Parameters followed by an equal sign are placeholders. In the script file, placeholder are enclosed by rectangular brackets, e.g. importBase("[basefile]") and will be replaced by the contents of a parameter of the same name

- Syntax of batch scripts:
  [* *comment*]
  [silent|loud]
  [catch] [force] [silent] [loud] *command*
  [exit]

- Command parameters have different types which are represented by the first uppercase letters of the placeholder:
  I = Integer
  N = Numeric
  L = Logic: .t. equals true, .f. equals false
  S = String in quotes: "…"

- Command parameters can be omitted if designated as optional (rectangular brackets) and will be replaced with default values. Commands reported in the searchengine.log always show the abridged version omitting default settings.

SearchEngine keywords

silent

activates silent mode suppressing the output of the executed commands. Can be put in front of a command or separately as global setting. Other output is not affected.

loud

writes the executed commands to output file. Can be put in front of a command or separately as global setting to cancel silent mode.

force

the following command has the permission to overwrite files or show other reckless behavior for the sake of undisturbed execution.

catch

the script does not stop if the following command generated an error. Can be used instead of force to assure the existence of a file or condition without recreating it.

exit

terminates the execution of the script.

activation(*Iactivation***)**

sets activation limit for feedback to get triggered. Enforces ordering of candidates by least relevant noise. Has no impact on actual results unless **cutoff** is zero. [Settings]

benchmark()

runs the benchmark based on current MP settings and reports the results. [Preferences]

cancel()

stops the current script.

configtmp([*Spath*]**)**

defines the default path for temporary files. If *Spath* is omitted, the default directory will be used. [Preferences]

create(*Ssearchtypes***)**

creates the SearchEngine. The search types are separated by commas. A search type consists of the name of the search field followed the preparer names, separated by blanks. Can be forced to overwrite existing SearchEngine. [Create]

cutoff(*Icutoff***)**

sets the cutoff limit. The SearchEngine will try to keep the number of candidates per search term below this value. Works in conjunction with **activation**. [Settings]

darwinistic(*Ldarwin***)**

defines whether the SearchEngine keeps only the best candidates (.t) or transfers all candidates with an identity equal or greater than the limit into the results (.f. = default). [Settings]

depth(*Idepth***)**

sets the search depth, which can be a number between 0 and 8388608 (0 = default = 262144). [Preferences]

erase()

    erases the internal SearchEngine files. Base, search and result table are not affected.

expand(*expandMode***)**

    expands the SearchEngine by merging a virtual registry of the search table with the registry adjusting the occurrences. No new entries will be created. The parameter *expandMode* defines how the occurrences will be merged: 0 = restore original occurrences (default), 1 = replace with search table occurrence, 2 = use the maximum, 3 = use the minimum, 4 = increment by search table occurrence, 5 = use the average of both occurrences. [Expand]

exportExtended(*Stable* [, *Ssearchkey, Sfoundkey* [, *Ssearchgroupkey, Sfoundgroupkey*]]
    [, *Nlow, Nhigh* [, *Lexclusive*]] [, *Srunfilter*]**)**

    exports the result table using the extended format. If *Stable* has ".txt" as extension, the file format will be tab-delimited. *Ssearchkey* and *Sfoundkey* have to be specified if groupkeys are used. If they are empty, record numbers replace the keys. If a groupkey is specified in rectangular brackets ([key]), the key will be used for grouping but will not be reported (useful in conjunction with exportMeta). Can be forced to overwrite existing *Stable*. [Extended Export]

exportGrouped(*Stable* [, *Scascade*] [, Sbasekey] [, *Nlow, Nhigh* [, *Lexclusive*]] [, *Srunfilter*]
    [, *Lnotext* [, L*nosingles*]]**)**

    exports the result table using the grouped format. If *Stable* has ".txt" as extension, the file format will be tab-delimited. Can be forced to overwrite existing *Stable*. [Grouped Export]

exportResult(*Stable* [, *Nshuffle*] [, *Nlow*, *Nhigh*] [, *Srunfilter*] [, Lnewrun]**)**

    extracts a filtered copy/sample of the result table. If *Nshuffle* is not omitted or zero, a sample is drawn. The size is either a defined as share (*Nshuffle* < 1) or absolute number (*Nshuffle* ≥ 1) . If *Lnewrun* is .t., the run field in the new table will be without gaps. Tab-delimited export is not supported. Can be forced to overwrite existing *Stable*. [Result Export]

exportMeta(*Stable* [, *Smeta*] [, *Lraw*] [, *Nlow, Nhigh*] [, *Srunfilter*]**)**

    exports meta data of the result table. Can be forced to overwrite existing *Stable*. [Meta Export]

feedback(*Nfeedback***)**

    sets the feedback, which can be a number between 0 and 100. [Settings]

SearchEngine commands

ignorant(*Lignorant***)**
    defines whether the SearchEngine is ignoring words not represented in the Registry (.t.) or is giving them the average identification potential of the corresponding search type (.f. = default). [Settings]

getpara([*Spara*]**)**
    displays the content of placeholder *Spara* or of all placeholders, if *Spara* is omitted or empty (see **getpara**).

importBase(*Sfile* [, *Ldecode* [, *Lnomemos* [, *Lfast*]]]**)**
    imports respectively declares the base table. If the file extension is ".txt", the file is imported into Foxpro format. If it is already imported, the existing file will be used. Parameters can be omitted from right to left. [File Locations]

importSearch(*Sfile* [, *Ldecode* [, *Lnomemos* [, *Lfast*]]]**)**
    imports respectively declares the search table. If the file extension is ".txt", the file is imported into Foxpro format. If it is already imported, the existing file will be used. Parameters can be omitted from right to left. [File Locations]

info(*Sinfo***)**
    sets the notes in the info section of the SearchEngine structure string. The tag "<br>" will be translated to a line break. [Notes]

join(*Sfield* [, *Ssearchfield*]**)**
    links a field of the search table (*Sfield*) to a search field (*Ssearchfield*). If both have the same name, the search field can be omitted. [Join Search Fields]

limit(*Nlimit***)**
    set the threshold for the identity of the candidates. Can be a number between 0 and 100. Identical to **threshold**. [Settings]

load([*Sslot*]**)**
    loads the specified SearchEngine slot. If omitted or empty, the current slot will be reloaded. [Load]

SearchEngine commands

message(*Stext***)**

    opens a message box showing *Stext*. Program halts until confirmation.

mirror([Srunfilter]**)**

    mirrors the matches without a reverse entry to enforce symmetry. Can be restricted by a run filter. Can be forced to always increment run counter, even if no new matches were created. [Mirror]

mp([Icpu]**)**

    dedicates *Icpu* number of CPUs to SearchEngine actions. Negative numbers determine the CPUs not used by the SearchEngine. Omitting *Icpu* returns the current setting and a zero reserves up to 6 CPUs (initial setting). [Preferences]

note(Snote**)**

    appends a new line to the notes in the info section of the SearchEngine structure string. The tag "<br>" will be translated to a line break. [Notes]

output(Slogfile [, Lappend]**)**

    defines a new log file for the SearchEngine output. If *Lappend* is .t., an existing file will be complemented instead of overwritten. This command can be used instead of the SearchEngine parameter.

refine([*Iidentitymode***,** *Icomparemode*] [**,** *Srunfilter*] [**,** *Ldestructiveonly*]**)**

    refines the existing matches in the result table. *Iidentitymode* defines how the refined identity relates to the existing identity of the respective match: 1 = replace, 2 = maximize, 3 = minimize, 4 = additive, 5 = average. *Icomparemode* defines the direction on the LRCPD comparison: 1 = searched in found, 2 = dynamic, 3 = found in searched. If *Ldestructiveonly* is .t., only search types containing destructive preparer are included renouncing the priorities of the other search types. This option should be used in conjunction with **research** and additive identity mode. [Refine]

relative(*Lrelative***)**
>   defines whether the SearchEngine redistributes the priorities of missing search fields (.t.) or leaves them missing reducing the maximum identity by the corresponding priorities (.f. = default). [Settings]

remove(*Sslot***)**
>   removes the specified save slot. [Save]

research([*Iidentitymode***,** *Iscoremode*] [**,** *Srunfilter*] [**,** *Lnondestructiveonly*]**)**
>   researches the existing matches in the result table. *Iidentitymode* defines how the researched identity relates to the existing identity of the respective match: 1 = replace, 2 = maximize, 3 = minimize, 4 = additive, 5 = average. *Iscoremode* defines how the researched score relates to the existing score of the respective match: 1 = replace, 2 = maximize, 3 = minimize. If *LnondestructiveOnly* is .t., only search types not containing destructive preparer are included renouncing the priorities of the other search types. This option should be used in conjunction with **refine** and additive identity mode. [Research]

result(*Sresult***)**
>   sets the result table. Can be exchanged independently. Will be overwritten without warning, if a search is initiated. Every result table has its own run counter. [File Locations]

run()
>   writes the current run count of the active result table to the output file.

safemode([*Lsafemode*]**)**
>   if Lsafemode is .t., the SearchEngine will use visible Foxpro workers instead of invisible Parallelfox workers. Safe mode will be automatically activated, if the SearchEngine cannot register the Parallelfox workers, which requires administrator rights. [Preferences]

save(*Sslot***)**
>   saves the current SearchEngine structure in the specified save slot. [Save]

SearchEngine commands

say(*Ssomething***)**
 writes a comment line to the output file.

scope(*Iscope***)**
 defines the width of the LRCPD scope (12 = default). [Preferences]

screen(*Sproperty* [, *value*]**)**
 sets various screen properties determining the look of the execution window:
 screen("width", 550) sets the screen with to 550 pixel.
 screen("height",400) sets the screen height to 400 pixel.
 screen("left",100) sets the screen coordinate of the top left corner to 100 pixel.
 screen("top",100) sets the screen coordinate of the top left corner to 100 pixel.
 screen("hide") hides the screen.
 screen("maximize") maximizes the screen.
 screen("minimize") minimizes the screen.
 screen("normal") switches the screen from maximized/minimized state into normal state .
 screen("backcolor", "0,0,0") sets the back color of the screen in RGB format.
 screen("forecolor", "25,245,75") sets the fore color of the screen in RGB format.
 screen("font", "Courier New") sets the font.
 screen("fontsize", 9) sets the font size in points.

setpara(*Spara* [, *value* [, *Lkeep*]]**)**
 sets the value of the placeholder *Spara* to the value *value* (of any type). If *Lkeep* is .t. and the placeholder is already defined as command line parameter or by an earlier **setpara** call, the original value will not be changed.

SearchEngine commands

search([*Iincrement*] [, *Icomparemode* [, *Lrefineforce*] [, *Nrefinelimit*]]**)**
> executes a search. *Iincrement* determines the interaction with the existing results: 0 = replace existing results, 1 = complete for unmatched search records, 2 = merge results, 3 = continue canceled search, -1 = like 1, but removes last run, -2 = like 2, but removes last run. If destructive preparer are involved, candidates will be automatically refined. By default, only search fields with destructive preparer will be refined. All other fields will be researched. *Icomparemode* defines the direction on the LRCPD comparison: 1 = searched in found, 2 = dynamic, 3 = found in searched. If *Lrefineforce* = .t., all search fields will be refined regardless of the involvement of destructive preparer. If the results will be refined, a separate threshold can be defined with *Nrefinelimit* [0,100].[Search]

show()
> writes the SearchEngine structure string to the output file.

slot()
> writes the current save slot name to the output file.

threshold(*Nthreshold***)**
> sets the threshold for the identity of the candidates. Can be a number between 0 and 100. Identical to **limit**. [Settings]

time()
> writes the current date and time to the output file.

timer([*Ltimer*]**)**
> if *Ltimer* is .t. or omitted, the run time of every action will be reported in the log file as a comment. This behavior can be deactivated with *Ltimer* = .f.. [Preferences]

SearchEngine commands

types(_Ssearchtypes_**)**

determines the search types settings. Search types are separated by commas. A search type definition consists of the search type name, a priority, an optional offset (preceded with a plus or minus sign) , an optional softmax parameter (preceded with a hash #) and an optional "log" keyword to enable logarithmic smoothing. It is advised to always specify all search types in order of definition. Unused types have a priority of zero. A softmax value below 1 smooths while a value above 3 accentuates the search term distribution (max. accentuation = 30). Example: "firm 35 #10, firm 35 log, street 10 -5000, city 10, state 10". [Search Types]

unjoin([_Sfield_]**)**

removes the link between a field of the search table (_Sfield_) and a search field. If _Sfield_ is omitted, all links will be cleared. It is best practice to first unjoin all links before joining them. [Join Search Fields]

version()

writes the current version to the output file.

wait(_Iseconds_**)**

halts execution for _Iseconds_ seconds or until button press. if _Iseconds_ is zero, execution will be continued after button press.

zealous(_Lzealous_**)**

defines whether the SearchEngine is dynamically lowering the threshold to guarantee matches (.t.) or is complying to the threshold disposing all candidates below it (.f. = default). [Settings]

SearchEngine technical notes

- **Development Environment**
  - Microsoft Visual Foxpro 9.0 SP2 (32-Bit)
  - Microsoft Visual C++ 6.0 (32-Bit)
- **Minimum Requirements**
  - Windows, any Version from XP upwards
  - 2 GB RAM
- **Limitations**
  - Result table has a maximum size of 2GB
    - 79,536,413 records
  - Foxpro export tables may not exceed 2GB
  - Text based export tables have no size limitation
  - Base/Search tables have no size limitation
    - internal representation as table clusters of 2GB tables

License agreement

Send bug reports to: doherr@zew.de