

ACCESSDATA SUPPLEMENTAL APPENDIX

dtSearch Search Requests

Note: This following dtSearch information was developed by DT Software, Inc. Copyright 1991–1997 DT Software, Inc. www.dtsearch.com.

This appendix provides the search information for the dtSearch engine included in AccessData[®] Forensic Toolkit[®] (FTK[®]):

- *Overview* on page 1
- *Words and Phrases* on page 2
- *Wildcards* on page 3
- *Natural Language Searching* on page 3
- *Synonym Searching* on page 4
- *Fuzzy Searching* on page 4
- *Phonic Searching* on page 4
- *Stemming* on page 5
- *Variable Term Weighting* on page 5
- *Field Searching* on page 5
- *AND Connector* on page 6
- *OR Connector* on page 6
- *w/n Connector* on page 6
- *NOT and NOT w/n* on page 7
- *Numeric Range Searching* on page 7

OVERVIEW

dtSearch supports two types of search requests. A natural language search is any sequence of text, such as a sentence or a question. After a natural language search, dtSearch sorts retrieved documents based on their relevance to your search request.

A boolean search request consists of a group of words or phrases linked by connectors such as “and” and “or”. The following table includes some boolean search examples.

Example	Result
apple and pear	Both words must be present.
apple or pear	Either word can be present.
apple w/5 pear	“Apple” must occur within five words of “pear.”
apple not w/5 pear	“Apple” must not occur within five words of “pear.”
apple and not pear	Only “apple” must be present, not “pear.”
name contains smith	The field “name” must contain “smith”.

If you have more than one connector, use parentheses. For example, “apple and pear or orange” could mean “(apple and pear) or orange” or “apple and (pear or orange)”.

“Noise words” like “if” and “the” are ignored in searches. The following table describes the special characters that can be included in search terms.

Character	Description	Result
?	Matches any single character	"appl?" matches "apply" or "apple"
*	Matches any number of characters	"appl*" matches "application"
~	Stemming	“apply~” matches “apply”, “applies”, “applied”
%	Fuzzy search	“ba%nana” matches “banana” and “bannana”
#	Phonic search	"#smith" matches "smith" and "smythe"
&	Synonym search	"fast&" matches "quick"
~~	Numeric range	"12~~24" matches "18"
:	Variable term weighting	"apple:4 w/5 pear:1"

WORDS AND PHRASES

You do not need to use any special punctuation or commands to search for a phrase. Simply enter the phrase the way it ordinarily appears. You can use a phrase anywhere in a search request. For example, “apple w/5 fruit salad”.

If a phrase contains a noise word, dtSearch skips it. For example, a search for “statue of liberty” would retrieve any document containing “statue”, any intervening word, and “liberty”.

Punctuation inside of a search word is treated as a space; thus, “can't” would be treated as a phrase consisting of two words, “can” and “t”. In this example, “1843(c)(8)(ii)” would become “1843”, “c”, “8”, and “ii” (four words).

WILDCARDS

A search word can contain the asterisk (*) and question mark (?) wildcard characters. A ? in a word matches any single character, and a * matches any number of characters. The wildcard characters can be in any position in a word. The following table includes wildcard search examples.

Example	Result
appl*	Matches “apple”, “application”, etc.
cipl	Matches “principle”, “participle”, etc.
appl?	Matches “apply”, and “apple”, but not “apples”.
ap*ed	Matches “applied”, “approved”, etc.

Note: Use of the * near the beginning of a word slows searches somewhat.

NATURAL LANGUAGE SEARCHING

A natural language search request is any combination of words, phrases, or sentences. After a natural language search, dtSearch sorts retrieved documents by their relevance to your search request. Weighting of retrieved documents takes the following into account: the number of documents each word in your search request appears in (the more documents a word appears in, the less useful it is in distinguishing relevant from irrelevant documents), the number of times each word in the request appears in the documents, and the density of hits in each document. Noise words and search connectors like “not” and “or” are ignored.

SYNONYM SEARCHING

Synonym searching finds synonyms of a word in a search request. For example, a search for “fast” would also find “quick”.

You can enable synonym searching for all words in a request, or you can enable synonym searching selectively by adding an ampersand (&) after certain words in a request. For instance, “fast& w/5 search”.

FUZZY SEARCHING

Fuzzy searching can find a word even if it is misspelled. For example, a fuzzy search for “apple” can find “appple”. Fuzzy searching is useful when you are searching text that might contain typographical errors. You can add fuzziness to searches in two ways:

- Enable fuzziness for all of the words in your search request. You can adjust the level of fuzziness from 1 to 10.
- You can add fuzziness selectively by using the percent sign (%). The number of % characters you add determines the number of differences dtSearch ignores when searching for a word. The position of the % determines how many letters at the start of the word have to match exactly. The following table includes some examples.

Example	Result
ba%nana	The word must begin with “ba” and have, at most, one difference between it and “banana”.
b%%anana	The word must begin with “b” and have, at most, two differences between it and “banana”.

PHONIC SEARCHING

Phonic searching looks for a word that sounds like the word you are searching for and begins with the same letter. For example, a phonic search for “Smith” can also find “Smithe” and “Smythe”.

To search for a word phonically, place a pound sign (#) before the word in your search request—for example, “#smith”.

You can also select the **Phonic** option in the search form to enable phonic searching for all words in your search request. Phonic searching is somewhat slower than other types of searching and tends to make searches over-inclusive; therefore, consider performing selective phonic searches.

STEMMING

Stemming extends a search to cover grammatical variations on a word. For example, a search for “fish” would also find “fishing”. A search for “applied” would also find “applying”, “applies”, and “apply”. You can add stemming to your searches in two ways:

- Select the **Stemming** option in the search form to enable stemming for all of the words in your search request. Stemming does not slow searches noticeably and usually helps you find what you want.
- To use stemming selectively, add a tilde (~) after words that you want stemmed—for example, “apply~”.

VARIABLE TERM WEIGHTING

When dtSearch sorts search results, all the words in a search count equally in determining hits. However, you can change this by specifying the relative weight for each term in your search request. For example, “apple:5 and pear:1” would retrieve the same documents as “apple and pear”, but dtSearch would weight “apple” five times heavier than “pear” when sorting the results.

In a natural language search, dtSearch automatically weights terms based on an analysis of their distribution in your documents. However, you can override these weights by providing specific term weights.

FIELD SEARCHING

When you index a database or other file containing fields, dtSearch saves the field information so that you can perform searches limited to a particular field. For example, suppose that you indexed an Access database with a Name field and a Description field. You could search for “apple” in the Name field like this: “Name contains apple”.

You can also define a field at the time of a search by designating words that begin and end the field, like this: “(beginning to end) contains (something)”.

“(Beginning to end)” defines the boundaries of the field. “Contains (something)” indicates the words or phrases you are searching for in the field. The only connector allowed in the beginning and end expressions in a field definition is “or”. Below are two examples:

- (name to address) contains (john smith)
- (name to (address or xlastword)) contains (oak w/10 lane)

The field boundaries are not considered hits in a search. Only the words being searched for (“john smith”, “oak”, and “lane”) are marked as hits.

AND CONNECTOR

Use “and” in a search request to connect two expressions, both of which must be found in any document retrieved. The following table includes two examples.

Example	Result
apple pie and poached pear	Retrieves all documents containing both phrases.
(apple or banana) and (pear w/5 grape)	Retrieves all documents containing (1) either “apple” or “banana”, and (2) containing “pear” within five words of “grape”.

OR CONNECTOR

Use “or” in a search request to connect two expressions, at least one of which must be found in any document retrieved. For example, “apple pie or poached pear” retrieves all documents containing “apple pie”, “poached pear”, or both.

W/N CONNECTOR

Use “w/*n*” in a search request to specify that one word or phrase must occur within *n* words of the other. For example:

- apple w/5 pear (retrieves all documents containing “apple” within five words of “pear”)
- (apple or pear) w/5 banana
- (apple w/5 banana) w/10 pear
- (apple and banana) w/10 pear

Some types of complex expressions using “w/*n*” produce ambiguous results and should not be used. For example:

- (apple and banana) w/10 (pear and grape)
- (apple w/10 banana) w/10 (pear and grape)

In general, at least one of the two expressions connected by “w/*n*” must be a single word or phrase or a group of words and phrases connected by “or”. For example:

- (apple and banana) w/10 (pear or grape)
- (apple and banana) w/10 orange tree

dtSearch uses two built-in search words to mark the beginning and end of a file: “xfirstword” and “xlastword”. The terms are useful if you want to limit a search to the beginning or end of a file. For example, “apple w/10 xlastword” retrieves all documents containing “apple” within 10 words of the end of a document.

NOT AND NOT W/N

Use “not” before any search expression to reverse its meaning. This allows you to exclude documents from a search—for example, “apple sauce and not pear”.

“Not” standing alone can start a search request. For example, “not pear” retrieves all documents that do not contain “pear”.

If “not” is not the first connector in a request, you need to use either “and” or “or” with it—for example, “apple or not pear” and “not (apple w/5 pear)”.

The “not w/n” (not within) operator allows you to search for a word or phrase not in association with another word or phrase, for example, “apple not w/20 pear”.

Important: Unlike “w/n”, “not w/n” is not symmetrical. That is, “apple not w/20 pear” is not the same as “pear not w/20 apple”. In “apple not w/20 pear”, dtSearch searches for “apple” and excludes cases where “apple” is too close to “pear”. In “pear not w/20 apple”, dtSearch searches for “pear” and excludes cases where “pear” is too close to “apple”.

NUMERIC RANGE SEARCHING

A numeric range search looks for numbers within a range. To perform a numeric range search, enter the upper and lower bounds of the range separated by two consecutive tildes (~), like this: “apple w/5 12~~17”.

This search would retrieve all documents containing “apple” within five words of a number between 12 and 17. Numeric range searches include the upper and lower bounds, so in this example, “12” and “17” would be retrieved.

Numeric range searches work only with positive integers. Decimal points and commas are treated as spaces, and minus signs are ignored. For example, “123,456.78”78” would be interpreted as “123”, “456”, and “78” (three numbers).

Using alphabet customizing, the interpretation of punctuation characters can be changed. For example, if you change the comma and period from **space** to **ignore**, then “123,456.78” would be interpreted as “12345678”.