ZyLAB General Search Language Guide

Contact. Help. Training. Documentation.

For full contact details, visit the ZyLAB website - http://www.zylab.com

Ask us any question about installing, configuring and using ZyLAB at https://help.zylab.com with a support request. We offer a variety of information, a knowledge base, resources and a user community.

Follow our online training program - https://zylab.litmos.com

For our latest, most up to date documentation, please refer to https://docs.zylab.com

Disclaimer

Contents

Introduction

In this ZyLAB General Search Language Guide, we will discuss how to use the ZyLAB search language to search for one or more *keywords* (page 109) within a data set. The data set usually contains files in many forms. Not only the keywords used in these text, image or audio files, but also the information about these files (the metadata), can and should be searched.

Search Techniques

Keyword searches are prone to produce over- and under-inclusive results. Several search techniques are designed to resolve this issue. For example, wild card searches to help you find common spelling variations and misspellings, or Boolean searches to specifically include or exclude certain keywords. View the ZyLAB search language techniques in the table on the next page.

Please note that though some *operators* (page 110) are expressed in capital letters, this is only done for your personal clarity. The search engine does not differentiate between capital and lowercase letters.
It is possible to specify language specific (i.e. English, German, French, etc.) versions of these operators.

There should always be a space between an operator and a keyword, otherwise both operator and keyword will be seen as one *term* (page 113). These are correct: "NOT term", "not term". These are not correct: "NOTterm", "notterm".
However, when using parentheses to surround the keyword, no spaces are necessary: "NOT(term)". For more information, see the definition of *Parentheses* (page 110).

In addition, check the definitions defined in *Appendix B* (page 108).

Boolean and Proximity Operators		Term Operators	
AND		Fuzzy	~n
OR		Wild Cards	?
NOT			*
TO			[character(s)]
IN fieldname{query}			[character-range]
Within	W/n		[^]
	W/n/term		+
	/n,m/		{m,n}
Precedes	P/n		{m}
	P/n/term		{m,}
Number Range	<		
	<=		
	=		
	<>		
	>	Field Filter	
	>=	fieldname=query	
Quorum	n of {term, term, ..}		
Exclude List of Terms from Fuzzy/Wild Card Query			
fuzzy/wild card query - {exclude_term_1, ..., exclude_term_n}			

- Use the Term operators on the right to enhance your *search queries* (page 113). For more information, see:
  - *Fuzzy Searches* (page 67)

- *Wild Card Searches* (page 69)

- Use the Boolean and Proximity operators on the left (AND, OR, NOT, etc.) to connect terms or phrases, making the search query more effective. Operators can be used to broaden or narrow your search. They can also be used to define your search query more precisely.
  For more information, see:
  - *AND Operator* (page 49)
  - *OR Operator* (page 52)
  - *NOT Operator* (page 54)
  - *TO Operator* (page 56)
  - *IN fieldname{query}* (page 58)
  - *Within Operator* (page 59)
  - *Precedes Operator* (page 62)
  - *Number Range Operator* (page 63)
  - *Quorum Operator* (page 66)
- Use the Field filter to search on field names. For more information, see:
  - *Field Filter* (page 87)
- Use the Exclude List to exclude specific terms from your fuzzy or wild card query. For more information, see:
  - *Exclude List of Terms from Search Query* (page 88)
- For more search techniques see:
  - *Quoted and Unquoted Searches* (page 91)
  - *Complex Search Queries* (page 92)
  - *Macros* (page 103)

Search Results Explained

Once a *search query* (page 113) is being executed, a result list will appear. Retrieved terms (*occurrences* (page 109)) will be highlighted in the files. Of course, to be found, terms need to be present in the file. However, whether a term is retrieved also depends on the settings in the *character map* (page 108), the indexing structure and the *tokenizer* (page 114).
Based on the character map the tokenizer will process all files. How this is done, we will explain here.

The building blocks of a text file are characters, (hyphenated) terms and phrases. **Characters** are letters, numbers or symbols like %, @, &, ^, *, etc. **Terms** are characters or words; they are unique entries in the dictionary with a separator on either side. **Phrases** are two or more terms with no intervening operator. **Hyphenated terms** (such as sugar-free) are two or more separate terms, connected with a hyphen. Each part of a hyphenated term has the same token id, given by the tokenizer:

Token	I	like	sugar-	free	food	EOS	EOD
Token id	1	2	**3**	**3**	4	x	x

A token id is the natural number or position of a token, given by the tokenizer. Token ids are used to determine the distance between the terms. Separators do not have token ids. If a term or combination of terms you are searching for contains a hyphen, that term will be found, even if you did not include a hyphen in your search query. For example, when you search for 'email' or 'e mail', it will also find 'e-mail'. However, 'e-mail' will only retrieve 'e-mail'. In addition, 'e mail' will not find 'email' or the other way around ('email' will not find 'e mail').

The tokenizer extracts text from a file and produces tokens, based on the settings defined in the character map. Tokens can be anything between two separators. Tokens are the identified small parts that form or define a file. Tokens are not terms! For example, hyphenated terms all have the same token id, but are separate terms. And a separator (for example, EOD) can be a token, but not a term.

The character map determines which characters are used to separate terms, which characters are indexed, which ones are used for punctuation, etc. All possible characters that can be recognized and searched on are listed in the character map. By default, some characters are not indexed and will not be found unless the default character map is adjusted. How characters are defined in the character map, influences the outcome of a search. For example, when brackets are set to be separators, the following text will be identified as 3 terms: 'most definite(ly)'.
For more information on the character map and how to configure it, please contact support (http://help.zylab.com).

In addition to the characters defined in the character map to be recognized by the tokenizer as separators, the tokenizer creates separators to mark the end of a sentence (EOS, disabled by default), end of a paragraph (EOP), end of a line (EOL), end of a page (EOG) or the end of a document (EOD). You can search for the operators EOP, EOL, EOG and EOD.

Tip: When searching for EOD, the query returns all files with nothing highlighted. Since each file has an EOD token, it is an easy query to find all files in a data set.

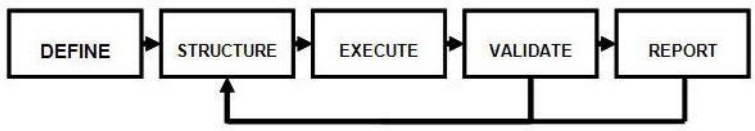
Define Search Queries

Legal Review

> Keyword formulations should be based on careful thought, quality control, testing and cooperation between all involved parties. Obtain a small sample from the result list to verify if a search query is valid or not. Adjust the search query to remove nonresponsive data and to ensure that search results are minimally over-inclusive or under-inclusive. Review the search hits, but also review at least some of the documents that were not selected. A quick review might reveal flaws in the search criteria and/or technique.

Conditions

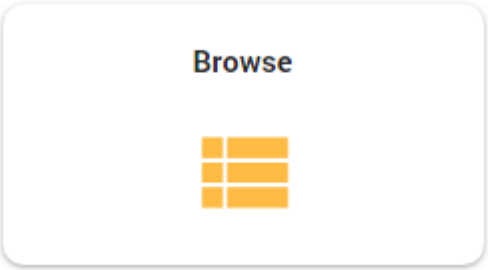
Before (and after) you start typing in your first search query, it is advised to follow the Search Framework as explained in the EDRM Search Guide (see Chapter 4), which can be found at http://www.edrm.net/resources/guides/edrm-search-guide/search-framework



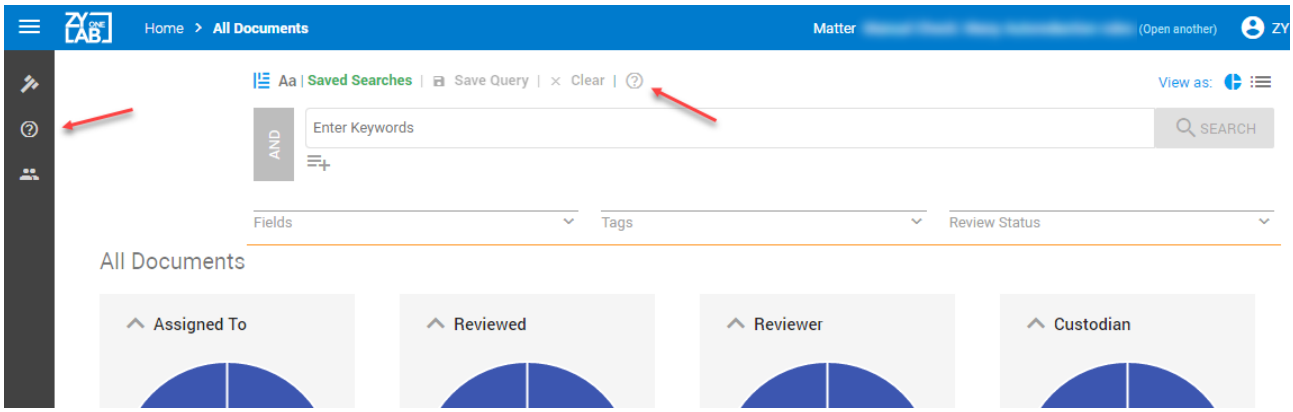
- Define the purpose of the search, and the goals to be achieved.
- Plan and structure the search; identify the scope of data to be searched, who will conduct the search, and the technology and process that will be used.
- Execute the search, monitor it and capture the process and results.
- Validate the efficacy of the search process to ensure its comprehensiveness and effectiveness.
- Report the search results (for example to opposing counsel).

Instructions

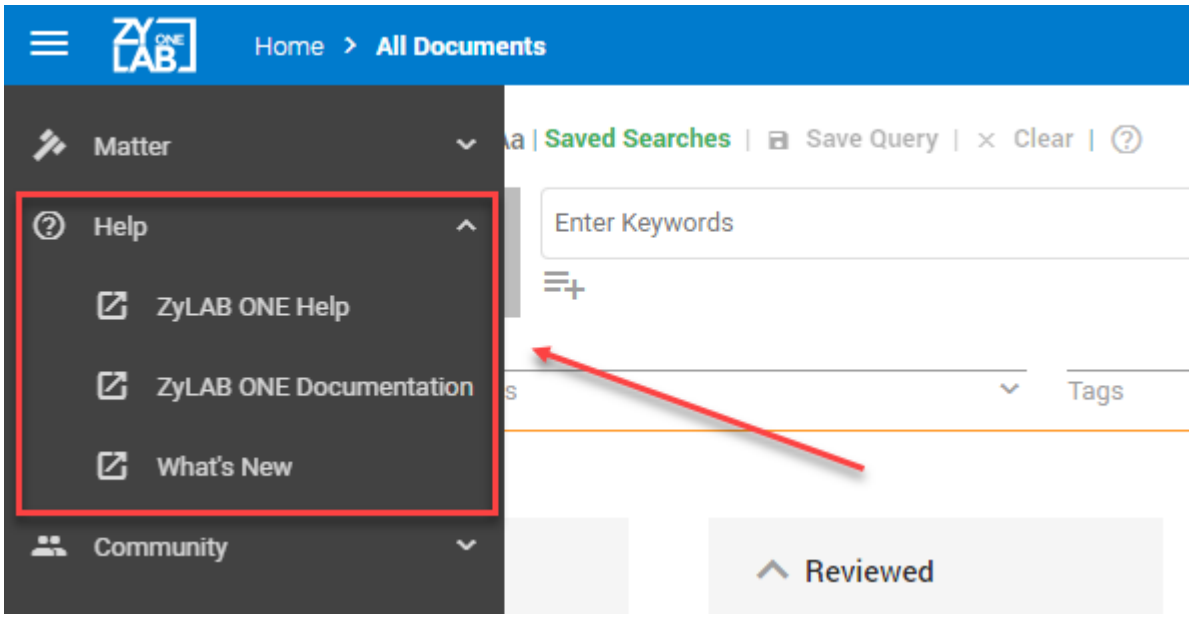
1. Click Browse (in ZyLAB ONE 5.0 Legal Review).



2. In the latest version of ZyLAB ONE, you can search with the latest technology. For the most recent information on searching, please select the help icon above the search field. You can also use the help icon in the menu on the left of your screen.



To contact Support, please select ZyLAB ONE Help.
For more information on Legal Review, Search and more, please select ZyLAB ONE Documentation.
For the latest information, please select What's New.

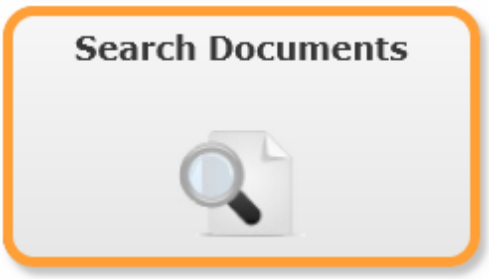


Search Audio and Video

Note: Silverlight only.

Instructions

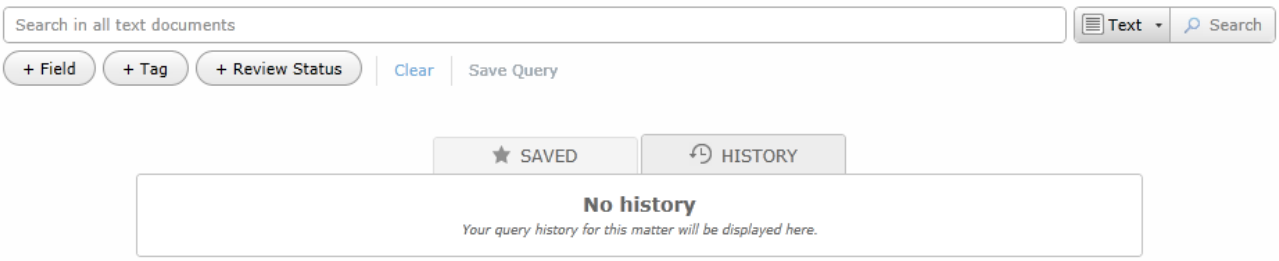
1. Click Search Documents.



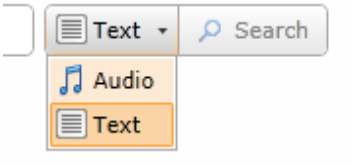
2. Insert keywords.

   Use the following audio search techniques:

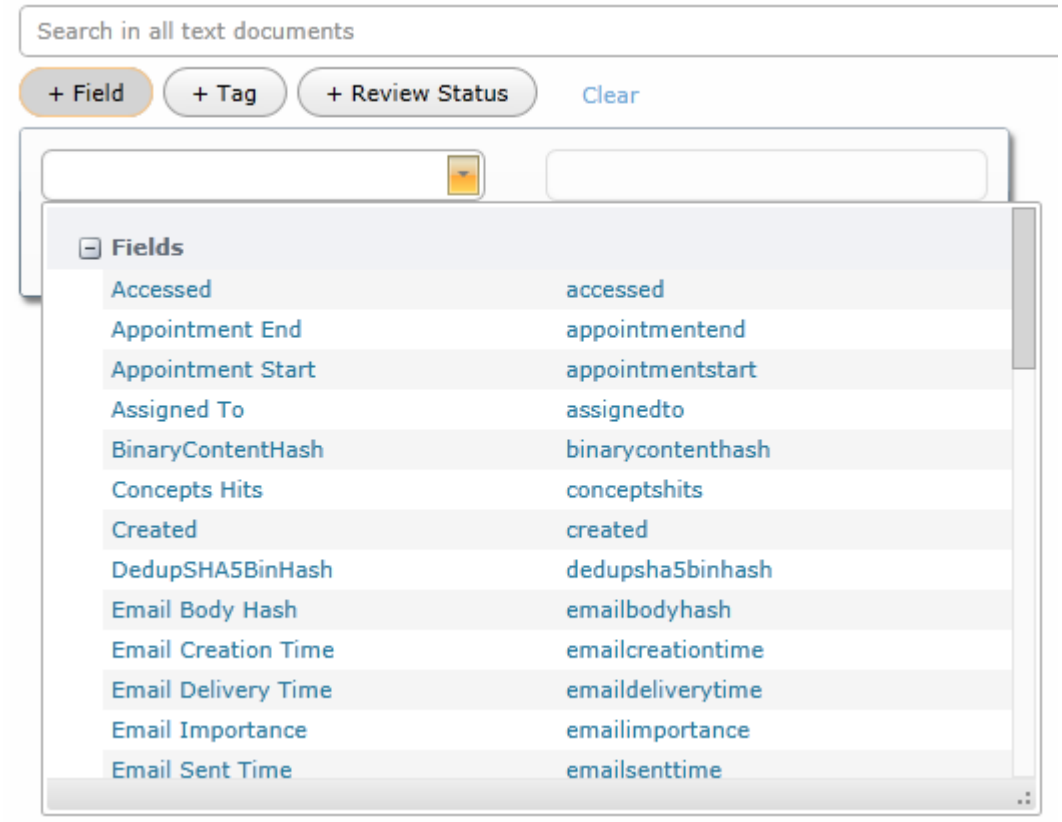
   - Phrase queries, e.g. 'I have a nice proposal for you'
   - AND
   - OR
   - n of {1, 2, …, x}, e.g. '2 of {fraud, crime, stealing, illegal}'
   - W/n (in seconds), e.g. 'fraud w/10 investigation'
   - Account for a minimum confidence level for your hits using the fuzzy operator, e.g. 'investigation~80'. In Audio Search, this means that you want to be for 80% sure this is the word you are looking for.
   - eod (lists all documents in the index)



3. Select Audio.

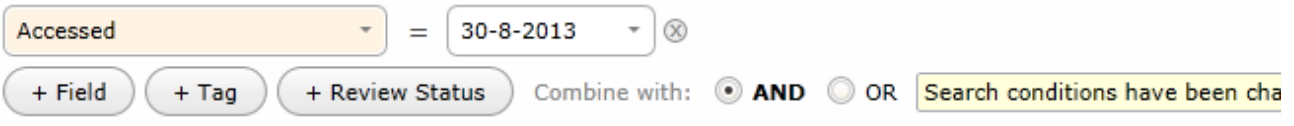


4. Click + Field to search your metadata.



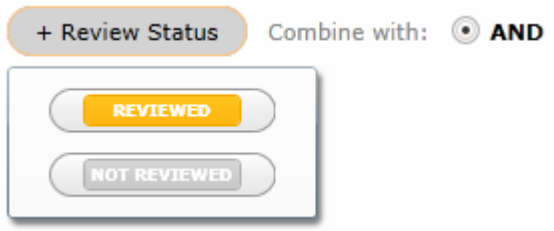
- Select a field from the list.
- Set a value.
- When you are finished, click Done (Enter) or select the Enter key.
  Click Cancel (Escape) if you do not want to use this field.

- Combine Field, Tag and Review Status searches with AND.

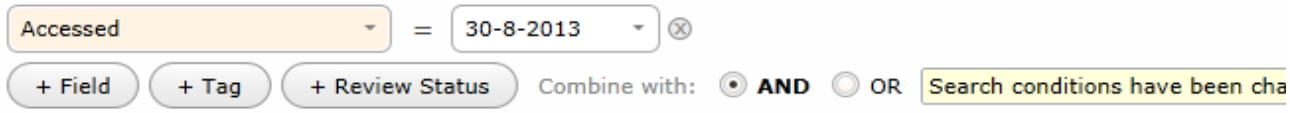


5. Click + Tag to search for documents with or without tags.

6. Click + Review Status to search for documents that are or are not reviewed.



7. Combine Field, Tag and Review Status searches with AND.

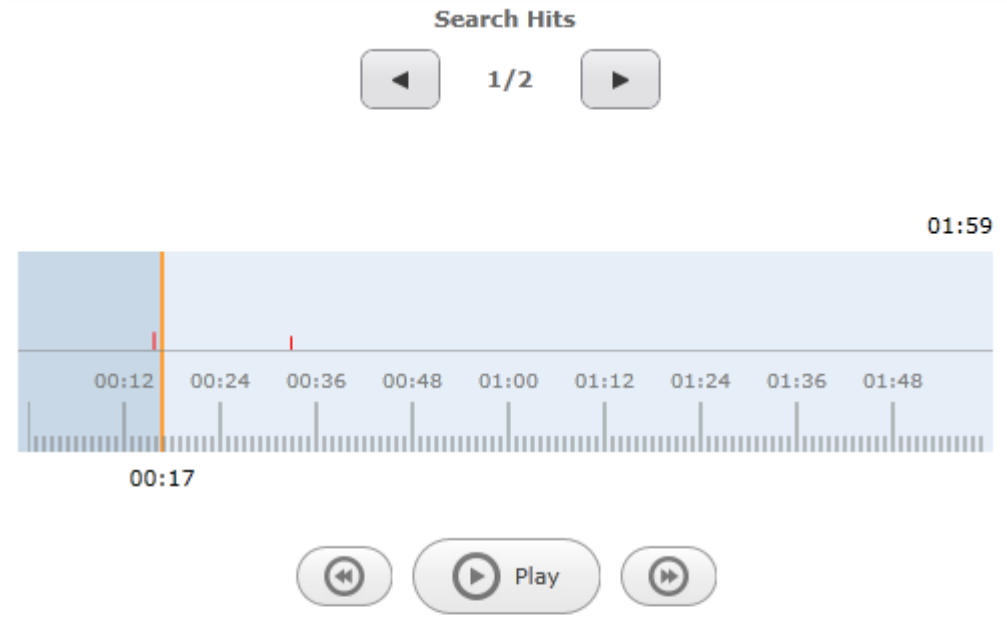


8. When you are finished with defining your search query, click Search.

Result

Your hits will be highlighted in red in the found audio files. The number of hits (Search Hits) can be seen at the top of the audio file. You can listen to the audio file and use the shortcuts to play/pause (P), fast forward (L), rewind (K), go to the next hit (O) or the previous hit (I).

> Use your left mouse button to click anywhere in the timeline to jump forward or backward, either to repeat a sequence or to skip parts of the audio file. You can use this option in both the Play and Pause mode.
> A video file will look the same but with the video playing above the timeline.



Shortcuts:

P - Play/Pause
L - Fast Forward
K - Rewind
O - Next Hit
I - Previous Hit

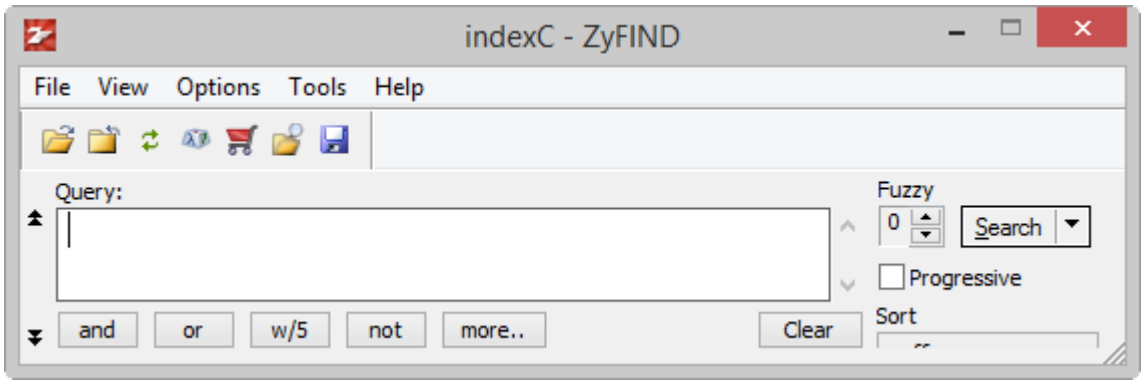
ZyFIND

Conditions

ZyFIND is open.

Instructions

1. Enter a word (query) in the *Query* text box.



2. Select the 'and, 'or', 'w/5', 'not' or 'more..' button to create a more complex search query.
3. Specify the Fuzzy degree (the degree of closeness to your query).

   For TBIE indexes, use Fuzzy Search in your search query with a tilde ~. For each word in your search query you can assign a different fuzzy degree. For example, 'car~1 transportation~3'. For more information, see *Fuzzy Searches* (page 67).

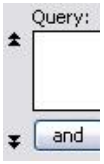
4. If you executed a search already, and want to search in the last results, select Progressive.
5. Click Search.

Result

The search results are listed in ZyResult.

Note

- You can save a (complex and/or long) query via File > Save Query to File (define a name, and click Save). Reuse this query, via File > Open Query from File (select the correct file, click Open).
- Use the arrows at the left side of the screen to minimize or maximize your search possibilities.



- The settings of the Search tools may defer, depending on the settings of the index. If you selected two or more indexes, select the correct tools by selecting an index from the drop down list box.

- Click the arrow button (next to Search), if you want to determine a search method (other than the standard ZyFIND Query, where ZyFIND will look for search operators, wild cards, brackets, etc., in your query):

  - All Words (AND operators are inserted automatically between the words)
  - Any Word (OR operators are inserted automatically between the words)
  - Exact Phrase (all words are automatically put in quotes)

FNAME, FDATE and FPATH

Summary

FNAME, FDATE and FPATH are special content words that can be used in search queries, but *only* in ZyFIND and the Web Client. FNAME locates text files by filename, FDATE by file modification date and FPATH by file path.

> FNAME and FDATE cannot be used in Legal Review. Please use *Field Filter* (page 87) instead.

Syntax	Semantics	Recommended Use
FNAME=query	Where a result of a *query* (a term, phrase or other query) can be found in a *file name*.	Search your file names and restrict the search to a specific area of a file.
FDATE=query	Where a result of a date query can be found in the file modification date field. Use the following format: mm/dd/yyyy	Search your file modification date field and restrict the search to a specific area of a file.
FPATH=query	Where a result of a *query* (a term, phrase or other query) can be found in a *path name*.	Search your file paths.

Examples

Example of query	Results
FNAME=biba*	retrieves all files relating to the Biba case
FNAME=lambada.let	retrieves a specific file, in this case 'lambada.let'
FNAME="4738"	retrieves a file with a number
FDATE=04/05/1988	retrieves files last edited on that date
FPATH="d:\zylab data\index data\myindex"	retrieves file path

Search for All Items

The 'all docs' search function clears search text from the query window and enters @@docs(0). A search returns all items in your index except for TIFF files that have been created in ZySCAN.

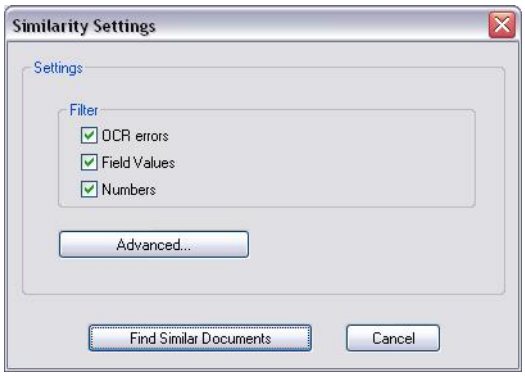
Find similar documents

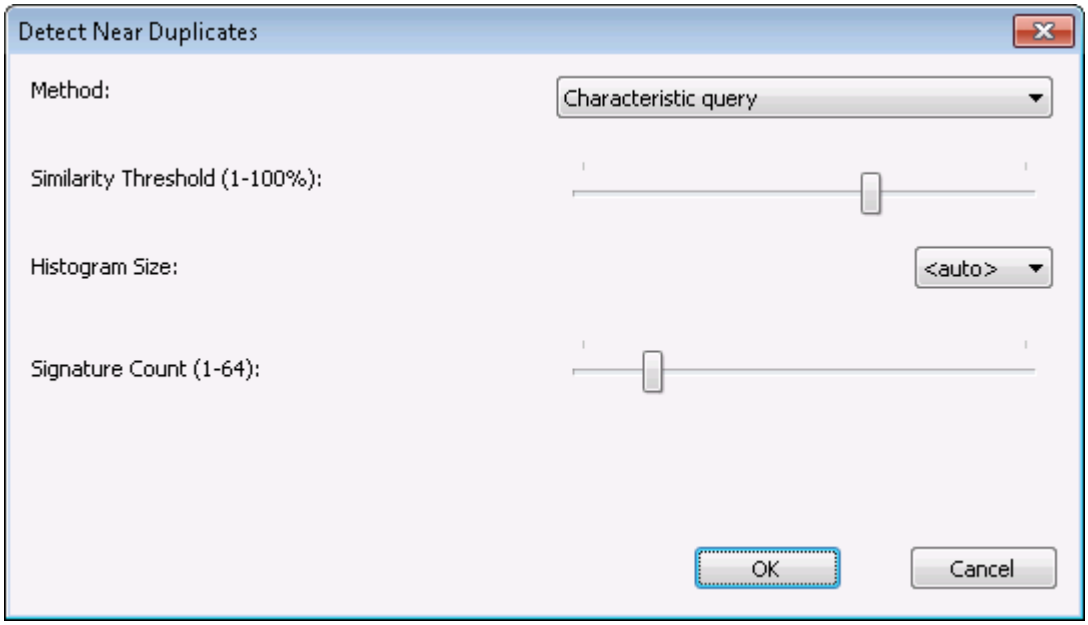
Conditions

You have executed a search and opened a document in ZyVIEW. You can retrieve similar documents or detect near duplicates. The method slightly differs based on the type of index that is being used (HAPI or TBIE).

Instructions

1. Click the "Find similar documents" button:  .
   The Similarity Settings dialog appears when using a HAPI index (for an explanation of the settings see step 2 to 5), the Detect Near Duplicates dialog appears when using a TBIE index (see step 6 to 9).





2. In the Similarity Settings dialog, indicate whether you want to filter OCR errors, field values and/or numbers.
3. If you want to set the Precision and Recall settings, click the Advanced button.

ZyVIEW takes the first N words of the documents and uses it to formulate a quorum search to find at least M (<=N) of these N words. By default N is chosen at 60% of the number of words in the documents with a maximum text buffer of 500 characters and M is set to 60% of N.
The size of N may be set as a precision parameter of the file size and M may be set using the recall parameter. With a large precision few documents will be returned but with many matching words. A large recall will yield a small M so that more documents will be returned. There is a delicate relation between recall and precision in retrieving information from the index. Searching with a large precision is typically for someone who wants to find relevant information but does not care if one or more documents are missed; for example, a journalist trying to find background information for a story. Searching with a large recall is typically meant for someone who wants to find all documents related to a certain topic; for example, a lawyer that needs to find every piece of evidence in a court case.

4. After you have adjusted the settings, click OK.
5. Click Find Similar Documents.
6. When using a TBIE index, in the Detect Near Duplicates dialog, select a method. Currently only one method is available, the Characteristic query. This method will analyze the original document and select characteristic terms (or phrases) which will be compared with other documents.
7. Define the Similarity Threshold. Define in percentages how much the documents should be similar, in which 0% is not similar at all and 100% is an exact duplicate.
8. Define the Histogram Size. A histogram is a graphical representation showing a visual impression of the distribution of data. A digital image can be defined by the width, height and bit depth (1-bit is 2 gray levels (black and white), 2-bit is 4 gray levels, 4-bit is 16 gray levels, 8-bit is 256 gray levels, 16-bit is 65,536 gray levels, 32-bit is 4,294,967,296 gray levels).
   The advantage of increasing the bit depth is that each pixel can represent a greater range of values and record measurements more precisely. The disadvantages are that doubling the bit depth doubles the memory and increases the storage needed for the image.
9. Define the Signature Count. How many signatures (representations of large sets of strings (semantic units) from the documents) do you want to compare to establish the similarity of documents?

Limitations
Near-duplicates are not detected if there is not enough text in it. For example, is does not work for audio, video, images files, as well as for small text documents.

Minimum Requirements for successful detection
- It should be possible to perform text extraction on the document.
- It should be possible to generate at least a requested amount of valid signatures from the text in the document (the default number of signatures is 16 in ZyFind). A valid signature contains 3 different words following each other. Each word should be no longer than 13 chars and no less than 3 chars. Example of correct signature is "test1 test2 test3". Example of incorrect signature is "test 1 test", because words are not unique and number "1" has only one char in it.
- The distance between the first correct signature and the last should be at least 100 words.

Result

You have searched for similar documents. If similar documents are found, they will be presented in ZyRESULT.

Search in current document

Conditions

You have executed a search and opened a document in ZyVIEW.

Instructions

1. Click the "Search in current document" button:  .

2. Enter a search query.
3. If you want to include the current search query, select "Include current hits".
4. Click Search.

Result

You have searched in the current document.

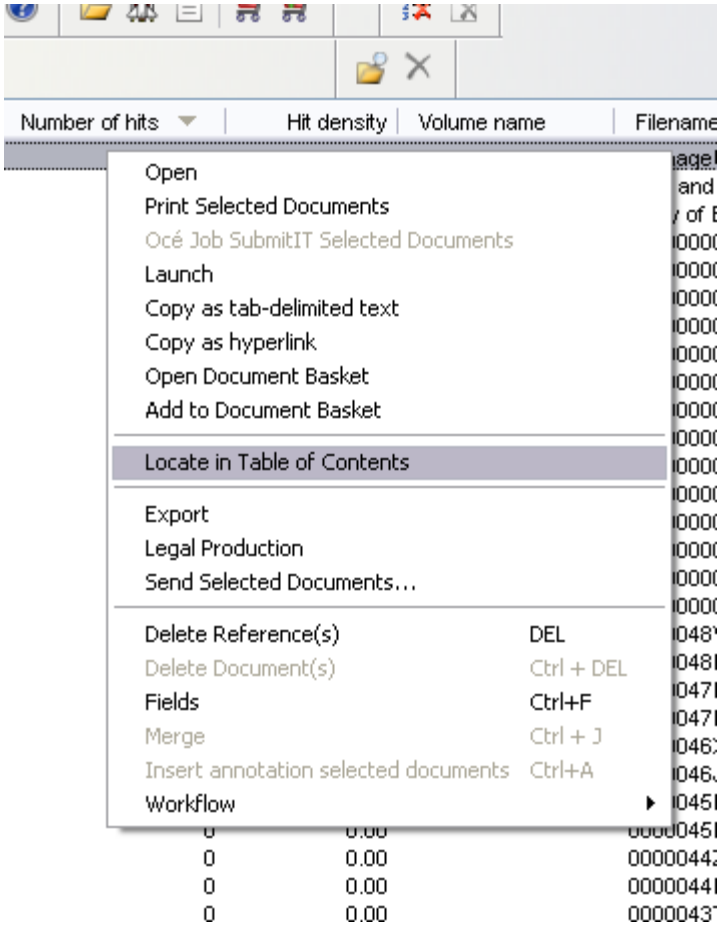
Contents: Locate documents

Conditions

ZyFIND is open. You have done a search and the results are listed in ZyResult. You would like to know whether similar or related documents are available. These documents might be located in the Table of Contents. However, the Table of Contents is large and you are not familiar with its hierarchical structure.

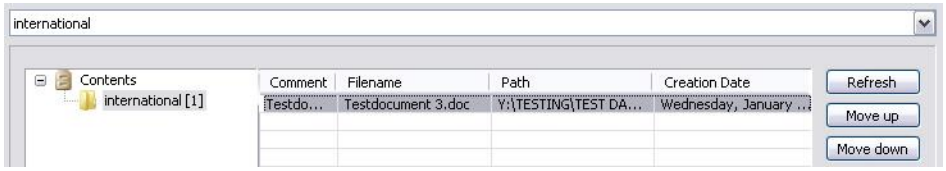
Instructions

1. Select a document from the result list.
2. Click the right mouse button.
3. Select Locate in Table of Contents.



Result

If the document is present in the Table of Contents, the location will be shown.



Note

- You can also locate a document in ZyVIEW. Click the 'Locate in table of contents' button:  . You can navigate through multiple occurrences of the document with 'Locate Next in Table of Contents'.

Contents: Search marked folders

Conditions

ZyFIND is open. You have entered a search query and want to restrict your search to documents in marked folders of the Table of Contents.

Instructions

1. Select a folder from the Table of Contents.
2. Click the right mouse button.
3. Go to Search folders, and select one of the following options:
    - Mark 'selected folder' for searching
    - Mark 'selected folder' and subfolders for searching
4. Select the Search button.

Result

You have searched only the folders that were marked.

Note

- To deselect all marked folders, select Search folders > Unmark all folders.

History: Reuse or delete previous searches

If you cannot remember a successful previous search, look it up in History and reuse it. Be sure to delete any unsuccessful searches.

Conditions

ZyFIND is open.

Instructions

1. Click the History icon: 

. A list with previous searches (max. 100) appears.
2. Double click a previous query to reuse it.

   If added to a query, you can replace the automatically inserted search operator 'or' with your own.
3. Select a previous query and click Delete to remove it.

Result

You viewed, reused and/or deleted previous searches.

Vocabulary: Find related documents

View all searchable words and find all related documents.

Conditions

ZyFIND is open.

Instructions

1. Click the Vocabulary icon: 

. A list with all searchable words appears.
2. Scroll through the list, view the number of occurrences for each.
3. Double click a word to insert it in the Search dialog.

   If added to a query, you can replace the automatically inserted search operator "or" with your own.
4. Click Search.

Result

You viewed all searchable words and searched on one to find all related documents.

Thesaurus: Include synonyms

Broaden the scope of your search statement and include synonyms.

Conditions

ZyFIND is open.

Instructions

1. Type a search statement.
2. Click on a word in your search statement.
3. Click the Thesaurus icon to look for synonyms: 

.
4. View the available synonyms and the number of occurrences for each.
5. Double click synonyms to add them to your search statement.

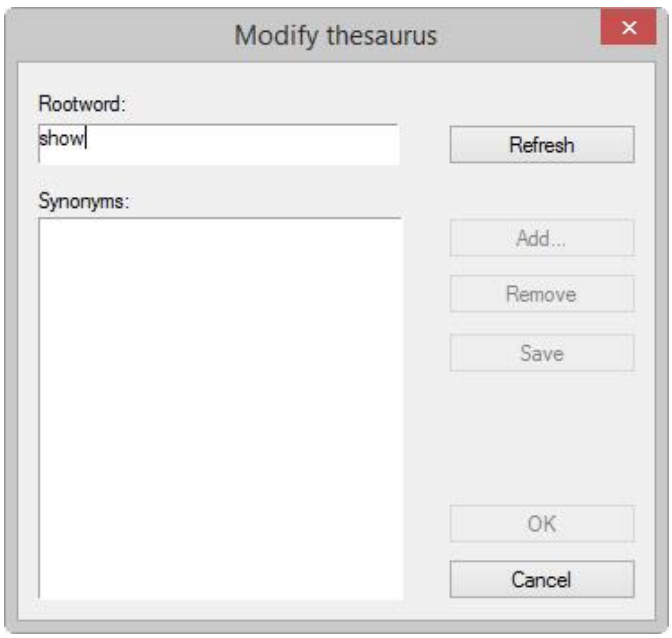
> You can replace the automatically inserted search operator "or" with your own.

Result

You added synonyms to your search statement.

Note

Customize the standard thesaurus with Edit. Add synonyms to a root word, or delete them. Only custom synonyms can be deleted.



Concepts: Predefined complex queries

Concepts are (complex) search statements with a name. Define often repeated searches as concepts and save them for later use. For example, a concept named "legal" could be defined as:

lawyer OR **justice** OR **rechtsanwalt** OR **advocaat** OR **court**

Conditions

ZyFIND is open.

Instructions

1. Click the Concepts icon: 

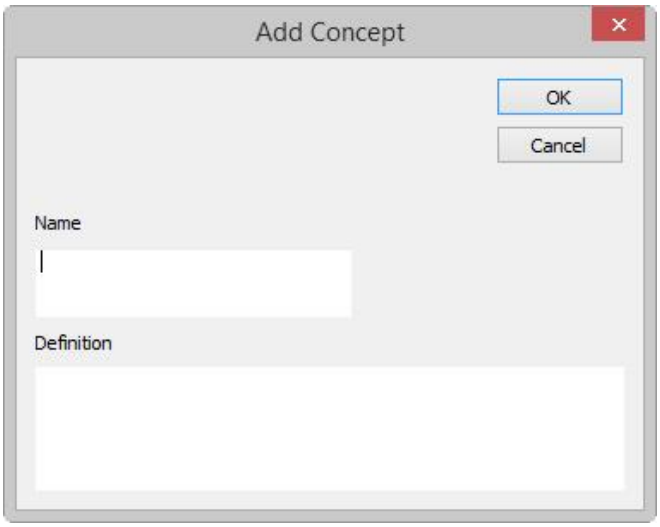
.
2. Search for a concepts in the index related, private or public folders.
3. Double click to insert a concept.

Result

You added a concept to your search statement.

Note

- Click Add to add concepts (fill out a name and definition (search statement)).



- Click Modify to change the definition of a concept.

Fields: Meta information

Search for information **about** your documents. For example, author or description.

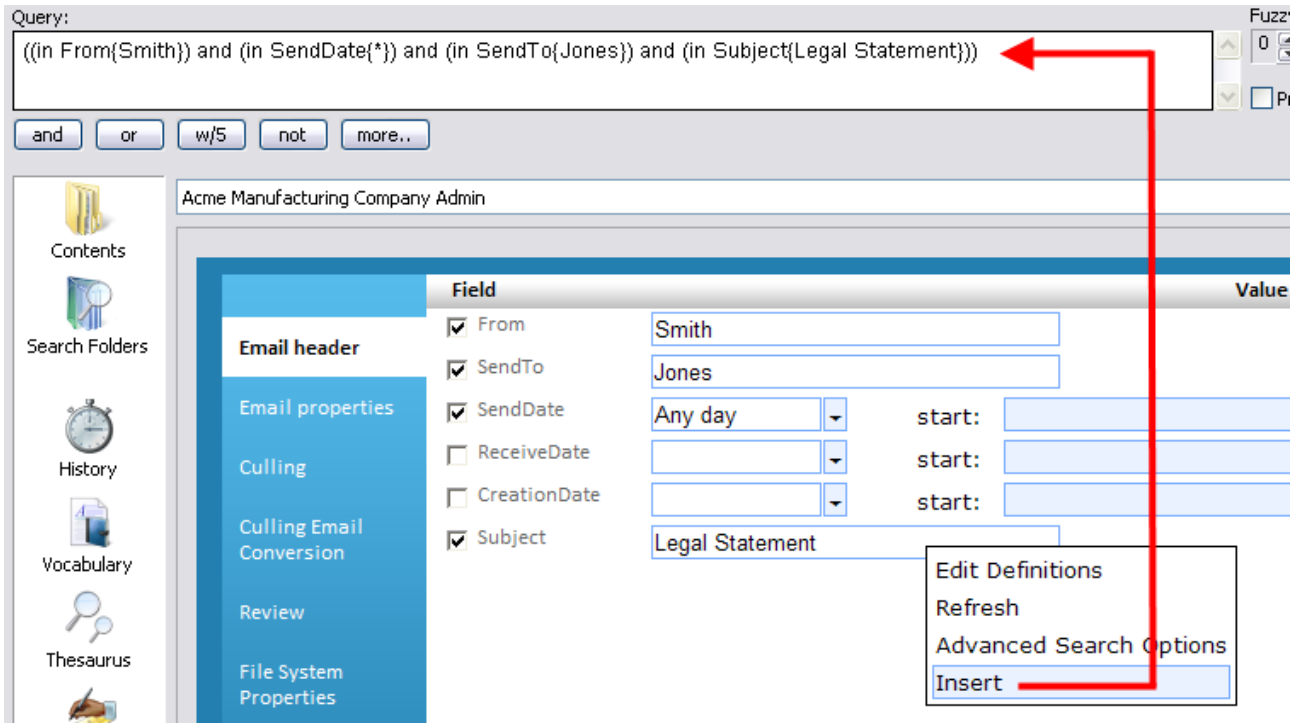
Conditions

ZyFIND is open.

Instructions

1. Click the Fields icon: 

.
2. Select the field(s) you want to search on.
3. Type a field value, select one from a list or calendar, or leave it empty.
4. Optionally, if you want to search the field contents on a number of field values, and edit the search query before executing it, you can use this method. Define the field values you want to search on, click the right mouse button and select 'Insert'. All values are inserted in the full text query box. Edit the search query and execute the search.



Fields can be inserted if:
- Show as search field (Client tab) is selected
- Field storage type is either Document, Document Catalog or XML Wrapper (Storage & Index tab)
- Document Catalog and XML Wrapper fields will only be inserted if:
  TBIE: 'Duplicated/cached' or 'Full text searchable' is selected

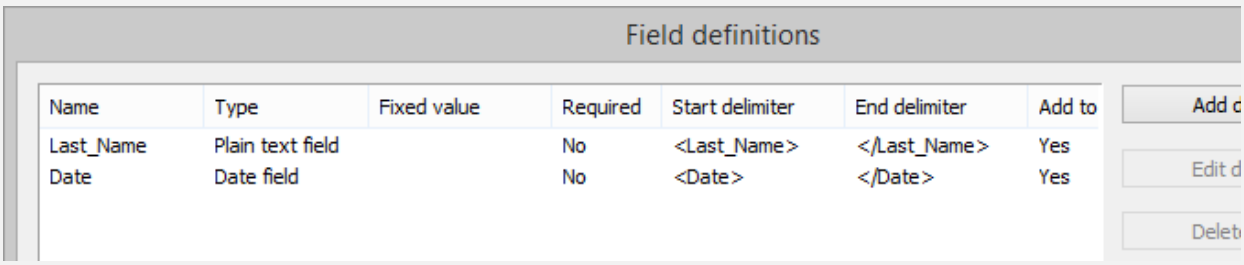
HAPI: 'Full text searchable' is selected

You can check this via Edit Definitions.

Search XML Wrapper fields
If you want to search on fields to get a result list with all populated or empty field values of a specific field, make sure you search an XML Wrapper index with XML Wrapper fields and the option 'Full text searchable' selected in the Storage & Index tab (of the New Field Definition dialog).

Select the Fields icon, click the right mouse button and select Edit Definitions. Use the Id (delimiter) of a field to define a search query.

Query syntax: <delimiter> to "/<delimiter>" {}



Examples are as follows:
name to "/name" {john} will return all documents with the value john in the name field
name to "/name" {*} will return all documents with values in the name field
name to "/name" {} will return all documents with empty values in the name field

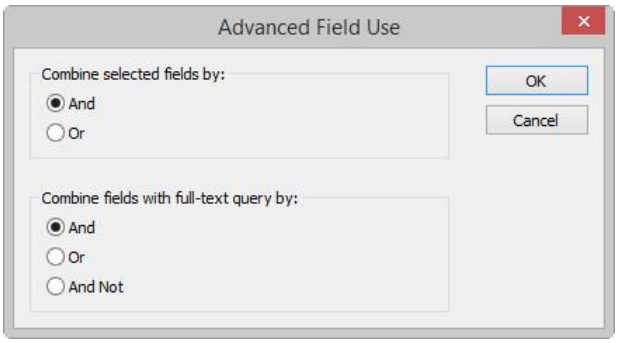
5. Click Search.

Result

A list of results appears in ZyResult.

Note

- Use Refresh to remove all field values.
- Use Advanced Search Options to define the and/or operators:



- Change the look of Fields by choosing a different template.
  a) Click the right mouse button.
  b) Select Edit Definitions.
     The Field Definitions dialog appears.
  c) Select Advanced settings.
     The Advanced Field View Settings dialog appears.
  d) Choose a different template for View/edit values and Query values. Use the VerticalPluginsTab template to group and store the field definitions underneath one tab. Select Show undefined fields, to show all fields (including the ones of which the field values are not defined).
  e) Click OK twice.

Web Client

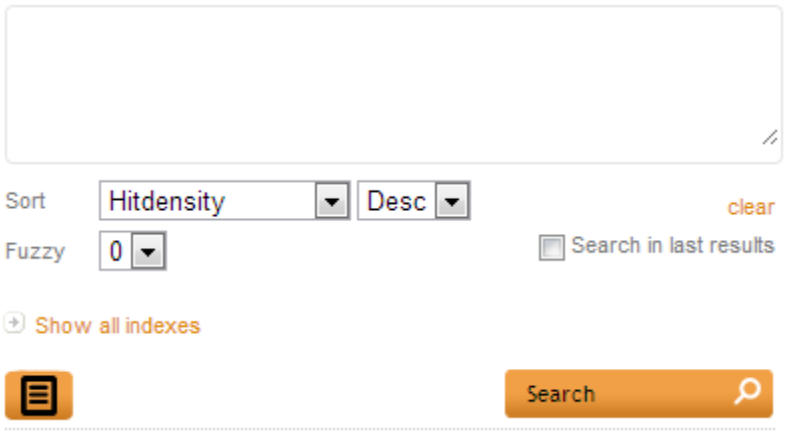
Conditions

A Web Client is open.

Instructions

1. Enter a word (query) in the *Query* text box.

> For more information, see the ZyLAB Web Client for Users manual. This manual is accessible via the Help button in the Web Client (Help > Help Contents).



2. Click Search.

FNAME, FDATE and FPATH

Summary

FNAME, FDATE and FPATH are special content words that can be used in search queries, but *only* in ZyFIND and the Web Client. FNAME locates text files by filename, FDATE by file modification date and FPATH by file path.

> FNAME and FDATE cannot be used in Legal Review. Please use *Field Filter* (page 87) instead.

Syntax	Semantics	Recommended Use
FNAME=query	Where a result of a *query* (a term, phrase or other query) can be found in a *file name*.	Search your file names and restrict the search to a specific area of a file.
FDATE=query	Where a result of a date query can be found in the file modification date field. Use the following format: mm/dd/yyyy	Search your file modification date field and restrict the search to a specific area of a file.
FPATH=query	Where a result of a *query* (a term, phrase or other query) can be found in a *path name*.	Search your file paths.

Examples

Example of query	Results
FNAME=biba*	retrieves all files relating to the Biba case
FNAME=lambada.let	retrieves a specific file, in this case 'lambada.let'
FNAME="4738"	retrieves a file with a number
FDATE=04/05/1988	retrieves files last edited on that date
FPATH="d:\zylab data\index data\myindex"	retrieves file path

Refine Your Results

Using the Refine your results button  you can refine your results by filtering on fields that have content. On the Results Options screen the 'Show refine results' option must be enabled (the default setting is "enabled"). The button shows a list of the fields with content (metadata or a value). Selecting a field or fields filters the results list on those fields. You can filter down to a single document.

Only fields that have the option 'Use for visualization' selected will be displayed in the Refine your results section.

Keyword in Context (KWIC) View

In KWIC (keyword in context) view, the (first) hit is shown in a part of its surrounding text to give an idea of its applicability. The hit is highlighted.

Enable KWIC on the Results Options screen.

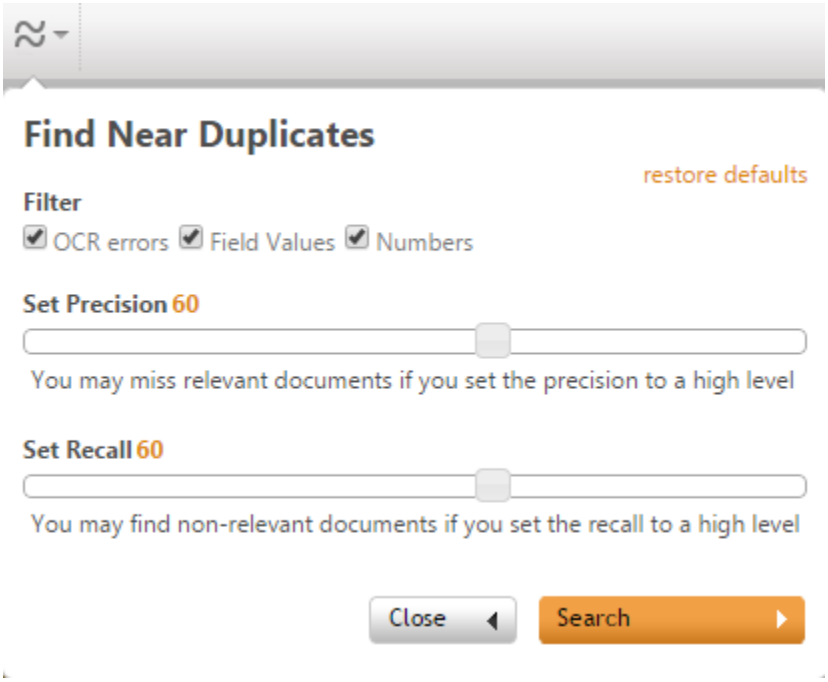
Fast Key Fielding

If you want to change the field values of a set of documents in the web client, using **different field values for each document**, use the Fast Key Fielding mode to walk through your documents:

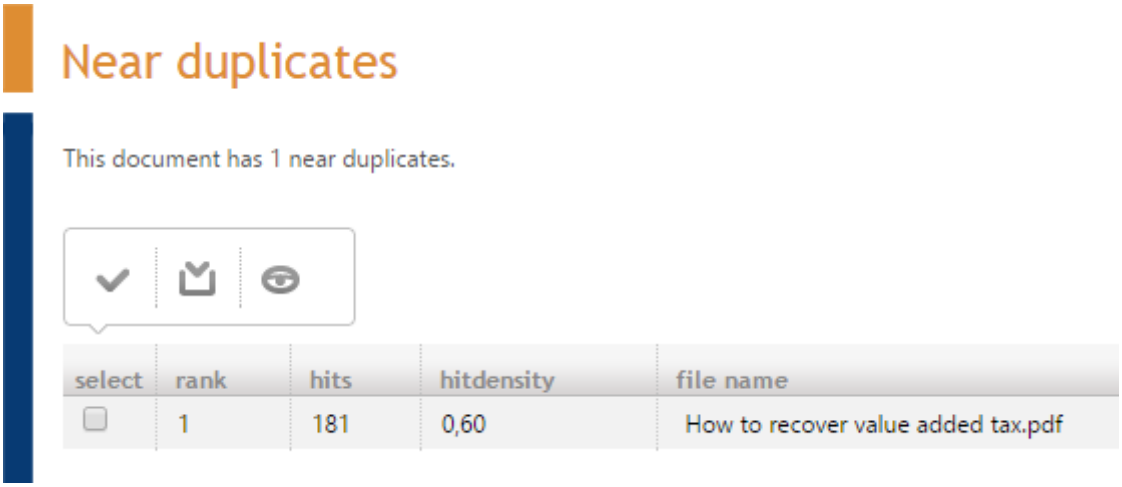
1. Search index(es) using the web client.
2. Select the Open In Fast Key Fielding Mode button  .
3. Change the field values of the first document.
4. Use the Next and Previous arrow buttons (or Ctrl < and Ctrl >) to go quickly through all the documents, changing field values as required.

Find Near-Duplicates

- You have executed a search and opened a document.
- Click the Find near-duplicates button .
  The Find Near Duplicates dialog appears.



- Define if you want to find near-duplicates that are related to OCR errors, Field values or Numbers.
- Set the Precision and Recall.
- Click Search.
- If one or more near-duplicates have been found, you can Select all documents shown, Download selected documents and Show fields.



Search for All Items

To search for all items in your index except for TIFF files that have been created in ZySCAN enter **@@docs(0)** in the search panel.

Contents: Locate documents

Contents allows you to structure your relevant documents in a Table of Contents and TOC sub-folders.

When you have searched for and found documents you can either add one or more selected documents in the results list to a TOC using the Add selected documents button  , or you can add an open document to a TOC using the Add document to the table of contents button  .

The Contents button  shows the table of contents, and its sub-folders if they have been made. You can create new sub-folders by right clicking on the index or a sub-folder.

You can delete a sub-folder by right clicking on a sub-folder and selecting 'Delete folder'.

If you want to search on a selected folder (and sub-folders), select 'Search selected branch only'. The contents of a folder appear in the results panel, from where documents can be viewed.

Search Folders: Add, Edit or Delete

Search Folders are pre-set search queries that automatically group the documents found by the search settings in a dynamic table of contents (TOC). There are two types:

- Index-specific search folders which only find documents in the related index
- Global search folders, which find documents in all the indexes in the Web Client. A global search folder can only be created and added using ZyFIND.

Search Folders can also be created in ZyFIND, which has more options available. Search Folders created in one program are usable in the other program.

Conditions

You want to create, edit or delete Search Folders in Web Client. A Web Client is created, and one or more indexes are added. An optional Global Search Folder can be present.

Instructions

1. Open your web client and select the Search Folders icon  .
2. Right click on the folder you want to edit to display these four options:
    - Edit query
    - Delete query
    - Add folder
    - Delete folder
3. If you selected Edit query, the Edit search folder query dialog appears. You can adjust all settings which are available in ZyFIND too:
    a) In the 'Search for' box, define a full text query.
    b) Select a search method from the dropdown list box.
    - ZyLAB query (ZyINDEX will look for search operators, wildcards, brackets, etc., in your query)
    - All Words (automatically AND operators are inserted between the words)
    - Any Word (automatically OR operators are inserted between the words)
    - Exact Phrase (automatically all words are quoted)
    c) Select Yes or No for Progressive search. A progressive search searches in the last results.
    d) Define the search values of fields you want to include in the search.
    e) Combine the selected fields with AND or OR.
    f) Combine the fields and full-text query with AND or OR.
    g) Click OK.
    h) Refresh the browser window to implement the changes.
4. If you selected Delete query, the query defined for the folder is deleted immediately in both the Web Client and ZyFIND (reopen the index(es) in ZyFIND to see the changes).

5. If you selected Add folder, define the name of the new subfolder, and click OK. The folder is added to the Search Folder Table of Contents in both the Web Client and ZyFIND (reopen the index(es) in ZyFIND to see the changes). Edit the folder's query so it returns a results list - see step 3 above.
6. If you selected Delete folder, the folder is deleted immediately in both the Web Client and ZyFIND (reopen the index(es) in ZyFIND to see the changes).

Result

You have edited Search Folders (in ZyFIND) through the web client.

History: Reuse or delete previous searches

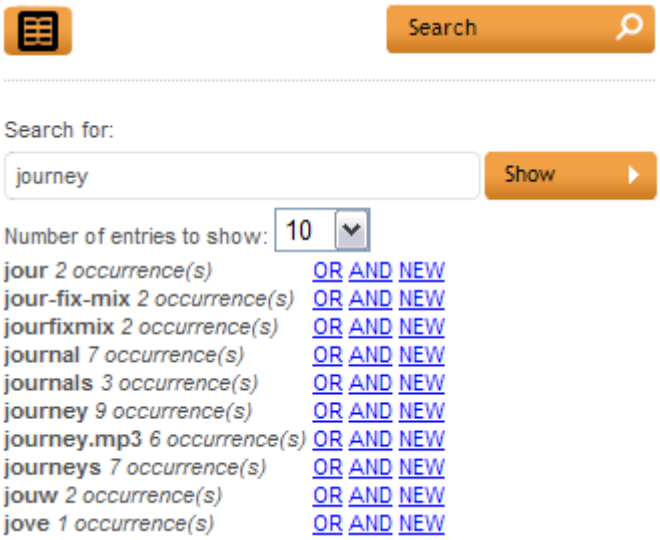
View and re-execute the queries you have made in the past. Click the history button  to show your search history. Click on a search entry to re-execute it.

If you want to use this search tool, go to Options > Search > "Remember query history" and select either "Only when something is found" or "Always" (Search Options).

Vocabulary: Find related documents

The vocabulary tool lists similarly spelled words found in the index. The number of similar words listed is set in the 'Number of entries to show' box.

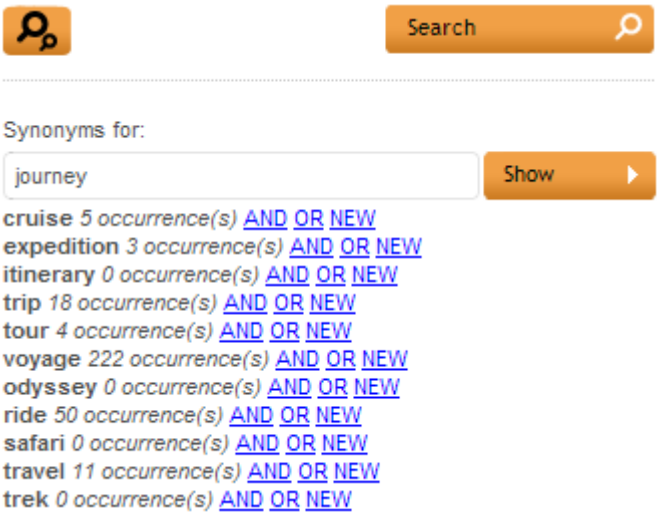
Define the word you want to find, and define the total number of vocabulary entries you want returned. Click Show. A list of words and the number of their occurrences appears, with the defined word in or near the middle of the list. Click the AND, OR and NEW options to add the word to a search query. The results list immediately shows the documents found using the query.



Thesaurus: Include synonyms

Broaden the scope of your search statement by including synonyms.

Click the Thesaurus button  . Define the word you want to find synonyms for. Click Show. A list of synonyms and the number of occurrences in the selected indexes appears. Click the AND, OR and NEW options to add the word to a search query. The results list immediately shows the documents found using the query.



Concepts: Predefined complex synonyms

The Concepts button  displays all available Concepts (predefined queries) that have been defined in the chosen index using ZyFIND.

Concepts are search statements. Define often-repeated searches as concepts in ZyFIND and save them for later use.

Searching on a concept

1. Select a concept.
   All concepts that are defined in the chosen index are displayed.
2. Click Search.

Searching on two or more concepts

1. Select a concept.
2. Click OR or AND of the next concept you want to search on.
3. Repeat step 3 until finished.
4. Click Close.
5. Click Search.

Fields: Meta information

The Fields tool allows you to search in your index's fields; this can be done as fields only, or in combination with a text search.

The Fields tab displays all available fields that have been defined in the chosen index. Note that fields must be defined in ZyFIND or ZyINDEX before they appear in the fields page. The Fields tab has the same function as in ZyFIND: Search on the meta data that is added to the documents during data processing.

Instructions

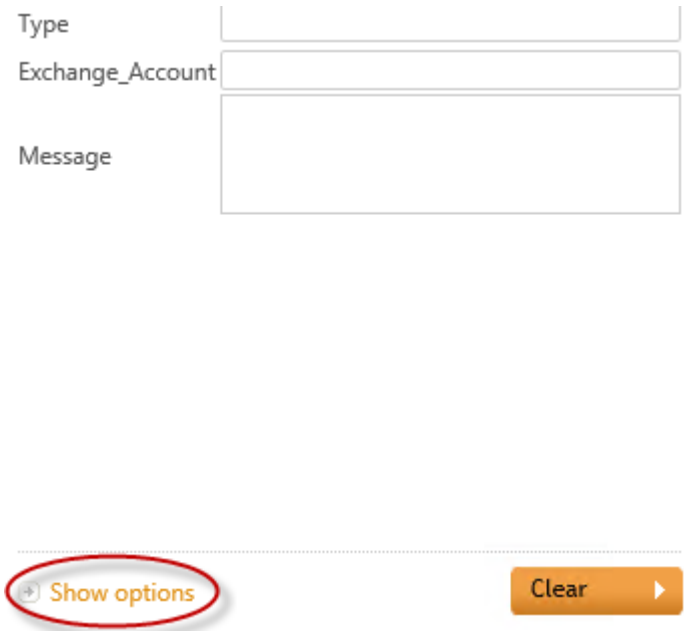
1. Select the fields you want to search in and enter your query.
   All fields that are defined in the chosen index are displayed.
2. Click Search.
3. Select the Clear button to remove the queries.

Result

All files that contain a search term within the area defined as the field are retrieved.

Show Options

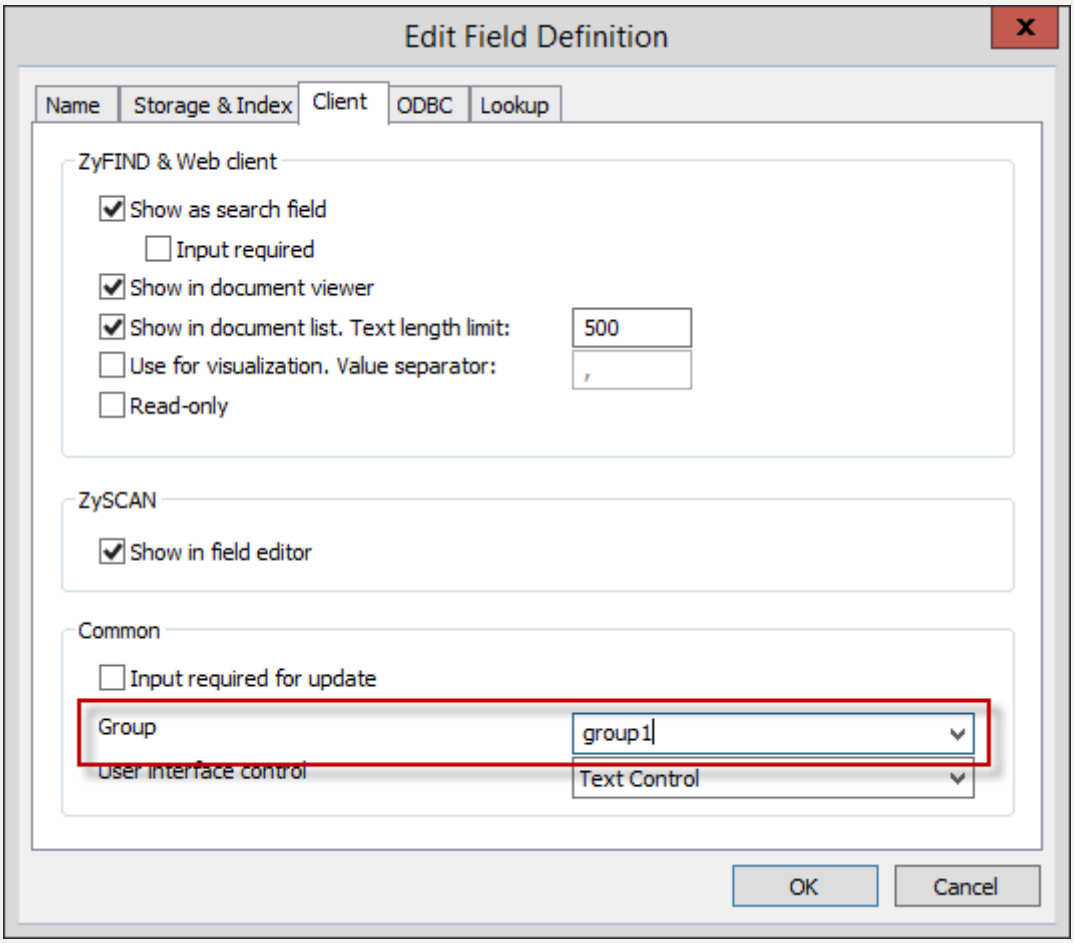
1. Click Show Options at the bottom left corner of the screen to view advanced search field options.

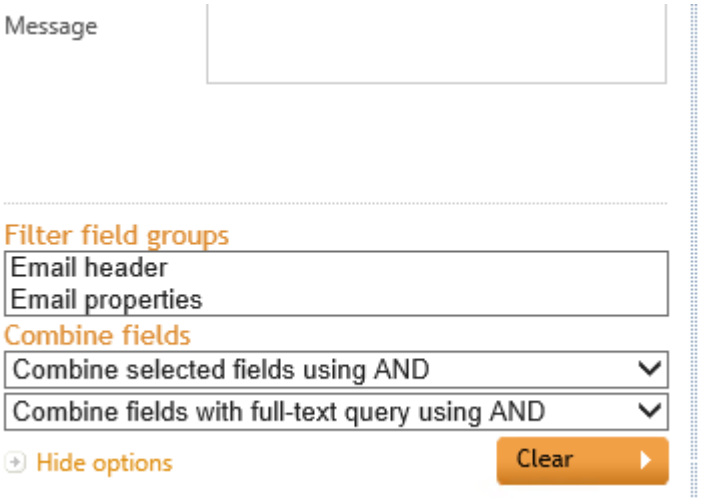


2. You can enhance your search with the following options:
   - Combine Fields
     This allows you to search on combinations of fields (with full-text query) using the parameters AND or OR.

- Filter Field Groups
  This allows you to view your fields grouped as Email header or Email properties. To view all fields again, simultaneously press the Ctrl button and click a group.

The groups Email header and Email properties are automatically added if the index has been created with the Email Archive TBIE template. You can manually add groups by adding a Group name to a collection of fields. For each field of your collection, go to the Edit Field Definition > Client tab and define the Group name.

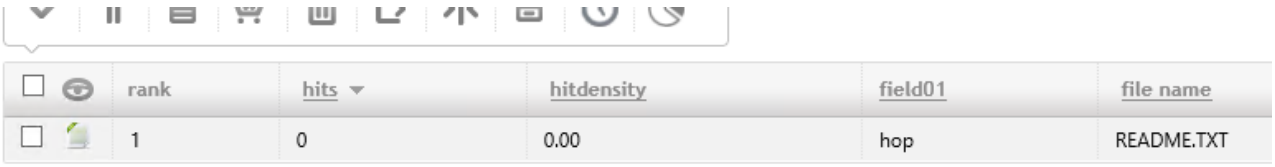




Start New Search for Selected Field

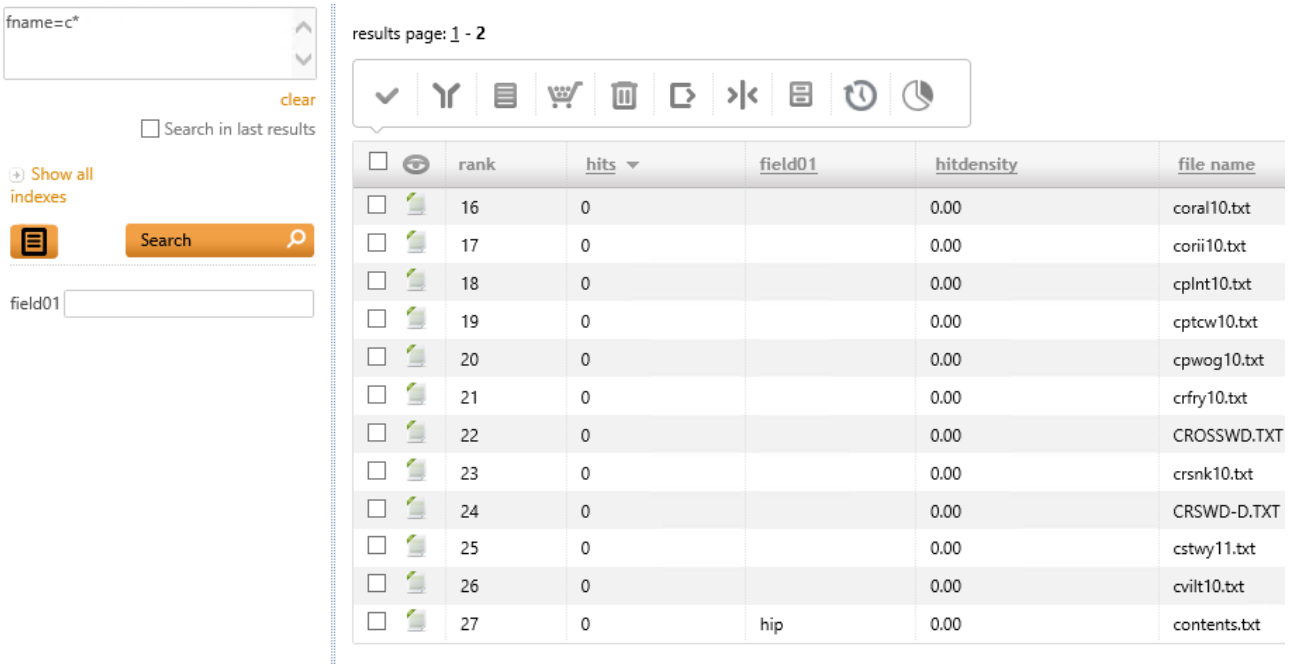
1. Define a broad field search query. For example:

2. Click Search.
3. To narrow down your search for documents with a specific field value, right click on that field value and select 'Start new search for selected field'.
4. Now, only files with the selected field value will be shown in the result list:



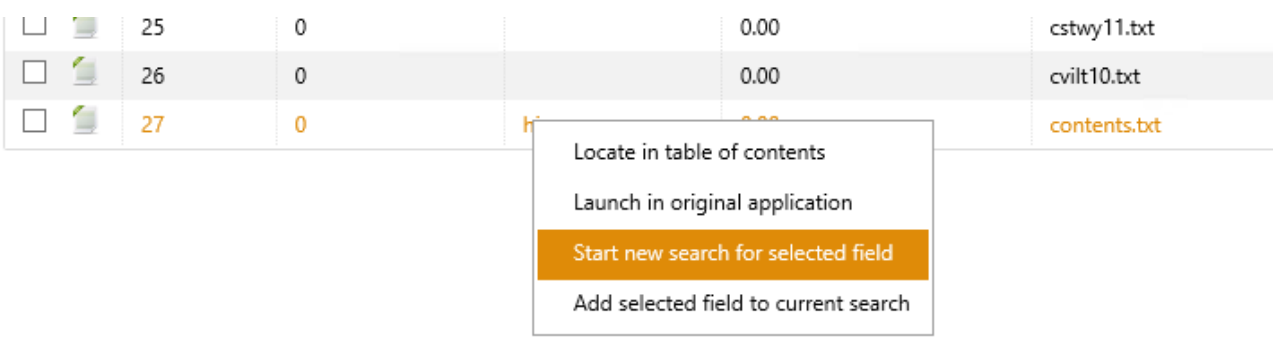
Add selected field to current search

1. Define a full text search query. For example:

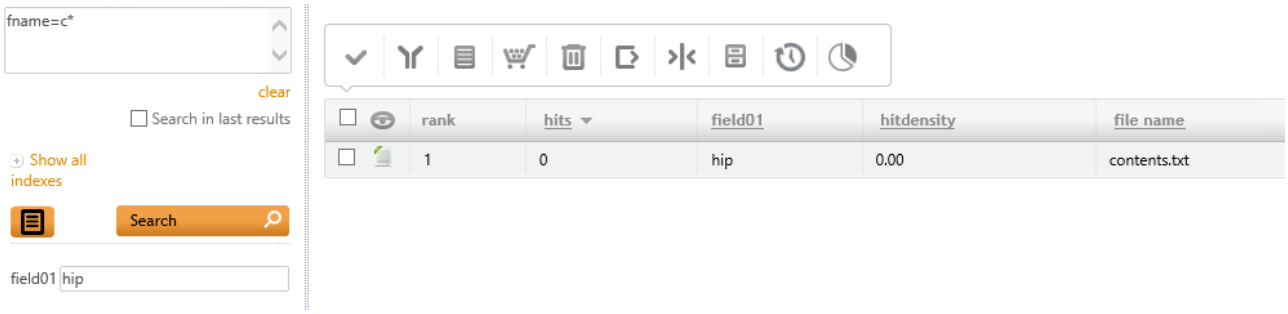


	rank	hits	field01	hitdensity	file name
	16	0		0.00	coral10.txt
	17	0		0.00	corii10.txt
	18	0		0.00	cplnt10.txt
	19	0		0.00	cptcw10.txt
	20	0		0.00	cpwog10.txt
	21	0		0.00	crfry10.txt
	22	0		0.00	CROSSWD.TXT
	23	0		0.00	crsnk10.txt
	24	0		0.00	CRSWD-D.TXT
	25	0		0.00	cstwy11.txt
	26	0		0.00	cvilt10.txt
	27	0	hip	0.00	contents.txt

2. To narrow down your search and add a specific field value to your search, right click on a field value and select 'Add selected field to current search'.



3. Now, only files with the selected field value will be shown in the result list:



The field query will be added to the correct field in the Fields tab. If the field query does not show or you get a warning, you might have the wrong fields displayed (from a different index).

Connect Terms/Queries in Search Query

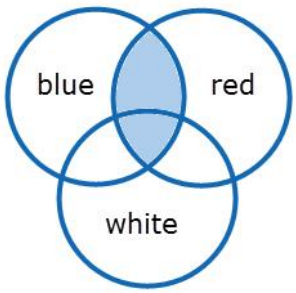
AND Operator

Summary

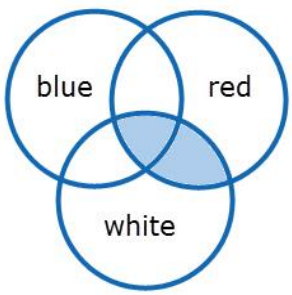
The AND operator is a Boolean search operator that allows you to find files that contain the same terms. Combining terms using the AND operator narrows a search query. It will retrieve all files containing all terms connected with AND, not just one of them. The terms connected with AND can be anywhere in the (audio) file, the position of the terms is not relevant.

See the Venn Diagrams (http://en.wikipedia.org/wiki/Venn_diagram) below for a visual overview of the search results:

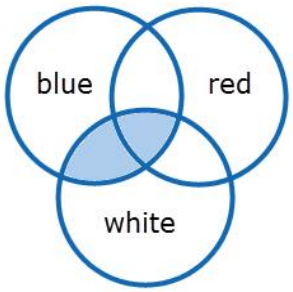
blue AND red



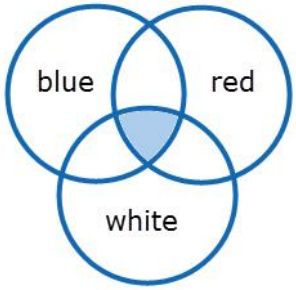
red AND white



blue AND white



blue AND red AND white



Though all your search terms will be included in the results, you may be looking for specific combinations of words or a particular proximity of the search terms. In that case, it is recommended to use phrases or the Within operator, to find more contextually related search results. For the Within operator, the position of the terms is vital. Use the Within operator if you want to specify how close the search terms are connected with each other. For more information about the Within operator, see *Within Operator* (page 59).

When two or more terms are next to each other with no intervening operator, it is a phrase. For example, 'apple pie' will only find documents with the words 'apple' and 'pie' next to each other. When it is not relevant where 'apple' or 'pie' are located in the file, use 'apple AND pie' as the search query.

Syntax	Semantics	Recommended use
left_query AND right_query (left_query) AND (right_query)	Where AND connects two search queries such as a term, phrase or another query. For clarity, the queries can be placed between brackets. Please be aware that sometimes using brackets can influence the search results.	Narrow your search and look for terms that have different meanings. The more queries you connect using AND, the fewer files you get. Only files with all terms you searched on will be retrieved. Use phrases to make your results more specific.
left_query AND NOT right_query	Where AND is combined with the NOT operator to connect two queries.	Narrow your search and exclude terms often connected to your search.
"AND"	Wrap the AND operator in double quotes, so it loses its special meaning.	If you want to search for the term 'and'.

Examples

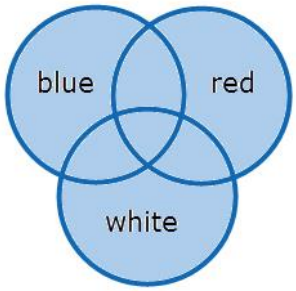
Example of query	Results	Results Explained
president AND america	'president', 'america'	Will only find files that contain both 'president' and 'america'.
global warming AND climate change AND world	'global warming', 'climate change', 'world'	AND is generally used to join different kinds of concepts, different aspects of a question.
apple AND NOT pear	Only files that contain the term 'apple'.	Files that contain both 'apple' and 'pear' will not be found.
bow "and" arrow	'bow and arrow'	Wrap the AND operator in double quotes, so it loses its special meaning.
war W/1 peace	'war peace' and 'peace war'	The W/n operator ensures that the terms 'war' and 'peace' appear close to each other.

OR Operator

Summary

The OR operator is a Boolean search operator that allows you to retrieve all files containing either or both search terms/queries. It is often used to join similar, equivalent, or synonymous concepts. It is also used to search for antonyms and abbreviations.

blue OR red OR white



Syntax	Semantics	Recommended Use
left_query OR right_query (left_query) OR (right_query)	Where OR connects two search queries such as a term, phrase or another query. For clarity, the queries can be placed between brackets. Please be aware that sometimes using brackets can influence the search results.	The more terms you connect using OR, the more files you will find. It will broaden your search.
left_query OR NOT right_query	Where OR is combined with the NOT operator to connect two queries.	Narrow your search and exclude terms often connected to your search.
"OR"	Wrap the OR operator in double quotes, so it loses its special meaning.	If you want to search for the word 'or'.

Examples

Example of Query	Results	Results Explained
car OR transportation	only 'car' only 'transportation' both 'car' & 'transportation'	
u.s OR us	Only 'u.s', only 'us', or files with both instances.	The search result 'u.s' can be followed by a - not highlighted - dot.
e mail OR email	only 'e mail', 'email' or 'e-mail' only 'e mail' and 'e-mail' only 'email' and 'e-mail' only 'e mail' and 'email' 'e mail', 'email', and 'e-mail'	Phrase query combined with term. Hyphenated instances of both phrase and term are also found.
apple OR NOT pear	Files that contain the term 'apple' (the term 'pear' can be present in the file, but not highlighted) and files that do not contain the term 'pear' (no hits, since 'apple' is not present either).	

Tip: If you have a very long OR query like 'aa* or ac* or ad* or ae* or af* or ag* or ah* or ai* or aj* or ak*', use the Quorum operator instead to improve performance: ' 1 of {aa*, ac*, ad*, ae*, af*, ag*, ah*, ai*, aj*, ak*}'.

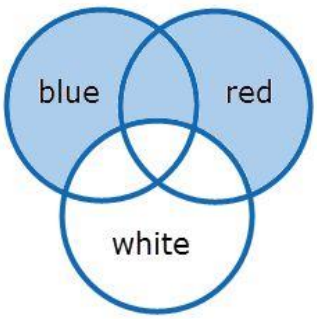
NOT Operator

Summary

The NOT operator is a search operator that allows you to exclude files with certain search terms and narrow your search. It can be used in:

- A Boolean context, where NOT is used stand-alone or in combination with AND/OR to exclude a query (which can be part of a larger query). For example, 'NOT(green apple)', which will find all files where the phrase 'green apple' is not present. It will find files with 'red apple', but only if 'green apple' is not present in the file. In addition, 'red apple' will not be highlighted.
- A positional context, where NOT is used in a sequence of queries. For example, 'NOT(green) apple', which will find all files with two consecutive terms where the first one is not 'green' and the second one is 'apple'. Therefore, you will find files with 'shiny apple' and 'red apple', but not files with 'green apple'.

NOT white



Syntax	Semantics	Recommended Use
NOT (query)	Where NOT is used stand-alone (**Boolean context**) and is connected to a term, phrase or other query. **Note:** 'NOT(a b c)' is not similar to 'NOT(a) NOT(b) NOT(c)', which is an ambiguous search query and will generate an error.	This query will narrow down your result set. Used when you want to find all files where a specific term, phrase or result from a specific query is not present.
NOT(first_query) second_query n_query first_query NOT(second_query) n_query	Where NOT, used in a **positional context**, excludes a query placed in a sequence. NOT can be placed anywhere in a sequence.	Used when you want to specifically omit (at a specific position in the sequence) a word or sequence of words usually connected with the other queries.
left_query TO right_query {sub_query}	NOT is used in a **positional context**, in the sub query of a TO query. See *TO Operator* (page 56).	Used when you want to specifically omit a term or phrase often related to the terms found between the left_ and right_query.

Boolean Context

Example of query	Results	Results Explained
NOT apple	All files that do not contain 'apple'.	Narrow your search and look for files that do **not** contain the query after NOT.

Positional Context

Example of query	Results	Results Explained
warm NOT(green) tea	warm black tea, warm white tea, warm warm tea, warm [anything but green] tea	

For complex search queries containing the operator NOT, see *Complex Search Queries* (page 92).

TO Operator

Summary

The TO operator allows you to search for occurrences of a term or query falling between two other terms or queries.

Syntax	Semantics	Recommended use
left_query TO right_query {sub_query}	Where the sub_query is only retrieved if it falls between the left_ and right_query. The left_ and right_query can be a term or phrase. Negated queries are not allowed. For example, 'apple pie' is allowed, but 'NOT(apple) pie' is not allowed.	This method of searching can be very helpful in files with an accepted standard, such as letters, memos, proposals, contracts, invoices, etc.

Examples

Example of query	Results
dear TO sincerely {John}	'John' Only 'John' is highlighted, dear and sincerely function as start and end delimiters.
(dear OR hello OR hi) TO (sincerely OR regards OR faithfully OR greetings) {John}	**Error** The left_ and right_queries are not terms or phrases.

Example of query	Results
left TO right {sub}	For text 'left sub left right', no match is returned. For text 'left sub right left', one instance of 'sub' is found. For text 'left sub sub right left', two instances of 'sub' are found. For text 'left sub right sub right left', the first instance of 'sub' is found. For text 'left sub left sub left sub right', the last instance of 'sub' is found. For text 'right sub left', no match is returned.
experience TO EOP {(driver OR chauffeur) and (>= 3)}	Locates resumes of persons with a minimum of three years' experience as a driver.
EOP TO EOP {economic AND policy}	Searches the whole document for both economic and policy. Better is it to search for 'economic AND policy'.

IN fieldname{query}

Summary

The IN operator allows you to find terms in fields added to the file (metadata).

> Please note that you cannot use the IN operator in Legal Review. Instead, use the *Field Search* (page 5) or *Field Filter* (page 87).

Syntax	Semantics	Recommended Use
IN fieldname{query}	Where a result of a *query* (a term, phrase or other query) can be found IN the field with the specified *fieldname*. Braces {} mark the start and end of the search within the field.	Search your meta data and restrict the search to a specific area of a file.

Example

Query	Results	Results Explained
IN occupation{dentist}	dentist	The word 'dentist' is found in the field 'occupation'.
IN age{>17}	18, 19, or a higher number	A number higher than 17 is found in the field 'age'.

Within Operator

Summary

The Within operator is used to ensure that your search terms are related to each other. This is called a proximity search, an advanced search where terms can be matched (are related to each other) while avoiding search results where the terms are scattered across a file and are unrelated to each other. You can combine several Within operators in the same search query.

syntax	semantics	Recommended Use
left_query W/n right_query	where n is the maximum number of terms to look for for the left_query to the left and right of the right_query. The left_ and right_query may not contain a NOT query (any context), TO query, Field query (IN) or Field filter query (for example, 'date=10/12/1980').	Limit your search to words that appear within a defined range in either direction.
left_query W/n/term right_query	where term can be a word (for example, `sandwich`), a number or a separator (for example, EOL (End of Line)).	
left_query /n,m/ right_query	where n is the maximum number of terms the left_query may precede the right_query, and m is the maximum number of terms the left_query may follow the right_query.	

Examples

Examples Search Queries	Search Results	Search Results Explained
Tom w/2 John	'**Tom** knows **John** because **Tom** went to school with John's brother.'	The term 'Tom' has token ids 1 and 5, and the term 'John' has token id 3. Therefore, the token at position 1 is within 2 positions of the token at position 3, and so is the token at position 5.
Tom Smith w/2 John Doe	'**Tom Smith** knows **John Doe** because Tom Smith went to school with John's brother.'	The term 'Smith' has token ids 2 and 8, and the term 'John' has token id 4. Therefore, the token at position 2 is within 2 positions of the token at position 4, but the token at position 8 is not.
John Doe w/2 Tom Smith	'Tom Smith knows **John Doe** because **Tom Smith** went to school with John's brother.'	The term 'Doe' has token id 5, and the term 'Tom' has token ids 2 and 7. Therefore, the token at position 5 is within 2 positions of the token at position 7, but not within 2 positions of the token at position 2.
Tom Smith w/3 John Doe	'**Tom Smith** knows **John Doe**, **Tom Smith** went to school with John's brother.'	The term 'Smith' is at positions 2 and 7, and the term 'John' is still at position 4. Therefore, both tokens at positions 2 and 7 are within 3 positions of the token at position 4.
Minnesota W/3/EOS Maine	Minnesota appears within 3 sentences of Maine, and vice versa.	The EOS (End Of Sentence) separator is used.
John /4,10/ Tom	John appears within a maximum of 4 positions before Tom and 10 positions after Tom.	

Performance Tip
Choose W/3 - W/5 to search in approximately the same phrase.
Choose W/15 to search in approximately the same sentence.
Choose W/50 to search in approximately the same paragraph.

Though there is an upper boundary set to 1000, it is advised to choose a number below 100 to prevent too many search results with unrelated search terms.

Precedes Operator

Summary

The Precedes operator allows you to find files in which the first search term precedes the second search term by not more than the defined number of terms. Use this operator when the order in which terms appear is important.

Syntax	Semantics	Recommended Use
left_query P/n right_query (left_query) P/n (right_query)	Where P/n connects two queries such as a term, phrase or another query. *n* defines the maximum number of terms that the left_query should precede the right_query. For clarity, the queries can be placed between brackets. Please be aware that sometimes using brackets can influence the search results.	Find terms in a specific order. Particularly useful in situations where a different order would change the meaning.
(left_query) P/n/term (right_query)	where term can be a word (for example, `sandwich`), a number or a separator (for example, EOL (End of Line)).	

Example

Text	Example Query	Results
'Do you live to work or do you work to live?'	live P/2 work	Will find 'live to work' but not 'work to live' (which has a different meaning).

Number Range Operator

Summary

The Number range operator enables you to search for numbers or number ranges. What is considered a number is tokenizer specific (for more information, see *Introduction to ZyLAB Search Language*).

Syntax	Semantics
<	less than
<=	less than or equal to
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to

Parentheses

Parentheses removes ambiguity when searching for a term or field value. For field value searches, the field name should be followed by a number range operator. If the field is a Boolean field, an error is given as number range operators are not allowed on Boolean fields.

`Month > 10` searches all documents that have a value in field month > 10

`Month (>10)` searches all documents that contain the word 'month' followed by a number > 10

Examples

Example of query	Results
"60615"	60615 **Note**: placed between double quotes it is not a numeric query anymore and you can use fuzzy on it. For more information, see *Fuzzy Searches* (page 67)).
60615	60615, 606154, 060615, etc.
(>=65) w/10 social security	Number 65 or higher within 10 terms from social security.
(> 21) AND high school graduate	22, high school graduate 23, high school graduate etc.
(>1 : <10)	All values between 1 and 10. • 1.000000000001 • 2 • 3.333333 • 9.9999999999999 **Note**: searches of this type take time to execute. A lack of system resources may cause the search to error out.
(<> 5)	Will find documents with numbers, but not with the value 5 in it. At least one other number than 5 should be present. It is not similar to 'NOT 5'. This query will find all documents that do not contain the number 5.

Note

The maximum upper boundary for the numeric repetition range has been set to the maximum length of decoded words, 128, defined by the Tokenizer.

The rules for the numbers within numeric repetition:

1) \{m\} where m is a positive whole number such that 0 < m <= 128

2) \{m,n\} where m and n are positive whole numbers such that 0 <= m <= 128, 0 < n <= 128, n > m

Quorum Operator

Summary

Use the Quorum operator to search for a specified number of terms in no particular order from a list.

Syntax	Semantics
n of {subquery1, subquery2, ...}	Finds all files that match at least n different sub queries. All hits of all sub queries that are in the files will be highlighted. *n* is equal to or less than the total number of terms in the list.

Examples

Example of query	Results
2 of {history, english, social studies, french, dutch, german}	• history, social studies • dutch, german • history, english, german • english, social studies, dutch, german • Etc. Always at least 2 or more from the list.
1 of {blue, green, red} equals blue OR green OR red	• blue • green • red • blue, green • blue, green, red • Always at least 1 or more from the list.
3 of {blue, green, red} equals blue AND green AND red	blue, green, red

Define Terms in Search Query

Fuzzy Searches

Use Fuzzy searches to find variations of a term, including the ones that were not recognized correctly during the conversion by the OCR engine. You can choose from 4 fuzzy degrees (the degree of closeness to your query). Fuzzy degree 2 is recommended for normal text searches. This provides for mistakes such as broken and joined characters. Use Fuzzy degree 3 or 4 only if you are searching for long terms. If multiple terms are used in the query, fuzzy will work on each term separately.
You can use four integer values (1, 2, 3, 4) or floating numbers in the range from 0 to 1 (0,1, 0,2).

The fuzzy algorithm is based on the Levenshtein distance:
http://en.wikipedia.org/wiki/Levenshtein_distance

Example query	Fuzzy degree	Results
computer email	0	computer, c-omputer, computer- email, e-mail
computer~1	1	computer, c-omputer, computer-, commuter, compute, computter, compoter, etc.
computer~2	2	computer, c-omputer, computer-, commuter, compute, computter, compoter, computw, cumpoter, etc
computer~3	3	computer, c-omputer, computer-, commuter, compute, computter, compoter, computw, cumpoter, ounputer, ccnpoter, chapter, etc.
computer~4	4	computer, c-omputer, computer-, commuter, compute, computter, compoter, computw, cumpoter, ounputer, ccnpoter, chapter, colour, charter, ocnputter, etc.
computer~0,5	0,5	computer, c-omputer, computer-, commuter, compute, computter, compoter, computw, cumpoter, ounputer, ccnpoter, chapter, colour, charter, ocnputter, combined, complete, complex, compared, center, footer, copied, collate, cope, matter, etc. **Note**: If a floating number is not accepted, try a higher number or a longer term.
"232"~1	1	232 230, 231, 233, 234, 235, 236, 237, 238, 239 202, 212, 222, 242, etc. 032, 132, 332, 432, etc. 22, 23, 32 22-4-1999

Wild Card Searches

Wild cards are used to replace or represent one or more characters in a term, making the search query more flexible and efficient.

Performance Tips
Prefer ? over *

Prefer character range [a-z] over ?

Avoid using a search query like '*c*'. This will search through the whole data set. When looking for singular and plural forms of short terms that have a common root, such as 'set', the * symbol may retrieve too many irrelevant terms, while the ? symbol may retrieve too few terms. In such a case, do not use wild cards and instead OR together the forms of the term you want, e.g., 'set OR sets'.

Question Mark ?

Summary

The question mark (?) matches a single character. To match ? as-is, escape it \?
However, please be aware that if the ? symbol was placed at the end of a word/number and followed by a space, it will by default be recognized as a separator and not indexed. If you want to change that, please contact support (http://help.zylab.com).

Syntax	Semantics	Recommended Use
t?rm	Where ? represents one character inside, before or behind a term.	
?er? t??m	The question mark can be used twice or more in a term, also in combination with each other.	Do not use the question mark too often, since this will slow down the search process.
t??*	This will match a word starting with t, followed by two or more characters.	

Examples

Example of query	Results
wom?n	woman, women
?andy	candy, dandy, sandy, handy, etc.
sh??e	shore, shade, shape, etc.
60?	600, 601, 602, 603, 604, 605, 606, 607, 608, 609
60\?	During indexing, the ? symbol followed by a space indicated the end of a sentence and was not indexed. Therefore, '60?' will not be found. If you want to find ? symbols at the end of words/numbers, you need to change the character set (please contact support (http://help.zylab.com)).
60?60	60?60, 60060, 60160, 60260, 60360, 60460, 60560, 60660, 60760, 60860, 60960

Asterisk *

Summary

* matches zero or more characters. This wild card is useful when you are not sure what you are looking for, but have a general idea. It expands a search term to include all forms of a root word, e.g., patent* retrieves patent, patents, patentable, patented, etc.
In addition, it can be used to find alternative (mis)spellings of a word, e.g. colo*r retrieves both color and colour, and useful* retrieves useful, usefull [sic], usefullness [sic] and usefulness.

Syntax	Semantics	Recommended Use
term term	Where * can be placed before, inside or right after a term.	Use it to expand your search terms to include all forms of a root word or to find alternative spellings.

Examples

Example of query	Results
*most	most, almost
auto*	auto, automobile, automotive, automatic, autobiography, autocracy, autograph, etc.
auto AND automo*	auto, automobile, automotive, auto-redaction, auto-increment
soft	microsoft, microsoftware, soft, software, ..
colo*r	color, colour
ch*ter	charter, character, chapter

Character Patterns []

Summary

[] matches a single character and/or range that is contained within the brackets. To match [] as-is, escape it \[\]. However, the [and] symbols are by default not included in the character set and will not be indexed and therefore not found. If you want to be able to search on these symbols, please contact support (http://help.zylab.com).

The - character (used in the character range) is treated as a literal character if it is the last or the first character within the brackets: [abc-], [-abc].

Backslash escapes are allowed in sequence search and bracket expressions.

> Wild cards *, ?, + do not have a special meaning inside [ ]. For example, a search 'a[abc*?+]m' will match characters in range as-is, so terms 'abm' 'a*m', or 'a+m' will return a match.

Syntax	Semantics	Recommended Use
[character(s)]	Where one or more characters are placed between two brackets.	Use it to match one character.
[character-range]	Where a character range is placed between two brackets.	Use it to match one character from a range of characters.

Processing Rules

Sequence Rule	[abc]	The sequence characters are matched one at a time. For range [abc], 'a' is matched, then 'b', then 'c'
Range Rule	[a-c]	The range matching is inclusive, i.e. for the pattern [a-k]pple, both words 'apple' and 'kpple' qualify. A valid range is defined as [c1 - c2] where c1 <= c2. For example, 'a-c' is a valid range but 'c-a' is not. 'a-' is a valid range too. This will match 'a' or '-'.
Negation Rule	^ (caret)	The caret (^) has a special meaning only inside the square brackets and only if it is the first character to follow the open square bracket, '['. Caret negates characters within the square brackets when it is the first character. If '^' is not the first character within the square brackets, it loses its special meaning and is matched directly as an ordinary character.

The following wild card range patterns match the word 'apple':

[abc]pple	character sequence
[a-k]pple	character range
[a]pple	one-character sequence
[^b]pple	caret (^) negates characters in sequence and/or range

Examples

Example of query	Results
[abc]	'a', 'b', and/or 'c'. Also, 'c-omputer'.
m[ae]n	'man' and/or 'men'
m[a-e]n	man, mbn, mcn, mdn, men
m[a-z]n	All terms with three letters that start with an **m** and end with an **n**.
[a-z]	any single lowercase letter from 'a' to 'z'
[abcx-z] or [a-cx-z]	'a', 'b', 'c', 'x', 'y', and/or 'z'
[a-z][a-z][a-z][a-z][a-z]	All terms with 5 letters. Also, hyphenated terms of which at least one part has 5 letters or less. For example 'c-omputer'.
[a-z][a-z][0-9][0-9][a-z][a-z]	All terms that start with 2 letters and are followed by 2 numbers and 2 letters.
[-abc]	'-', 'a', 'b', and/or 'c'. Also, 'c-omputer'.
[abc-]	'-', 'a', 'b', and/or 'c'. Also, 'c-omputer'.
[a-]pple	Matches 'apple' and '-pple'
[c-a]pple	Not a valid range. Error returned: 'Query contains an invalid wild card pattern.'
[0-9]	any natural number from '0' to '9'
[a^bc]	'a', 'b', 'c'. Also, 'c-omputer'. The ^ sign will not be found because it is not indexed by default. You can change this by adjusting the character set. Please contact http://help.zylab.com

Negation [^]

Summary

[^] negates a single character and/or range that is contained within the brackets. Literal characters and ranges can be mixed. If the caret symbol (^) is not placed at the start, the search engine will search for occurrences of the symbol itself. However, the ^ symbol is by default not included in the character set and will not be found. If you want to be able to search on the ^ symbol, please contact support (http://help.zylab.com).

Syntax	Semantics	Recommended Use
[^character(s)]	Where one or more characters (preceded by the caret) is placed between two square brackets.	Use it to negate one or more character.
[^character- range]	Where a character range (preceded by the caret) is placed between two square brackets.	Use it to negate a range of characters.

Examples

Example of query	Results
[^abc]	any single character (numbers, letters) other than 'a', 'b', or 'c'. Also, if this single character is part of a hyphenated word. For example 'e-mail'.
[^a-z]	matches any single character (numbers, letters) that is not a letter from 'a' to 'z'.
[^b]pple	'apple', but also '-pple', 'cpple', 'dpple', 'cpple', etc.
[a^bc]	'a', 'b', 'c'. Also, 'c-omputer'. The ^ sign will not be found because it is not indexed by default. You can change this by adjusting the character set. Please contact http://help.zylab.com
m[^b-y]n	Only 'man' and 'mzn' will give a hit, because the ^ sign excludes the range b-y.
m[a^b-s]n	man, mbn, mcn, mdn, men, mfn, mgn, mhn, min, mjn, mkn, mln, mmn, mnn, mon, mpn, mqn, mrs, msn.

Preceding Element +

Summary

+ matches the preceding element one or more times. The element can be:

- literal, e.g. 'a', 'z'
- literal range, e.g. [abc], [a-bx-z]
- '?' wild card
- number

Syntax	Semantics	Recommended Use
character+	Where one character is followed by the + sign.	Use it to match the preceding character one or more times.
[character- range]+	Where a character range is followed by the + sign.	Use it to match the preceding character range one or more times.
n+	Where a number is followed by the + sign.	

Examples

Element	Example of query	Results
literal	ab+c	abc, abbc, abbbc, abbbbc, and so on, but not ac
literal range	[abc]+	a, aa, aaa, bb, cab, ba, bca, and so on (all single instances and combinations of a, b and c). Also, a hyphenated word like 'c-omputer'.
	[a-dx-z]+	ax, axax, bzaxbzax, and so on (all single instances and combinations of a, b, c, d, x, y z). Also, a hyphenated word like 'c-omputer'.

Element	Example of query	Results
'?' wild card	?+	All words highlighted.
	b?+	All words starting with b, as the ? symbol resembles every character.
number	10+	10, 100, 1000000, and so on

Preceding Element {m,n}

Summary

{m,n} matches the preceding element at least *m* and not more than *n* times. The element can be:

- literal, e.g. 'a', 'z'
- literal range, e.g. [abc], [a-bx-z]
- number

Syntax	Semantics	Recommended Use
character(s){m,n}	Where one or more characters is followed by {m,n}	Use it to match the preceding character at least *m* and not more than *n* times.
[character- range]{m,n}	Where a character range is followed by {m,n}	Use it to match the preceding character range at least *m* and not more than *n* times.

Examples

Example of query	Results
a{3,5}	only 'aaa', 'aaaa', 'aaaaa'
ba{2,3}b	'baab', 'baaab', **but not** 'bab', 'baaaaab'
ba{0,3}b	'bb', 'bab', 'baab', 'baaab', **but not** 'baaaab', 'baaaaaab'
[0-9]{2,7}	23, 601, 060615 and all other combinations from 2 to 7 numbers.
2{1,3}	2, 22, 22-4-1999

Preceding Element {m}

Summary

{m} matches the preceding element exactly *m* times. The element can be:

- literal, e.g. 'a', 'z'
- literal range, e.g. [abc], [a-bx-z]
- '?' wild card
- number

Syntax	Semantics	Recommended Use
character(s){m}	Where one or more characters is followed by {m}	Use it to match the preceding character(s) exactly *m* times.
[character- range]{m}	Where a character range is followed by {m}	Use it to match the preceding character range exactly *m* times.

Examples

Example of query	Results
[0-9]{3}-[0-9]{4}	matches local phone number: 123-4567
[0-9]{3} - [0-9]{4}	Error: Open brace expected. Solution: Do not use spaces!
[0-9]{7}	1234567, 123-4567, 1-999999, 22-4-1999
ISSN [0-9]{4}-[0-9]{4}	matches 'ISSN 4793-2940'
ISSN [0-9]{8}	matches 'ISSN 4-7932940', 'ISSN 47-932940', etc. The hyphen can be placed anywhere in the number.
ba{3}b	baaab
?{2}	is, us, de-facto, etc.
60{2}	600

Preceding Element {m,}

Summary

{m,} matches preceding element at least *m* times. The element can be:

- literal, e.g. 'a', 'z'
- literal range, e.g. [abc], [a-bx-z]
- '?' wild card
- number

Syntax	Semantics	Recommended Use
character(s){m,}	Where one or more characters is followed by {m,}	Use it to match the preceding character(s) at least *m* times.
[character- range]{m,}	Where a character range is followed by {m,}	Use it to match the preceding character range at least *m* times.

Examples

Example of query	Results
ba{2, }b	Error: Unexpected input. Do not add a space behind the comma!
ba{2,}b	'baab', 'baaab', **but not** 'bb', 'bab'
ba{0,}b	'bb', 'bab', 'baab', 'baaab', ...
b?{3,}b	baaab, baaaab, baaaaab, etc.
[a-z]{15,}	transliteration, functionalities, Files\ZyLAB\Information, etc.
3{2,}	33, 333, etc.

Multiple Wild Cards in a Search Query

Rules for Multiple Wild Cards in a Pattern

Allowed

Combinations	Examples
Adjacent character sequences and/or ranges within square brackets	[^a][^p][^p][^l][^e]
Adjacent '?'	a??
Adjacent '?' and character sequences and/or ranges within square brackets	[^a][^p][^p]??
Adjacent '?' and '*' wild cards	ab?*?*?
Adjacent '?' and character range	ab[abc]?[d-z]
Adjacent '*' and character range	ab[abc]*[d-z]
Wild card '+' followed by wild card '?'	ab+?
Wild card '+' followed by wild card '*'	ab+*
Wild card '+' followed by '[ ]'	ab+[abc]

Not Allowed

Combinations	Examples	Comments
Adjacent '*' wild cards	e** will return an error	Error 4100: Adjacent '*' wild cards are meaningless, and thus are not allowed
Wild card '+' or numeric repetition range \{m,n\} at the beginning of a pattern followed by any wild card	++, +*, +?, +[abc], \{m,n\}+, \{m,n\}*, \{m,n\}?,\{m,n\}[abc]	Error 4153: Wild card '+' by itself is meaningless since it has to follow some valid symbol
Wild card '+' or numeric repetition range \{m,n\} following wild card '*'	*+, * \{m,n\}	Error 4153: Precedence rules for wild card '+' are violated

More Search Techniques

Field Filter

Summary

Find terms in field names using 'fieldname=query'.

Syntax	Semantics	Recommended Use
fieldname=query	Where a result of a *query* (a term, phrase or other query) can be found in a *fieldname*.	Search your field names and restrict the search to a specific area of a file.

Example

Query	Results	Results Explained
fieldname=colo*r	color, colour	The word 'color' or 'colour' is found in a fieldname.

Exclude List of Terms from Fuzzy/Wildcard Query

Summary

When performing a search with wild cards or a fuzzy term, it is possible to exclude terms or a subset of terms defined by wild cards or a fuzzy query.
Exclusion lists are supported only by the TBIE engine. The HAPI engine will return an 'Unexpected Operator' error whenever the exclusion list syntax is used. Similarly, the Audio bundle will return this error for the exclusion list use.

syntax	semantics

`query -{exclusion_list_query_1, ..., exclusion_list_query_n}`	• in which `query` is a wild card search, e.g. `ab*`; or a fuzzy search, e.g. `constitution~2`
	• if `query` does not have a wild card or fuzzy term, a syntax error will be returned
	• in which `exclusion_list_query_n` is a
	  • term, e.g. `apple`,
	  • wild card search, e.g. `ab*`, or `app[a-m]e`, or `app?e`,
	  • fuzzy search, e.g. `constitution~2`
	• Empty term lists are allowed and will be ignored
	  • A search query `a* -{ }` will return all terms in a dictionary starting with a letter 'a'
	• Terms in the exclusion list should be separated by commas, otherwise a syntax error will be returned
	• Wild card query within the exclusion list cannot have an exclusion list
	  • A search query `a* -{abc*, avocado}` is valid, but a search query `a* -{avocado, ap* -{apple}}` is not allowed because `ap*` has an exclusion list with term 'apple', thus, an error will be returned: 'Error 4150: Unexpected nested exclusion list'.
	• Terms and wild card queries within exclusion lists may be quoted
	  • A search query `a* -{abc*, avocado}` and `"a*" -{"abc*", "avocado"}` will return identical results
	• A wild card query with wild card queries in the exclusion lists might take longer than just the original wild card query
	  • A search query `a* -{abc*, *do}` might take longer than a search query `a*`
	• An unquoted wild card query with an exclusion list must have a white space between the wild card query and the '-' symbol indicating introduction of the exclusion list. A quoted wild card query can omit the white space.
	  • A search query `a* -{avocado}` is correct because '-' is preceded by a white space
	  • A search query `a*-{avocado}` will be parsed as a single search term: "a*-{avocado}". Usually no results will be found.
	  • A search query `"a*"-{avocado}` is correct because the wild card query is bordered by double quotes

Examples

Examples of allowed queries
• o* - {olive, or*,other} • o* - {ol[a-m]ve, or*} • o* - {ol?ve, or*} • o* - {olive~2, *r)
• *e - {olive} • *e - {or*, *ve} • *e - {ol[a-m]ve} • *e - {ol?ve} • *e - {olive~1, above}
• constitution~4 - {constitution, cong*} • constitution~4 - {*o, constitution~2} • co* - {constitution~1, cohabitation, cong*, conb*}

Quoted and Unquoted Searches

Examples Unquoted Search

Type of search	Example Query	Results
full-text search	a* - {avocado}	supports exclusion lists, finds 'apple' in text.
field filter search	field1=t* - {talk}	returns an error: '`Error 4150: Unexpected nested exclusion list.`'
IN fieldname{query}	in field1{t* - {talk}}	supports exclusion lists, finds 'test' in text 'simple test'.

Examples Quoted Search

Type of search	Example Query	Results
full-text search	"a* - {avocado}"	does not recognize exclusion lists: assumes they are part of the search pattern. No results.
field filter search	field1="t* - {talk}"	does not recognize exclusion lists: assumes they are part of the search pattern. No results.
IN fieldname{query}	in field1{"t* - {talk}"}	supports exclusion lists, finds 'test' in text 'simple test'.

Complex Search Queries

Summary

Complex search queries contain multiple operators and/or wild cards or other search methods (fuzzy search, field search, excluded lists).

Examples

Example of query	Results
car AND (IN <fieldname> {rental})	Files containing 'car' and a specified field with the value 'rental' are selected.
1 of {o* - {other, often, out}, p* - {process, procedure}, const* - {constitution, constitute, constant}}	Files containing at least one word starting with 'o', 'p' or 'const'. The files may contain the words 'other', 'often', 'out', 'process', 'procedure', 'constitution', 'constitute', or 'constant', but these words will not be highlighted.
2 of {(common law), (tort and (chief engineer) and due diligence), 1 of {breach, litig*, process, damag*, liability}}	Files containing at least two results from three search queries (query 1: common law, query 2: tort and chief engineer and due diligence, query 3: 1 of breach, litig*, process, damag*, liability). For example, the result might be: 'common law', 'breach' or 'tort', 'chief engineer', 'due diligence', 'damage'
contract and 2 of {(bidding or rate or placement or offer or pro rata or exclu* clause), (breach or litigation or arbitration or damag*), (honor* or due diligence), (limited liabil* or warrant* or liquid*)}	Files containing the word 'contract' and at least two results from three search queries. For example, the result might be: 'contract', 'bidding', 'offer', 'breach'
valu w/10 (*book* or mark* or public*) w/20 (wrong or *correct)	The result might be: 'The right (and wrong) way to market a book. The value of a book...' Where 'wrong', 'market', 'book' and 'value' will be highlighted.
(sell* TO book* {(Donoghue or Ferris or King or Moyes)}) AND (2016 or 2017)	The result might be: 'In 2017, we are selling the new thriller from Stephen King in our bookstore.' Where 'King' and '2017' will be highlighted.
u.s.	No search results. A dot followed by a space is by default recognized as a separator by the tokenizer; The last '.' will not be part of the token. Therefore, the abbreviation u.s. will not be found, nor will u.s be found. To find the abbreviation u.s., search for 'u.s'

Hyphenated Text

Example 1

Text: 'some text left-right apple orange pear left-apple-right'

Example of query	Results
left TO right {apple}	We expect just 'left-apple-right' to be found, however, 'apple' following 'left-right' is also found. This is because both 'left' and 'right' tokens in 'left-right' text are the same token and so are the 'left' and 'right' tokens in 'left-apple-right' text. This is because hyphenated tokens get the same token id during tokenization process. Since left and right query tokens are the same, we proceed to move the right query to the next occurrence and match sub query. Thus, the extra 'apple' word is matched.
EOP to EOP {apple}	Here, token ids for left and right query will be same since it is the same query, and we indeed want to match everything between the current token of EOP and the next token of EOP.

Example 2

Text: 'fresh apple-banana pie'

Where 'fresh' has position 1, 'apple' has position 2 and 'banana' has position 2 (since they are combined with a hyphen), 'pie' has position 3.

Example of query	Results	Results Explained
NOT(apple) banana pie	fresh apple-banana pie	First the term 'pie' is matched at position 3, then at position 2 'banana' is matched, next, NOT(apple) is matched with 'fresh' at position 1. In this example, 'apple' will be skipped.
apple NOT(banana) pie	no results	First the term 'pie' is matched at position 3, then at position 2 we will first find 'banana', but that does not match the query NOT(banana). Therefore, no results are returned.

Note 1: Since you may not know upfront if a combination of words is hyphenated, it is advised to try different combinations of search queries when you suspect a hyphenated word might be part of the results.

Note 2: If a word or combination of words you are searching for contains a hyphen, that word will often be found, even if you did not include a hyphen in your search query. For example, when you search for 'email' or 'e mail', it will also find 'e-mail'. However, 'e mail' will not find 'email' or the other way around ('email' will not find 'e mail').

Complex Search Queries using the NOT Operator

The processing rules for the NOT operator differ depending on the context in which it is used. Nevertheless, the NOT operator can be used several times in both a Boolean and positional context in a single query. For example, 'NOT(green AND NOT black) tea', in which the first NOT is used in a positional context and the second NOT is used in a Boolean context.

Syntax Examples

Syntax	Semantics	Recommended Use
left_query AND NOT right_query (left_query) AND NOT (right_query)	Where NOT is used in combination with AND (**Boolean context**) and connects any query, such as word(s), wild card or regular expression (regex) query, sequence query, etc. For clarity, the queries can be placed between brackets. Please be aware that sometimes using brackets can influence the search results.	The combination of AND with NOT is often used to refine search results after a previous query. Finds documents where the left_query is present and the right_query is not.
left_query OR NOT right_query (left_query) OR NOT (right_query)	Where NOT is used in combination with OR (**Boolean context**) and connects any query, such as word(s), wild card or regular expression (regex) query, sequence query, etc.	Finds documents where the left_query is present and/or where the right_query is not present.
NOT (left_query) AND NOT (right_query)	Where both the first stand-alone NOT and the second NOT used in combination with AND are used in a **Boolean context**.	Finds all documents where neither the left_query nor the right_query occur.

Syntax	Semantics	Recommended Use
NOT (left_query) OR NOT (right_query)	Where both the first stand-alone NOT and the second NOT used in combination with OR are used in a **Boolean context**.	Finds documents where at least one of the queries is not present.
NOT (first_query second_query n_query)	Where NOT is used stand-alone (**Boolean context**) and is connected to a sequence (several terms) at one position (through parenthesis). **Note:** 'NOT(a b c)' is not similar to 'NOT(a) NOT(b) NOT(c)', which is an ambiguous search query and will generate an error.	This query will narrow down your result set. Used when you want to find all documents where a specific word, phrase or results from specific queries are not present.
NOT(first_query) second_query n_query	Where NOT, used in a **positional context**, excludes any query placed in a sequence. NOT can be placed anywhere in a sequence.	Used when you want to specifically omit (at a specific position in the sequence) a word or sequence of words usually connected with the other queries.

Syntax	Semantics	Recommended Use
first_query NOT(second_query) n_query	Where NOT, used in a **positional context**, excludes any query placed in a sequence. NOT can be placed anywhere in a sequence.	
left_query TO right_query {sub_query}	NOT is used in a **positional context**, in the sub query of a positional TO query. See *TO Operator* (page 56). Where *left_query* and *right_query* can be a word query or a sequence query without negated sub-queries and *sub_query* can be any valid query. For example, {word NOT word}, {NOT word} or {word AND/OR NOT word}.	Used when you want to specifically omit a word or sequence of words often related to the words found between the left_ and right_query.
NOT(left_query AND NOT right_query) query	Where the first NOT is used in a **positional context** and the second NOT is used in a **Boolean context**.	

Syntax	Semantics	Recommended Use
NOT(NOT left_query AND NOT right_query) is equal to: left_query OR right_query	Where all NOT operators are used in a **Boolean context**.	
NOT(NOT NOT term AND NOT term)	**incorrect syntax** Will result in an error because of nested NOT operator in a Boolean context in sub query 'NOT NOT term'.	
NOT(first_query) NOT(second_query) NOT(third_query)	**incorrect syntax** Search query will generate an error.	Too many combinations are possible, making the search query pointless.

Query Examples

Example of query	Results	Results Explained
NOT apple AND NOT orange	All documents that contain neither 'apple' nor 'orange'.	
apple AND NOT pear	Only files that contain the term 'apple'.	
apple OR NOT pear	Files that contain the term 'apple' (the term 'pear' can be present in the file, but not highlighted) and files that do not contain the term 'pear' (no hits, since 'apple' is not present either).	
NOT(green AND NOT black) tea	Will find 'black tea', 'white tea' and just 'tea'.	Because it was double negated, it will find black tea.
NOT(green AND black) tea	Will find 'white tea', 'warm tea', just 'tea'. Will **not find** 'green tea' and 'black tea'.	
leaf AND NOT tree	Only documents where 'leaf' is present.	Narrow your search and exclude terms often connected to your search.
used cars AND NOT cars	**no results**	No error generated, it is syntactically correct, but semantically meaningless.
cars AND NOT used cars	Will find 'cars', but not 'used cars'.	
apple AND NOT(orange or pear)	Will find documents with only 'apple' in it, but not documents with 'apple' and 'orange' or 'pear'.	
apple OR NOT(orange pear)	Will find 'apple'. And all documents that do **not** contain the phrase 'orange pear'.	

Example of query	Results	Results Explained
used cars OR NOT cars	Will find documents with 'used cars'. Documents with only 'cars' will not be found.	
cars OR NOT used cars	Will find documents with 'cars'. Also, every word in front of 'cars' will be found/ highlighted (except when that word is 'used'). These documents can also contain the phrase 'used cars', but 'used' will not be highlighted.	
NOT (NOT(apple))	**no results, error**	Nested NOT query is not allowed.
NOT NOT apple	**no results, error**	Nested NOT query is not allowed.
NOT (NOT apple)	**no results, error**	Nested NOT query is not allowed.
apple NOT(orange) pear	matches sequence 'apple <anything but orange> pear'.	
apple NOT(orange OR pear)	matches sequence 'apple <anything but orange and pear>'	NOT combined with a Boolean OR query.
apple NOT(orange AND pear)	matches sequence 'apple <anything but orange and pear>'	NOT combined with a Boolean AND query.
warm NOT(green) tea	warm black tea, warm white tea, warm warm tea, warm [any other word than green] tea, etc.	
apple TO orange {pear NOT yellow}	matches text where word 'orange' follows 'apple' and sequence 'pear <anything but yellow>' in between (order is important).	

Example of query	Results	Results Explained
apple TO orange {pear AND NOT yellow}	matches documents with 'orange' following 'apple' and 'pear' and something not 'yellow' in between the two terms (order not important).	
NOT apple TO orange {pear}	**no results, error**	'NOT apple' is a NOT query, which is not allowed in a positional filter TO query.
apple TO (orange NOT pear) {yellow}	**no results, error**	'orange NOT pear' is a sequence query with a negated sub-query, which is not allowed in a positional filter TO query.

Boolean Context: Negated Sequence

Query matching at the beginning of a document

If a sequence query contains a number of negated terms followed by a positive term that matches the first token in a given text, the first single token will be matched. Consider the text 'apple pie smells good' and a sequence query 'not(banana) not(orange) apple'. The query matches the first word, 'apple', in the text above.

Query matching at the end of a document

If a sequence query contains a number of trailing negated terms preceded by a positive term that matches the last token in a given text, the last token will be matched. Consider the text 'apple pie smells good' and a sequence query 'good not(red) not(blue)'. The query matches the last word, 'good', in the text above.

Macros

Use macros for complex search queries that will be used regularly. Contact your Administrator for more information on the type of macros that can be used. By default, you can use macros for date manipulations, e.g. to be able to find document that have been created, modified or sent 2 days, 5 weeks, 7 months or 3 years ago. With other macros you can search for social security numbers, amounts, holidays, ip numbers, etc. For the complete list, see below. For the complete queries, see the QueryMacros.config file. In this file the predefined queries can be adjusted. Please contact your Administrator.

Instructions

1. By default, the following macros can be used to find files that have been created, modified or sent a number of days, weeks, months or years ago:
    - #DaysAgo(<number>)#
    - #WeeksAgo(<number>)#
    - #MonthsAgo(<number>)#
    - #YearsAgo(<number>)#

The mm/dd/yy format will be used for the evaluation, relative to the current date. Leap years and the number of days in a month are taken into account. If the resulting day is not a valid day in the resulting month, the last valid day of the resulting month is used.
#DaysAgo(0)# is resolved in 00:00 of the current day.
Note: Date db type is not supported in Legal Review, only DateTime. Hence "=" is not applicable with these types of macros.

2. The following macros can also be used:
    - #ssn#
      Query: (Social Security* or Soc Sec or SSN* or SSNS or SSID) w/2 [0-9]{3}-[0-9]{2}-[0-9]{4}
    - #amount#
      Query:
      (1 of {EUR, CAD, USD, GBP, US$, $, £, €} w/1 1 of {[0-9]+, [0-9]+?[0-9]+, [0-9]+?[0-9]+?[0-9]+, [0-9]+?[0-9]+?[0-9]+?[0-9]+, [0-9]+?[0-9]+?[0-9]+?[0-9]+?[0-9]+})
      or
      (1 of {[$€£][0-9]+, [$€£][0-9]+?[0-9]+, [$€£][0-9]+?[0-9]+?[0-9]+, [$€£][0-9]+?[0-9]+?[0-9]+?[0-9]+, [$€£][0-9]+?[0-9]+?[0-9]+?[0-9]+?[0-9]+})
      or

(1 of {[0-9]+[$€£], [0-9]+?[0-9]+[$€£], [0-9]+?[0-9]+?[0-9]+[$€£], [0-9]+?[0-9]+?[0-9]+?[0-9]+[$€£], [0-9]+?[0-9]+?[0-9]+?[0-9]+?[0-9]+[$€£]})

- #holiday#
  Query: 1 of { <comma divided list of holidays> }
  Note: For the complete query, see the QueryMacros.config file.
- #month#
  Query: 1 of {january, february, march, april, may, june, july, august, september, october, november, december}
- #day#
  Query: 1 of {monday, tuesday,wednesday,thursday,friday,saturday, sunday}
- #date#
  Query: ([0-3][0-9]?[0-3][0-9]?[12][09][0-9]{2}) or
  (((1 of {jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec} or #month#) w/1 [0-9]{1,2}) w/1 [12][09][0-9]{2})
- #ip#
  Query: *[1-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}*
- #nonlatin#
  Query: [^a-z0-9]+
- #DutchAddress#
  Query that searches for Dutch streets, postal codes and cities
  Note: For the complete query, see the QueryMacros.config file.
  Please be aware that very long queries take some time to process.
- #DutchStreetPostalCode#
  Query that searches for Dutch streets and postal codes
  Note: For the complete query, see the QueryMacros.config file.
  Please be aware that very long queries take some time to process.
- #DutchCity#
  Query: 1 of { <comma divided list of Dutch cities> }
  Note: For the complete query, see the QueryMacros.config file.
  Please be aware that very long queries take some time to process.
- #DutchPostalCodeCity#
  Query: (([1-9][0-9]{3} [A-Z]{2}) or ([1-9][0-9]{3}[A-Z]{2})) p/2 (1 OF { <comma divided list of Dutch cities> })
  Note: For the complete query, see the QueryMacros.config file.
  Please be aware that very long queries take some time to process.
- #DutchIBAN#
  Query: 1 of {(NL[0-9]{2} [A-Z]{4} [0-9]{4} [0-9]{4} [0-9]{2}), (NL[0-9]{2} [A-Z]{4}[0-9]{10,11}), (NL[0-9]{2}[A-Z]{4}[0-9]{10,11}), (NL[0-9]{2} [A-Z]{4} [0-9]{10,11})}

3. This functionality is available in:

    - Legal Review
        - Search
        - Document-Level Security
        - Auto-Classification
        - Production
        - Batch Assignment
        - Keyword Highlighting
    - Legal Review Collector

Appendix A: Search Rules & Conventions

- Investigate/know your data set, look at the vocabulary and keep that data set in mind when designing queries.
- Formulate your expectations and test your queries.
- Operators are placed only between search terms, except NOT.
- NOT can be placed at the beginning of a search statement, but never at the end.
- Two operators cannot appear in sequence, except NOT (AND NOT, OR NOT).
- Operators are reserved words, therefore, you cannot use them as content words. For example, the search statement **and OR or** will not be accepted.
- Avoid using long AND/NOT chains. Instead, use "1 of {worda, wordb, wordc}".
- Capital and lowercase letters are considered identical. We show operators in capitals for emphasis and clarity.
- An operator can appear more than once in a search statement.
- Use ? over *.
- Use [a-z] over ?.
- Avoid using a search query like "*c*". This will search through the whole library.
- Use positional queries over proximity queries. Positional queries are more precise.
- The **W/n** operator must include an integer in the range 1 to 16382, followed by a space and a content word.
- You can use one term to retrieve both the hyphenated and non-hyphenated spellings of a term; for example, the search term:

  - database retrieves database and data-base, but not data base

  - data-base retrieves data-base, but not database and data base

  - data base retrieves data-base and data base, but not database.

- Searches with duplicate words in series also find single occurrences of that word; for example, the search statement, **sing sing**, would find single occurrences of *sing* as well as the phrase, *sing sing.*
- Words that are normally hyphenated are recognized, for example, Winston-Salem.
- All printable characters in the ASCII character set are recognized.
- A sentence-ending period and other trailing punctuation marks are ignored, when a space or a carriage return follows. Periods are recognized when followed by a character, as in I.B.M. or in 292.004. Apostrophes are treated as null characters, and are therefore ignored.
- It is not possible to search for capitalized letters, since all terms in the dictionary are stored in lower case.

Appendix B: Definitions

Character

A character can be a letter, a number or a symbol like %, @, &, ^, *, etc.

> Please be aware that by default many symbols are not indexed and will not be found unless the character set is adjusted.

See also: *Reserved Characters* (page **111**)

Character Map (or Set)

The character map (or set) determines which characters are used to separate terms, which characters are indexed, which ones are used for punctuation, etc. All possible characters that can be recognized and searched on are listed in the character map. By default, some characters are not indexed and will not be found unless the default character map is adjusted. How characters are defined in the character map, influences the outcome of a search. For example, when brackets are set to be separators, the following text will be identified as 3 terms: 'most definite(ly)'.
For more information on the character map and how to configure it, please contact support (http://help.zylab.com).

Element

An element is a character or a range of characters.

Fuzzy

Fuzzy searches are used to find all variations of a word, including the ones that were not recognized correctly during the conversion from paper files to digital files. For more information, see Fuzzy Searches.

Hyphenated terms

Hyphenated terms are not uncommon (for example, sugar-free). Each part of a hyphenated term will get the same token id. However, this does not mean that the hyphenated term will be seen as one term. This is important to know, especially when you consider that when a sequence query is being processed (and a hyphenated term might be what you are looking for), the query is processed from back to front.

We will explain how this will effect your search with an example search query using the NOT operator:

Consider the text "fresh apple-banana pie", where "fresh" has position 1, "apple" has position 2 and "banana" has position 2 (since they are combined with a hyphen), "pie" has position 3. As we start processing backwards, the search results can be very different depending on your query.

Example of query	Results	Results Explained
NOT(apple) banana pie	fresh apple-banana pie	First the term "pie" is matched at position 3, then at position 2 "banana" is matched, next, NOT(apple) is matched with "fresh" at position 1. In this example, "apple" will be skipped.
apple NOT(banana) pie	no results	First the term "pie" is matched at position 3, then at position 2 we will first find "banana", but that does not match the query NOT(banana). Therefor, no results are returned.

Note 1: Since you may not know upfront if a combination of terms is hyphenated, it is advised to try different combinations of search queries when you suspect a hyphenated term might be part of the results.

Note 2: If a term or combination of terms you are searching for contains a hyphen, that term will often be found, even if you did not include a hyphen in your search query. For example, when you search for "email" or "e mail", it will also find "e-mail". However, "e mail" will not find "email" or the other way around ("email" will not find "e mail").

Keyword

A keyword is a *term* (page 113) used in a search query.

Occurrence

An occurrence is the number of times a given term occurs in the collection. An occurrence is defined by a

combination of Document id, Field id and Token id. Occurrences will be highlighted in the files.

Operators

Operators connect terms in a search query, making the search query more effective. Operators can be used to broaden or narrow your search. They can also be used to define your search more precisely.
For your personal clarity, operators are expressed in capitalized letters in this Guide.

Parentheses

Group words or phrases with round brackets when combining operators in your search query to show the order in which connections should be interpreted. For example, "(cow or goat) and (farm or dairy)". The queries placed between brackets will be processed first.

Brackets are not always required; they are mostly used for your own clarity. However, please be aware that using brackets can influence the outcome of a search. For example, searching for "cars or not used cars" will return different results than searching for "cars or not (used cars)". The first query will return "cars" and all words in front of "cars", except "used". The second query will only find the word "cars".

When using brackets, you do not need to leave a space between the operator/query and the first bracket, you can do both:

NOT(query)

NOT (query)

Period

A period (".") is treated like a separator when defined as such in the tokenizer/character map, except when:

- the period is preceded and followed by a number ("0.1" is one term)
- the period is preceded by a space and followed by a number (" .1" is one term)
- the period is preceded and followed by one alphabetic character, which can be repeated. ("A.B.C" is one term)
  If the last character is followed by a period, this last period will not be recognized as part of the term. This is because a period followed by a space is recognized as a separator.

Phrase

A phrase is identified as two or more words.

Precedence

When no brackets are used to define the order of precedence (see *Parentheses* (page 110)), the following search order is applied:

1. NOT
2. OR
3. W/n, P/n (these operators are of equal precedence)
4. AND
5. TO

Quotes

Quotes are used to search for separators. Examples:

"and"

"http://localhost/?id=10"

Regular Expression (or Search Expression)

A regular expression (abbreviated as regex) is a subset of a Search Expression. A regular expression is a sequence of characters that forms a (codified) search pattern. This pattern is used to find what we want.

Reserved Characters

All printable ASCII characters are directly searchable, except for those designated as reserved for a special purpose within the Search Engine.

The following characters are reserved:

?	single character wildcard
*	multiple character wildcard
+	used to match the preceding element one or more times
,	used as a decimal separator in different operators

.	A period (".") is treated like a separator when defined as such in the tokenizer/character map, except when: • the period is preceded and followed by a number ("0.1" is one term) • the period is preceded by a space and followed by a number (" .1" is one term) • the period is preceded and followed by one alphabetic character, which can be repeated. ("A.B.C" is one term) If the last character is followed by a period, this last period will not be recognized as part of the term. This is because a period followed by a space is recognized as a separator.
:	numeric range operator
()	used to nest sub-expressions in a search expression
[]	used for character class specification
{}	used in search statements with field definitions and quorum searches, and used in regular expressions
<>	used for numeric and file date comparisons
=	used for numeric comparisons, and for searching by file name and file date
/	used in proximity range operator searches
-	used in defining a range in a character class, and in proximity range searches to indicate negative values

You can use ? to search on reserved characters, but only if the reserved character is not configured as a *separator* (page 113). For example, query "t??st" will retrieve "t++st".

Search Query

A search query consists of one or more terms or keywords. Terms/keywords can be enhanced with Term Operators (Fuzzy/Wild Cards) and connected with Boolean or Proximity Operators. When using Boolean or Proximity Operators in your search query, group terms or phrases with round brackets to show the search order in which connections should be interpreted. For more information, see *Parentheses* (page **110**).

Separator

A separator is used by the tokenizer to mark the beginning of a document (BOD), the end of a sentence (EOS), end of a paragraph (EOP), end of a line (EOL), end of a page (EOG) or the end of a document (EOD).

The ZyLAB *tokenizer* (page **114**) can be configured to have some characters behave as separators. For example, ".", "(", ")", etc. These separators act as boundaries between tokens. Once a character has been recognized by the tokenizer as a separator, the tokenizer will stop processing the current token, the separator will be removed and the tokenizer will continue with processing the next token.

- You cannot search for a character if that has been configured as a separator.
- You can search for operators like EOS and EOD.
- Separators do not have token ids.

The following separators are supported:

BOD	Not supported
EOS	Supported, but disabled by default
EOP	supported
EOL	supported
EOG	supported
EOD	supported

Tip: When searching for "EOD", the query returns all documents with nothing highlighted. Since each document has an EOD token, it is an easy query to find all documents in a data set.

Term

A term is a type of query, the word query. It is also a unique entry in the dictionary. A term can be a character, a word (for example, sandwich) or a number. A term has a separator on either side. When, for example, brackets are set to be separators, the following text will be identified as 3 terms: "most definite(ly)".

Token

A token is often a term (word, number or separator), but a token can also be anything between two separators. Tokens are the identified small parts that form or define a file.

Token id

A token id is the natural number or position of a token, given by the tokenizer. Token ids are used to determine the distance between the words/numbers. Separators do not have token ids.

Token	There	are	5	files	EOS	EOD
Position	1	2	3	4	x	x

Tokenizer

A tokenizer breaks a stream of text up into words, numbers or other meaningful elements called tokens.

Wildcards

Wildcards are used to replace or represent one or more characters in a term, making the search query more flexible and efficient.

Word

A word is identified as one or more characters.

Word Query

A word query consists of one term.