



# Nuix Discover 10.21

Search  
Query  
Syntax



# Search Query Syntax

## Nuix Discover - Search technology and query syntax

Nuix Discover<sup>®</sup> employs the following two search technologies:

- **Content searches:** Nuix Discover uses dtSearch to index and search the content of files loaded to Nuix Discover (that is, *Document Content*).
- **SQL searches:** Nuix Discover uses SQL to search data loaded to fields (hereinafter referred to as *fielded data*).

*Advanced searches* constructed on the Search page can simultaneously search both Document Content and fielded data by using multiple search expressions connected by Boolean operators (that is, AND, OR, NOT).

For clarity, additional explanation is helpful.

### Content searches

When you load documents to Nuix Discover either through the Ingestions or Imports features, or through a single file upload, the application stores files associated with those document records on file systems that the application manages, scans, and monitors.

For example, a document record with **Document ID ABC-000001** can point to a Microsoft Word file named **ABC-000001.docx**. When an indexing and enrichment job is executed, Nuix Discover scans the file system and records any file whose name matches the pattern **[Document ID].[extension]** in the database. These files are referred to as *content* files. When Nuix Discover creates the document content index, dtSearch extracts text from the content files and indexes that content for searching.

---

**Note:** dtSearch does not save extracted text as a separate file. In some instances, users load documents to Nuix Discover and include both a native file and an extracted text file. For instance, if both ABC-000001.DOCX and ABC-000001.TXT are present, Nuix Discover indexes both files for content searching.

---

### SQL searches

Nuix Discover enables searching of *fielded data* (for example, email subject, file path) with SQL. Nuix Discover field types include yes/no (that is, Boolean), dates, numbers, pick lists, text (up to 256 characters) and memos. Of these field types, pick list, text, and memo fields contain words that users search.

Behind the scenes, Nuix Discover leverages SQL LIKE and SQL CONTAINS methods to search these field types. SQL LIKE is a search method in which a string of characters can be searched within a field. When a SQL LIKE search is executed, Nuix Discover does a real time scan of field values to find matches. Because this search uses a real time scan, SQL LIKE searches can take longer.

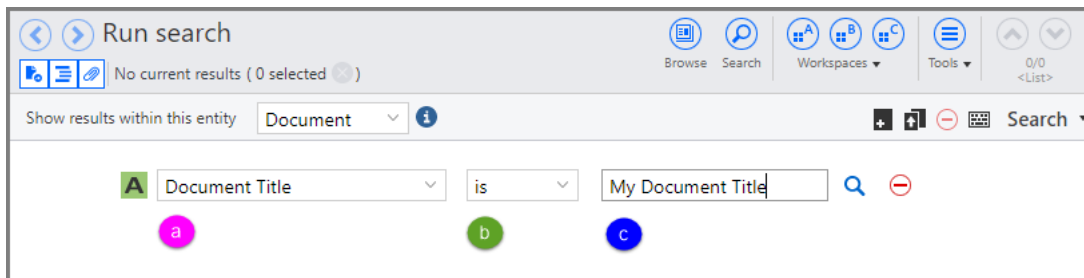
In contrast, SQL CONTAINS searches leverage a SQL full text index to find matches. Because SQL CONTAINS searches leverage an index, SQL CONTAINS searches are often faster and permit the use of limited wild cards as described below.

## Advanced searches

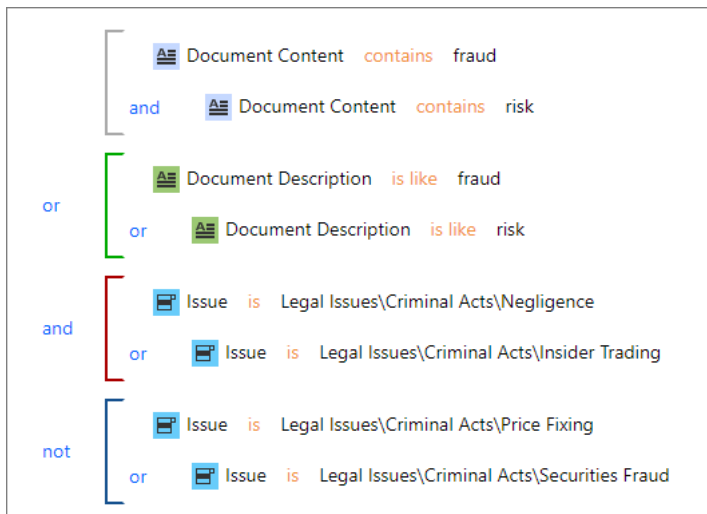
Advanced searches constructed on the Search page are comprised of *search expressions*.

For example, Document Title is My Document Title is an advanced search expression.

1. Document Title is the field (data).
2. “is” is the search expression operator.
3. My Document Title is the search term.



You can use multiple expressions along with Boolean operators such as AND, OR, and NOT. You can also group or nest search expressions to create more sophisticated filters.



For information about how to construct an advanced search, see “Perform an advanced search” in the Nuix Discover help.

The following sections provide more detail regarding searching in Nuix Discover.

- Search operators on the advanced Search page
- Search query syntax for Document Content
- Regular expressions
- Search query error messages and warnings

## Search operators on the advanced Search page

The following table describes the search operators used to construct search expressions in an advanced search on the Search page.

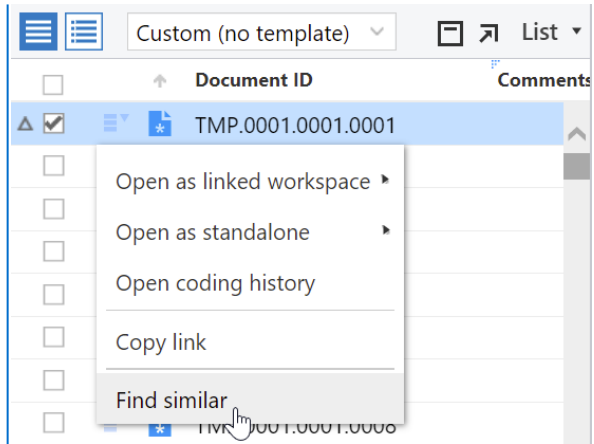
**Note:** The information in the following table does not apply to **Document Content contains** and **Document Content does not contain**, which use dtSearch syntax detailed in later sections. The People fields designation applies to People, Identities, and Organizations. Most search expressions apply to a single field (for example, Document Title), although some expressions apply to specific features in Nuix Discover (for example, History or Levels).

Operator	Applies To	Results
has a value	All Fields	Returns documents that are coded.
does not have a value	All Fields	Returns documents that are not coded.
is	<ul style="list-style-type: none"><li>• Pick Lists</li><li>• Text</li><li>• Number</li><li>• Yes/No</li><li>• People</li></ul>	Returns documents coded with the search term. One-to-many fields might be coded with additional values.
is not	<ul style="list-style-type: none"><li>• Pick Lists</li><li>• Text</li><li>• Number</li><li>• Yes/No</li><li>• People</li></ul>	Returns documents not coded with the search term. The result set includes documents that are not coded to any value in addition to documents coded with other values.
is only	<ul style="list-style-type: none"><li>• Pick Lists</li><li>• People</li></ul>	Returns documents that are only coded with the search term.
is any of	<ul style="list-style-type: none"><li>• Pick Lists</li><li>• People</li></ul>	Returns documents that are coded with the search term. This operator is used to create multiple search expressions that are grouped together separated by the OR operator.

greater than	<ul style="list-style-type: none"> <li>● Pick Lists</li> <li>● Text</li> <li>● Number</li> <li>● People</li> </ul>	Returns documents coded with a value that is alphabetically or numerically greater than the search term.
greater than or equal to	<ul style="list-style-type: none"> <li>● Pick Lists</li> <li>● Text</li> <li>● Number</li> <li>● People</li> </ul>	Returns documents coded with a value that is alphabetically or numerically greater than or equal to the search term.
less than	<ul style="list-style-type: none"> <li>● Pick Lists</li> <li>● Text</li> <li>● Number</li> <li>● People</li> </ul>	Returns documents coded with a value that is alphabetically or numerically less than the search term.
less than or equal to	<ul style="list-style-type: none"> <li>● Pick Lists</li> <li>● Text</li> <li>● Number</li> <li>● People</li> </ul>	Returns documents coded with a value that is alphabetically or numerically less than or equal to the search term.
is like	<ul style="list-style-type: none"> <li>● Pick Lists</li> <li>● Text</li> <li>● Memo</li> <li>● People</li> </ul>	<p>Returns documents where the search term is found anywhere in the field value.</p> <p>A search for <i>Document Title is like pl</i> returns documents with <i>apple</i> in the title.</p>
is not like	<ul style="list-style-type: none"> <li>● Pick Lists</li> <li>● Text</li> <li>● Memo</li> <li>● People</li> </ul>	Returns documents where the search term is not found anywhere in the field values. Also returns documents not coded with any value and documents coded with other values.
contains	<ul style="list-style-type: none"> <li>● Pick Lists</li> </ul>	Returns documents that satisfy the search expression.

	<ul style="list-style-type: none"> <li>• Text</li> <li>• Memo</li> <li>• People</li> </ul>	<p>Permissible syntax includes:</p> <ul style="list-style-type: none"> <li>• Single terms</li> <li>• Single terms with *. This will find values with the word stem.</li> <li>• Phrases (use double quotes around phrases)</li> <li>• Phrases with *. Use double quotes around phrases. Each term in the phrase is considered a stem. For example, "local wine*" will find "locally wined and dined".</li> <li>• AND, AND NOT, and OR are supported. The AND operator is evaluated before the OR operator. For example, Term A AND NOT Term B OR Term C is logically evaluated as (Term A AND NOT Term B) OR Term C.</li> <li>• AND NOT is not supported as the first part of a CONTAINS search.</li> </ul>
does not contain	<ul style="list-style-type: none"> <li>• Pick Lists</li> <li>• Text</li> <li>• Memo</li> <li>• People</li> </ul>	Returns documents that do not satisfy the search expression including documents that are not coded with any value. See permissible syntax in the previous row.
begins with	<ul style="list-style-type: none"> <li>• Pick Lists</li> <li>• Text</li> <li>• People</li> </ul>	<p>Returns documents where the search term is found at the beginning of the field value.</p> <p>A search for <i>Document Title begins with apple</i> returns documents with <i>Apple Pie</i> in the title but does not return documents with <i>Red Delicious Apple</i> in the title.</p>
does not begin with	<ul style="list-style-type: none"> <li>• Pick Lists</li> <li>• Text</li> <li>• People</li> </ul>	Returns documents where the search term is not found at the beginning of the field value, including documents that are not coded with any value and documents coded with other values.
is between	<ul style="list-style-type: none"> <li>• Pick Lists</li> <li>• Text</li> <li>• Number</li> </ul>	<p>Returns documents coded with a value that is between the search terms inclusive.</p> <ul style="list-style-type: none"> <li>• For text fields and pick list fields, returns documents with a value alphabetically</li> </ul>

	<ul style="list-style-type: none"> <li>• Date</li> </ul>	<p>between the two search terms, inclusive of the search terms.</p> <ul style="list-style-type: none"> <li>• For number fields, returns documents with a value numerically between the two search numbers inclusive.</li> <li>• For date fields, returns documents with a value chronologically between the two search dates inclusive.</li> </ul>
not between	<ul style="list-style-type: none"> <li>• Pick Lists</li> <li>• Text</li> <li>• Number</li> <li>• Date</li> </ul>	<p>Returns documents coded with a value that is not between the search terms inclusive.</p> <p>This includes documents that are coded with values outside the range, and documents that are not coded with any value.</p>
on	<ul style="list-style-type: none"> <li>• Date</li> </ul>	<p>Returns documents coded with the exact search date. One-to-many fields might be coded with other values.</p>
not on	<ul style="list-style-type: none"> <li>• Date</li> </ul>	<p>Returns documents not coded with the exact search date. This includes documents that are coded with no value and documents that are coded with other values.</p>
after	<ul style="list-style-type: none"> <li>• Date</li> </ul>	<p>Returns documents coded with a date that is later than the search date.</p>
on or after	<ul style="list-style-type: none"> <li>• Date</li> </ul>	<p>Returns documents coded with a date that is later than or equal to the search date.</p>
before	<ul style="list-style-type: none"> <li>• Date</li> </ul>	<p>Returns documents coded with a date that is earlier than the search date.</p>
on or before	<ul style="list-style-type: none"> <li>• Date</li> </ul>	<p>Returns documents coded with a date that is earlier than or equal to the search date.</p>
find similar to	<ul style="list-style-type: none"> <li>• Document ID</li> </ul>	<p>Returns documents that are conceptually similar.</p> <p>This type of search returns the same results that you receive if you select a document in the List pane, and then select Find similar on the Options menu.</p>

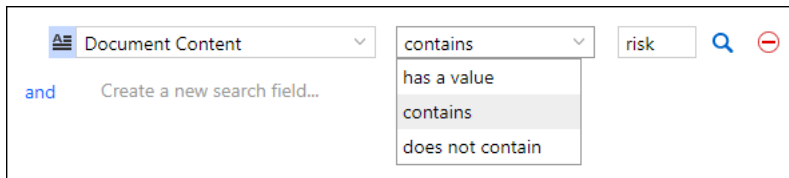


is correspondence between	<ul style="list-style-type: none"> <li>• People</li> </ul>	<p>Requires the selection of two people values. Returns documents where one of the people values is an author (FROM) and one of the people is a recipient (TO, CC, BCC). The order of selected names does not impact the result.</p>
is like correspondence between	<ul style="list-style-type: none"> <li>• People</li> </ul>	<p>Requires the input of two people search terms. Returns documents where one of the search terms matches an author (FROM) and the other search term matches a recipient (TO, CC, BCC). The order of search terms does not impact the result.</p> <p>The LIKE search syntax applies such that the search term can appear anywhere in the people value.</p>
contains correspondence between	<ul style="list-style-type: none"> <li>• People</li> </ul>	<p>Requires the input of two people search terms. Returns documents where one of the search terms matches an author (FROM) and the other search term matches a recipient (TO, CC, BCC). The order of search terms does not impact the result.</p> <p>The CONTAINS search syntax applies. See the <i>contains</i> entry earlier in this table.</p>
Previous Value Ever Was	<ul style="list-style-type: none"> <li>• Pick Lists</li> <li>• Text</li> <li>• Number</li> <li>• Yes/No</li> <li>• Date Fields</li> </ul>	<p>Returns documents where the search term matches how a document was previously coded in the searched field.</p>

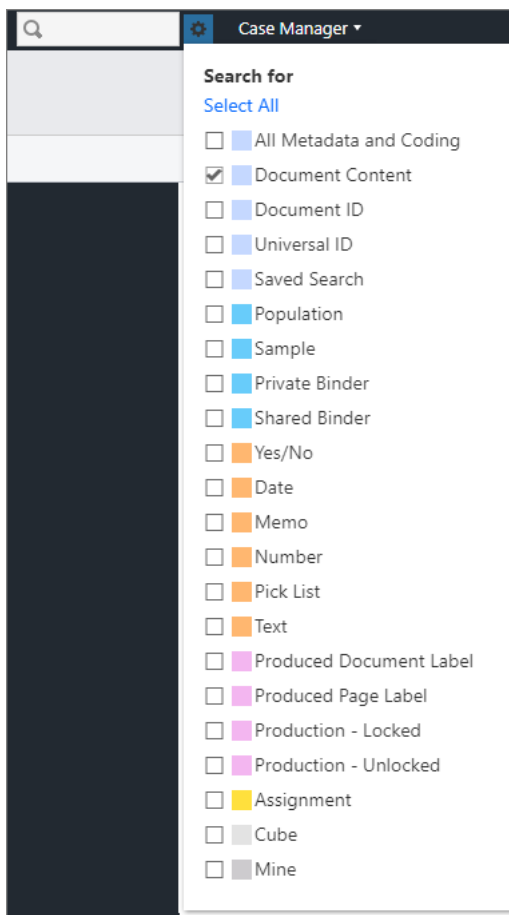
# Search query syntax for Document Content

Document Content is text that is extracted and indexed by dtSearch.

One method to search Document Content is to build an advanced search on the **Search** page. The available expression operators for Document Content are **contains**, **does not contain**, and **has a value**. This search method uses dtSearch syntax, which provides significant additional options, as described in the following table.



A second method to search document content is to use the quick search box on the navigation bar. The quick search box includes the option to search Document Content.



The syntax in the following table applies to those Document Content searches. For a full explanation of the functionality in the quick search box, see “Perform a quick search” in the Nuix Discover help.

## Document Content search operators

The following table describes the syntax and operators that you can use to query Document Content. Nuix Discover supports Unicode characters.

**Note:** Because dtSearch does not index punctuation, punctuation is not searchable unless your administrator changes the alphabet file. The same applies to the following characters because they are reserved as operators: %, #, &, ?, =, and ~

Operator	Description	Example query and results
AND	Requires all terms connected by AND to exist.	A search for <i>apple pie AND poached pear</i> finds documents that contain both <i>apple pie</i> and <i>poached pear</i> .
OR	Requires any term connected by OR to exist.	A search for <i>apple OR pear</i> finds documents that contain the word <i>apple</i> , documents that contain the word <i>pear</i> , and documents that contain both words.
NOT	Requires certain terms to not exist.	<ul style="list-style-type: none"><li>• A search for <i>NOT banana</i> finds documents that do not contain the word <i>banana</i>.</li><li>• A search for <i>apple AND NOT pear</i> finds documents that contain the word <i>apple</i> and do not contain the word <i>pear</i>.</li></ul>
not w/0	Searches for a word or phrase not in association with another word or phrase.	<ul style="list-style-type: none"><li>• A search for <i>Word?? not w/0 Word04</i> finds documents that include <i>Word01</i> or <i>Word02</i> or <i>Word03</i> but exclude documents that include <i>Word04</i>.</li><li>• Nuix Discover finds all words in the index that meet the criteria for <i>Word??</i> and excludes words after the <i>not w/0</i> proximity operator (in this case, <i>Word04</i>).</li></ul>
NEAR	The word <i>near</i> is treated as a search term, not an operator. Use a proximity search to locate terms that are near each other.	Not applicable.

Words and phrases	<ul style="list-style-type: none"> <li>• Quotation marks are not required when searching a phrase.</li> <li>• Noise words, such as <i>if</i> and <i>the</i>, are treated as any word.</li> <li>• To search for an exact phrase that includes the words <i>and</i>, <i>or</i>, or <i>not</i>, enclose the phrase in quotation marks.</li> <li>• Punctuation inside a search word is treated as a space.</li> </ul>	<ul style="list-style-type: none"> <li>• A search for tart apple pie finds tart apple pie but not apple pie.</li> <li>• A search for bill of sale finds documents containing bill, any intervening word, and sale.</li> <li>• A search for "apple and pie" in quotation marks finds documents containing apple and pie but not apple pie.</li> </ul>
( )	Use parentheses with searches that have two or more connectors. If you do not use parentheses, dtSearch evaluates OR operators before AND operators.	A search for <i>apple</i> AND ( <i>pear</i> OR <i>orange</i> ) finds the word <i>apple</i> with either <i>pear</i> or <i>orange</i> . If you do not use parentheses, this search will return the same result.
?	Wildcard that matches any single character.	A search for <i>appl?</i> finds <i>apple</i> and <i>apply</i> .
*	Wildcard that matches any number of characters. Use at the beginning or end of a search term.	A search for <i>*ppl*</i> finds <i>application</i> and <i>supply</i> .
=	Wildcard that matches a single digit.  Use multiple equals signs (=) to find multiple digits.	A search for <i>====</i> finds <i>1234</i> .
~	Stemming search: Finds grammatical variations of a word.	A search for <i>click~</i> finds <i>clicked</i> and <i>clicking</i> .
%	Fuzzy search is useful to find misspelled words or to search faulty text generated by optical character recognition (OCR). <ul style="list-style-type: none"> <li>• Each percent sign (%) in a search term represents one incorrect character.</li> <li>• Characters prior to percent signs must match exactly.</li> </ul>	<ul style="list-style-type: none"> <li>• A search for <i>capit%al</i> finds <i>capital</i>, <i>capitol</i>, and <i>capita</i>.</li> <li>• A search for <i>int%%ernet</i> finds <i>internet</i> and <i>intranet</i>.</li> </ul>
x w/n y	Proximity search: <ul style="list-style-type: none"> <li>• In a content search, <i>x w/n y</i> finds the term (<i>x</i>) within a specified number of words (<i>n</i>) of another term (<i>y</i>).</li> </ul>	<ul style="list-style-type: none"> <li>• A search for <i>apple w/5 pear</i> finds <i>apple</i> and <i>pear</i> where <i>apple</i> appears within five words of <i>pear</i>.</li> </ul>

	<ul style="list-style-type: none"> <li>• In a coding search (database search), <i>w/n</i> is treated as a proximity search when the value for <i>n</i> is 50 words or fewer. When the value for <i>n</i> is greater than 50 words, the proximity search is treated as an AND operator (both words exist in the text).</li> <li>• At least one of the two expressions connected by <i>w/n</i> must be a word, a phrase, or a group of words and phrases connected by OR.</li> <li>• The <i>x NOT w/n y</i> operator allows you to search for a term that is not associated with another term.</li> </ul>	<ul style="list-style-type: none"> <li>• A search for <i>(apple and banana) w/5 (pear or cherry)</i> finds documents that contain both <i>apple</i> and <i>banana</i> within five words of either <i>pear</i> or <i>cherry</i>.</li> <li>• A search for <i>apple NOT w/5 pear</i> finds <i>apple</i>, except where <i>apple</i> is within five words of <i>pear</i>.</li> </ul>
<i>x pre/n y</i>	<p>Proximity search:</p> <ul style="list-style-type: none"> <li>• Finds the term (<i>x</i>) within a specified number of words (<i>n</i>) of another term (<i>y</i>). The first term must occur before the second term.</li> <li>• At least one of the two expressions connected by <i>pre/n</i> must be a word, a phrase, or a group of words and phrases connected by OR.</li> </ul>	<ul style="list-style-type: none"> <li>• A search for <i>apple pre/5 pear</i> finds documents that contain <i>apple</i> within five words before <i>pear</i>.</li> <li>• A search for <i>(apple and banana) pre/5 (pear or cherry)</i> finds documents that contain both <i>apple</i> and <i>banana</i> within five words before either <i>pear</i> or <i>cherry</i>.</li> </ul>
<i>xfirstword</i> <i>xlastword</i>	<p>Built-in search words that indicate the beginning or end of a document, as follows:</p> <ul style="list-style-type: none"> <li>• <b><i>xfirstword</i></b>: Marks the beginning of a document.</li> <li>• <b><i>xlastword</i></b>: Marks the end of a document.</li> <li>• Combine <i>xfirstword</i> and <i>xlastword</i> with proximity operators to limit a search to the beginning or end of a document.</li> </ul>	<p>A search for <i>apple w/5 xfirstword</i> finds <i>apple</i> when it appears within five words of the beginning of a document.</p>

# Regular expressions

Regular expressions are an advanced syntax that allow you to search for a pattern of characters. For example, you can use regular expressions to locate Social Security numbers, credit card numbers, or other text that has a consistent pattern of characters.

To use a regular expression, you must enclose the regular expression in quotation marks and start the expression with `##`. For example: `"##[abc]"`

The following table describes the regular expressions that you can use in search boxes.

Regular expression	Description	Example query and results
<code>\d</code>	Represents one number in the pattern.	A search for <code>"##\d"</code> finds <i>1</i> and <i>2</i> .
<code>\w</code>	Represents one alphanumeric character or underscore in the pattern.	A search for <code>"##\w"</code> finds <i>a</i> and <i>1</i> .
<code>*</code>	Indicates zero or more numbers, alphanumeric characters, or underscores in the pattern.	A search for <code>"##click\w*"</code> finds <i>click</i> , <i>clicked</i> , and <i>clicking</i> .
<code>+</code>	Indicates one or more numbers, alphanumeric characters, or underscores in the pattern.	A search for <code>"##click\w+"</code> finds <i>clicked</i> and <i>clicking</i> , but not <i>click</i> .
<code>.</code> (period)	Represents any alphanumeric character or symbol.	A search for <code>"##appl."</code> finds <i>apple</i> and <i>appl2</i> .
<code>[abc]</code> <code>[123]</code>	Indicates a set of characters, one of which must be present. The order of the characters does not matter.	A search for <code>"##d[uo]g"</code> finds <i>dug</i> and <i>dog</i> , but not <i>dig</i> .
<code>[a-z]</code> <code>[0-9]</code>	Indicates a range of characters.	<ul style="list-style-type: none"><li>• A search for <code>"##[a-z]"</code> matches any lowercase character.</li><li>• A search for <code>"##ma[a-e]e"</code> finds <i>made</i>, but not <i>make</i>.</li></ul>
<code>[^abc]</code> <code>[^123]</code> <code>[^a-z]</code> <code>[^0-9]</code>	Indicates any character except the characters in the set or range.	A search for <code>"##c[^u]t"</code> finds <i>cat</i> , but not <i>cut</i> .

(abc xyz)	Indicates that either group of characters appears.	A search for "##require(d ment)" finds <i>required</i> and <i>requirement</i> .
?	Indicates that the preceding character appears zero or one times. Enclose the preceding character in parentheses.	A search for "##colo(u)?r" finds <i>color</i> and <i>colour</i> .
{n}	Indicates that the preceding expression occurs exactly n times.	<ul style="list-style-type: none"> <li>• To search for a pattern such as <i>ABC1234</i>, which consists of three letters followed by four digits, search for "##[a-z]{3}[0-9]{4}".</li> <li>• To search for a pattern such as <i>123-45-6789</i>, which consists of three digits, a hyphen, two digits, a hyphen, and four digits, search for "##\d{3}-\d{2}-\d{4}".</li> <li>• To search for multiple characters in a character set, use the <i>[abc]</i> expression followed by the <i>{n}</i> expression. For example, a search for "##the[resm]{2}" finds words that start with <i>the</i> and contain two of the characters in the set <i>[resm]</i>, such as <i>there</i> and <i>these</i>, but not words with one character in the set, such as <i>them</i>, or words with three characters in the set, such as <i>themes</i>.</li> </ul>
{n,}	Indicates that the preceding expression occurs a minimum of n times.	To search for a pattern that consists of the letters <i>ABC</i> followed by at least three digits, search for "##abc[0-9]{3,}". This search finds <i>ABC123</i> , <i>ABC1234</i> , <i>ABC12345</i> , and so on.
{n,m}	<ul style="list-style-type: none"> <li>• Indicates that the preceding expression occurs a minimum</li> </ul>	<ul style="list-style-type: none"> <li>• To search for numbers that are at least one digit and at most three digits, search for "##\d{1,3}".</li> </ul>

	<p>of <math>n</math> times, and a maximum of <math>m</math> times.</p> <ul style="list-style-type: none"> <li>To search for multiple characters in a character set, use the <code>[abc]</code> expression followed by the <code>{n,m}</code> expression.</li> </ul>	<p>This search finds 8, 12, and 374, but not 2001.</p> <ul style="list-style-type: none"> <li>A search for <code>###the[mres]{1,2}</code> finds words that start with <i>the</i> and contain one or two characters in the set <code>[mres]</code>, such as <i>them</i> and <i>there</i>, but not words with three characters in the set, such as <i>themes</i>.</li> </ul>
--	---	--

## Regular expression examples

When creating the syntax for regular expressions, keep the following in mind:

- To use a regular expression, you must enclose the regular expression in straight double quotation marks and start the expression with `##`. For example: `##[abc]`.
- Do not use curly quotation marks.
- To ensure that the syntax works as intended, your administrator must configure and index all characters and symbols in the syntax as letters, including the symbols in the **Index as letter** column in the following table.
- If you are an administrator, see the following section after the table in this topic: [For administrators: Configure characters and symbols in the index as letters](#).
- Use all lowercase letters in your syntax. By default, the application configures the index to be case insensitive unless an administrator changes the indexing options to case sensitive.
- Spaces are not searchable characters.

The following table provides examples of search types, the syntax to use, and sample results.

Search type	Syntax	Index as letter (administrator)	Sample results
United States Social Security Number (SSN) with hyphens	<code>##((?!000)(?!666)([0-8]\d{2}))\-(?!00)\d{2}\-(?!0000)\d{4}</code>	- (Hyphen; ASCII Code 45)	777-77-7777
SSN without spaces	<code>##\d{9}</code>		777777777
SSN with spaces	<code>##\d{3} ##\d{2} ##\d{4}</code>		777 77 7777

Document ID	"###doc\d{8}"		Doc00000001 or DOC00000001
Email	"##([\w_\.]+)@([\w_\.]+)\.([\w_\.]{2,6})"	@ (At symbol; ASCII Code 64)  . (Period symbol; ASCII Code 46)	jdoe@domain.com jane.doe@domain.com johnsmith@document.gov.edu
Date	"##[0-9]{1,2}/[0-9]{1,2}/[0-9]{2,4}"	/ (Forward slash; ASCII Code 47)	08/28/1963 8/28/1963 08/28/63 28/08/1963
Phone number  (United States format)	"##([0-9]{3}-)?[0-9]{3}-[0-9]{4}"	- (Hyphen; ASCII Code 45)	867-5309 800-649-2568
Visa	"##4\d{12,15}"		4321069823745 4321069823745123
Mastercard	"##5[1-5]\d{14}"		5490876543456782 5290876543456781
Discover	"##6011\d{12}" OR "##65\d{14}"		6011987623543109 6589043928435612
American Express	"##3[4,7]\d{13}"		340928374625172 371928172040238

For administrators: Configure characters and symbols in the index as letters

Only system administrators can configure the indexing options for a portal on the Portal Management > Settings > Indexing pages (Indexing: Options, Indexing: Alpha Standard, Indexing: Alpha Extended).

Portal administrators can edit the indexing options for a specific case on the Portal Management > Cases and Servers > Cases > [Case name] > Indexing pages (Indexing: Options, Indexing: Alpha Standard, Indexing: Alpha Extended).

---

**Note:** Case indexing options override portal indexing options.

---

To allow users to use the syntax described in the previous table, administrators must index all characters and symbols in the syntax as letters, including those described in the following table.

Symbol	Description	ASCII Code
-	Hyphen	45
@	At symbol	64
.	Period	46
/	Forward slash	47

---

**Important:** Any changes to the indexing options require you to delete and rebuild the index. After that, you must submit a new Indexing and enrichment job on the Portal Management > Processing > Jobs page. Depending on the size of the index, this can take time.

---

## Search query error messages and warnings

If a search query contains errors, the application displays an error or warning message. The following table describes search query error and warning messages, and provides examples of search queries that trigger the messages.

Message text	Example search query that triggers this message
Warning: term contains a leading wildcard. This term may be too broad and may not run successfully.  <b>Note:</b> It is best to not start a search with a wildcard.	*apple
Error: one side of the proximity operator must be a single word, phrase, or series of words and/or phrases separated by <b>or</b> .	(apple AND banana) w/5 (pear AND apricot)
Error: parentheses are not properly balanced.	(apple AND (banana OR pear)

Error: phonic operator # cannot be mixed with a stemming or synonym operator.	#apple~
Error: phonic operator # must be the first character of the word.	apple#
Error: quotes are not properly balanced.	"banana AND "apple pear"
Error: stemming operator ~ can only be placed at the end of the word.	abc~d
Error: synonym operator & must be placed at the end of the word.	abc&d
Error: term contains two or more consecutive boolean operators.	apple AND OR banana
Error: term or clause begins with an operator.	AND apple
Error: term or clause ends with an operator.	apple OR banana OR
Error: the negation operator must be in either the form <b>and not</b> or <b>not w/#</b> .	apple NOT banana
Error: the proximity operator cannot be preceded by <b>and</b> or.	apple AND w/2 banana
Error: wildcard operators cannot be used with the # (phonic), ~ (stemming), or & (synonym) operators.	#apple*
Warning: the # character is the phonic search operator. # is not searchable as a character.	apple#
Warning: the % character is the fuzzy search operator. % is not searchable as a character.	100%
Warning: the & character is the synonym search operator. & is not searchable as a character.	abc&d
Warning: term contains !. The wildcard operator character is *.	apple!
Warning: term contains ambiguous boolean operators.	apple AND banana OR pear

Warning: term contains ambiguous proximity operators.	apple w/2 banana w/5 pear
Warning: term contains noise word(s) <b>[comma delimited list of noise words]</b> . Search results will include documents that contain any word in this position.	apple on the banana
Warning: the proximity operator must be in the form <b>w/#</b> or <b>pre/#</b> .	apple /5 banana
Warning: the term is missing an operator.	(apple) (banana)
Warning: term contains a regular expression. This term may be too broad and may not run successfully.	"##[a-z][a-z][a-z]"