Advanced Search Syntax 10.3

White Paper

October 2018

Laserfiche®

Table of Contents

Introduction 3
Search Symbols 3
Operator Compatibility 4
<> Operator 4
Operator Precedence 4
Wildcards 5
Date, Time, and Date/Time Formats 6
Filtering by Entry Type 7
Attribute Search 8
Root Folder Name 8
Search String Limitations 9
Types of Searches 10
Searching by Properties 10
Text Searches on Document Text 11
Stop Words 11
Single Word Search 11
Multi-Word Search 12
Phrase Searches 12
Text Searches on Entry Properties 12
The ~= Operator 13
Escape Characters 14
Document Text Search Examples 15
AND and OR Text Searches 15
Searching for Words that are Close Together 16
Property Searches 20
Resolving Shortcuts 20
Entry Name Searches 20
Template/Field Searches 22
Search within Folder 33

Search Within Volume 35
Volume ID Search 35
Annotation Text Searches 35
Annotation Properties Searches 37
Electronic Document Searches 45
Date and Time Searches 47
Page Searches 48
Indexed and Indexable Searches 50
Tag Searches 51
Entry Creator Search 52
Entry Owner Search 52
Last Modified By Search 53
Entry ID Searches 53
Document Links Searches 54
Version Searches 56
Shortcut Searches 57
Checked Out Documents Searches 58
UUID Search 59
Digital Signature Searches 59
Classification Level Searches 60
Business Process Searches 60
Records Management Searches 61
Entry Access Rights Searches 68
Geography Searches 72
Using Advanced Search Syntax with Custom Quick Searches 75

Introduction

Laserfiche's advanced search syntax allows you to carry out searches that may not be possible using the other advanced search types available in the Windows client or web client. If you want to create custom quick searches for your users, you will also need to understand advanced search syntax. This paper is a collection of advanced search syntax strings consolidated into one location to give users a reference and make it easier for them to develop their own search strings. You will find explanations and examples of different types of advanced searches you might want to perform while working in Laserfiche.

Note: The Laserfiche client search and the web client use the same advanced search syntax.

Search Symbols

The following symbols function as operators in advanced search syntax:

=	equal to	<>	not equal to
<=	less than or equal to	>=	greater than or equal to
<	less than	>	greater than
&	and	|	or
-	not	^N	within N number of words

Since these symbols are operators in the search syntax, you have to be careful when any of these symbols are part of the phrase you are looking for. Special rules may determine whether and how you can include search symbols as part of the text you are searching for—see the section on document text searches for more details. Full-text searches ignore punctuation but do not ignore search symbols. All other searches never ignore punctuation or search symbols.

Example: To look for the phrase, "The mayor - John Brown," in a full-text search, you must enclose the phrase in double quotation marks. Searching "The mayor - John Brown," will return the following results:

The mayor - John Brown

The mayor! John Brown

The mayor, John Brown

The mayor John Brown

Laserfiche will look for results with the phrase in quotation marks and will return any symbol or no symbol in place of the – (dash).

If you do not include the phrase in quotation marks, the dash will be interpreted as the "not" operator and will exclude results with the word "John" in them.

To include a " in your search, you must precede the character with another double quotation mark (") and surround the entire phrase with double quotation marks. For example, to search for a document named a"report, use this syntax: **{LF:NAME= "a""report"}**.

Operator Compatibility

Whether a certain operator can be used in a search depends on the type of value being searched for. The following are some situations where certain operators cannot be used.

- Many of our search examples search for entities with empty values ("") or with any non-empty value ("*"). The ~= operator does not work for such searches; only = or <> should be used.
- To search across text or list fields, the search parameter must be a string and the operator must be ~=, = or <>, unless a field name is specified. If a field name is specified, you can use ordering operators (<, >, >=, <=). When such operators are applied to text fields or list fields, the order used is alphanumeric order, i.e. a < b, numbers < letters, and punctuation < numbers.

<> Operator

The <> operator is not a set operator. For example, **{LF:tags<>"tag1"}** looks for documents with a tag that is not tag1. It doesn't look for documents that don't contain tag1. If a document has both tag1 and tag2, the search will return this document because tag2 is not tag1.

Operator Precedence

Advanced search syntax is evaluated from left to right. However, when the search syntax has more than one operator, certain operators are always evaluated before other operators. The order of precedence, from highest precedence to lowest, is as follows:

- Parentheses **()**
- Within **^**

- Not –
- And **&**
- Or |

Because parentheses have the highest precedence, enclosing a portion of the search syntax in parentheses ensures that section of the syntax will always be evaluated first.

Wildcards

Wildcards can be used in both full-text and property searches. For field searches, wildcards can only be used to search fields that are a text or list type.

- **Asterisk (*)**. The asterisk wildcard represents any number of missing characters or no missing characters. For instance, a search on the term **report*** would find "report," "reports," "reporting," "reported," "reporter," etc.
- **Question mark (?)**. The question mark wildcard represents exactly one character. For instance, if you were unsure whether a name was spelled "Anderson" or "Andersen," you could search on the term **Anders?n**. This would return both results.
- **Brackets ([])**. Like a question mark, a set of brackets represents a single missing character. However, brackets can be used to specify a smaller range of options. For instance, searching for **d[io]ve** would find the words "dive" and "dove," but not "Dave."

  > **Note**: Oracle does not support brackets. If your repository uses Oracle for its database management system, you can use brackets for full text searches (on document text or entry properties) but not for other search types.

- **Dash (-)**. The dash character is used in conjunction with brackets to specify that only characters within a particular range should be found. For instance, you might want to locate all documents that contain an account number that begins with "100347" and ends with a number rather than a letter. You could search on **100347[0-9]**. This would return documents containing account number "1003475", but not those containing "100347C."

Wildcards can be used in combination. For instance, if you wanted to return all documents with reference to writing or written materials, you could search on the term **wr[io]t***. This could locate any of the following terms: "write," "written," "writing," or "wrote."

Note: Searches that use strings starting with a wildcard (e.g. ***term**) are significantly *slower* than searches that use strings ending with a wildcard (e.g. **term***).

Escaping Wildcards

To include the characters [,], ?, \, * in your search, instead of using them as wildcards, you must precede them with a backslash (\) and include the phrase in double quotation marks. For example, to search for a document named *, use this syntax: **{LF:NAME="*"}**. To search for these characters in a text search without using the symbol as a wildcard, the character must be escaped by putting a \ (back-slash) before the special character, in addition to being enclosed in double quotation marks (e.g., "Is it me\?").

Date, Time, and Date/Time Formats

The YMD (year, month, day) date format with a four-digit year is always recognized no matter what the defined date format is. For example, **`{LF:Created="2006/07/09"}`** will always work regardless of the client's regional settings. To specify years before the year 1000, prepend 0s to the year. For example, a day in the year 16 should be specified in the format **0016/*mm*/*dd***.

When you specify a date in your search string that is not in the YMD format with a four-digit year, Laserfiche will assume that it is in the MDY (month, day, year) format unless you insert a header specifying otherwise. For example, if I enter the search string **{LF:Created<="20/04/2018"}**, the client will return an error indicating that the date is out of range, because there is no 20th month in a year. To specify a different date format, such as DMY, insert a header in your search string indicating the order of month/day/year that you want to use. The header should have the format **%*???*** In which *???* can be M, D, or Y, in any order.

Example: To look for entries created before April 20, 2018, I can prepend a header to the previously mentioned search string, as follows: **%DMY {LF:Created<="20/04/2018"}**. This tells the Laserfiche search engine to interpret the date provided as being in the day/month/year format.

You can separate the month, day and year with different delimiters. Both slashes (/) and dashes (-) are allowed, as are both two-digit years and four-digit years. For example, if you are using the default MDY order, you can use the formats ***mm*/*dd*/*yy***, ***mm*/*dd*/*yyyy***, and ***mm*-*dd*-*yy***.

If the year is a two-digit number less than 60, the Laserfiche will assume that the year is in the 21st century. For example, 59 will be treated as 2059. If the two-digit

year is greater than or equal to 60, the Laserfiche will assume that the year is in the 20th century. For example, 60 will be treated as 1960.

Some date searches will also accommodate date/time searches. Instead of a date input, you can specify a time after the date, with a space separating the two.

Possible time formats include:

- 02:30:35
- 2:30:45
- 02.30.45

The following restrictions apply to time specifications:

- To specify AM, type in any characters beginning with a, such as a, a.m., or am.
- To specify PM, type in any string of characters beginning with p, such as p, p.m., or pm.
- If AM or PM is not specified, Laserfiche assumes the time is on the 24-hour system.
- There must be a space between the numbers and your specification of AM or PM.

Filtering by Entry Type

Including an entry type in your search will help create a more efficient search. You can specify that the results only contain specific types of entries when doing **LF:Name** and **LF:ChildName** searches. Use the following letter codes to specify the types of entries to search for:

- **F**: All folders
- **O**: Folders excluding record series and record folders
- **E**: Record series
- **R**: Record folders
- **B**: Documents without assigned templates
- **D**: Documents, whether they have templates or not
- **S**: Shortcuts

Example: {LF:Name="*", Type="S"} will return all shortcuts in the repository, and no other types of entries.

Attribute Search

Similarly to how you can specify a type of entry to search for, you can specify additional attributes that the property you are looking for should have. This is particularly useful when you are searching for entries that could have multiple properties (e.g. annotations or entry access rights) of the type you are looking for, and you want multiple conditions to apply to only one property. For example, if you are searching for an annotation that has a red color and contains the word "important", you should use an attribute search as follows:

{LFAnn:Color="255", Text~="important"}

Using the **&** operator will not necessarily produce the same results. Suppose you attempt to reproduce the above search using the **&** operator as follows:

{LFAnn:Color="255"} & {LFAnn:Text~="important"}

This search will return not just entries with red annotations containing the text "important", but also any entries that have both at least one red annotation and at least one annotation containing "important". This happens because an entry can have multiple annotations, and it is possible for an entry to meet both conditions in the conjunction without those conditions applying to the same annotation.

The attribute search is also important for stamp searches, because different private stamps can share the same name while having different owners. To differentiate between these stamps in your search, you can include the **owner** parameter in an attribute search.

Root Folder Name

The root folder appears in search results under the name ROOT FOLDER to indicate the identity of the folder. Searches relating to the root folder behave as follows:

- The root folder generally does not behave as though its name is ROOT FOLDER. For example, searching for "ROOT" will not return the root folder.
- Searching for "*" returns the root folder.
- The following are exceptions where the root folder does behave as though its name is ROOT FOLDER:
    - If you search for <>"ROOT FOLDER", the root folder is not included in the results.

- For the operators >, <, >=, and <=, the root folder behaves as though it is named ROOT FOLDER.

Search String Limitations

An advanced search string cannot be longer than 65536 Unicode characters including spaces. Also, you cannot have more than 500 Boolean operators in your advanced search string.

Types of Searches

Searching by Properties

Full-text searches on document text search for words or phrases in the indexed contents of a repository. In addition to full-text searches, you can also search for entries that have specific properties. For example, you may want to search for entries that were created within a certain date range, or that have a name beginning with a certain letter. All search types that are not full-text searches must be enclosed in curly brackets {}. If you want to find entries that have all of a set of properties, you can combine search criteria using this syntax: **{*Criteria1, Criteria2, Criteria3, ...*}**. This kind of syntax acts similarly to the **&** operator.

> **Example**: To look up entries that are in the Default volume with the word "statement" in their name, use one of the following syntaxes: **{LF:VolName="Default", LF:Name="*statement*"}** or **{Lf:Name="*statement*"} & {LF:Volname="DEFAULT"}**.
>
> The asterisks mean that results will be returned on any entry with the word "statement" in its title that is preceded or succeeded by any string. Combining terms within curly brackets implies the use of an ampersand (**&**).

You can also use a comma to separate values when searching for fields in a template or to specify a property's options. As examples of the latter, you can specify a type of entry (document, folder, shortcut, or batch) when searching for an entry by name, or specify an owner when searching for a stamp. However, if you use commas to search for a set of properties, you cannot also use commas to define property options (such as type or owner) in the same search.

Property searches that do not use the ~= operator lack some of the features that property searches on indexed document properties using the ~= operator have. Searches using the ~= operator can take advantage of fuzzy search, the Within search, logical operators between search phrases, sub-queries, word stemming, and ranking of search results. We elaborate on this latter type of text search on properties later.

> **Note**: Except for field searches, property searches that search for only the wildcard (="*") will include results with empty strings. <>"*" behaves the same as ="*".

Text Searches on Document Text

To search for phrases in document text using advanced search syntax, enter the search string without any curly brackets. For example, entering ***word1* & *word2*** as your advanced search syntax string will return all documents that have both *word1* and *word2* in their generated text.

Stop Words

Stop words are commonplace words, such as **and**, **a**, and **the**, that would bring up many irrelevant search results if included in a search term. These words are ignored in searches unless either of the following conditions hold:

- They are included in double quotation marks.
- The search is a single word search.

Example: If you search for the phrase **The Unnamable** without using quotation marks, the Laserfiche search engine will ignore the **The**, and the search results will be equivalent to those you obtain from simply searching for the single word **Unnamable**. If you want the stop word **The** to be included, use double quotes around your search phrase: **"The Unnamable"**.

Single Word Search

Searching for a single word is done without the use of double quotation marks. Stop words are *not* ignored for single word searches. Most delimiter characters are not allowed in single word searches. The only delimiter characters allowed are the following:

- Wildcard characters
- Periods (.)
- Single quotes or apostrophes (')
- @
- !

Note: If the word contains an ampersand (&), the word must be contained in quotation marks, otherwise the search feature will read the symbol as the AND operator.

Example: Searching for the term **Johnson&Johnson** requires the use of quotation marks (i.e., **"Johnson&Johnson"**).

Multi-Word Search

A multi-word search is a full-text search that consists of multiple words and no double quotes. This type of search returns the entries that contain all the words from the search. It functions similarly to a search that uses the **&** operator. Entries in which the words are located closer to one another are ranked higher.

Stop words are ignored for multi-word searches.

You can use logical operators like &, |, -, and ^ with multi-word searches. Use parentheses to separate the terms linked by the logical operators.

Phrase Searches

If you are searching for an exact phrase, enclose the phrase in double quotation marks. This will return results that match the phrase exactly and will include stop words. If you do not enclose the phrase in double quotation marks, Laserfiche will automatically perform a multi-word search: It will insert an **&** between each word and ignore stop words.

> **Example**: A search for **"quarter surplus"** will return entries with the word "quarter" followed by the word "surplus" with only white space or punctuation between them. A search for **quarter surplus** (without double quotation marks) will search for documents that have the words "quarter" and "surplus" in them, where those words are not necessarily adjacent to each other. Phrase searches will not ignore stop words.

In a phrase search, you can include any character in the search phrase. Delimiter characters are ignored. Logical operators are treated as delimiters and also ignored.

Text Searches on Entry Properties

Unlike text searches on generated document text, text searches on entry properties search for text in indexed properties like annotations, names, and comments. Since they search on properties, they should always be enclosed in curly brackets—unlike text searches on document text. However, unlike property searches, they can take advantage of advanced text search features like fuzzy search and stemming.

The following metadata and entry properties are automatically indexed:

- Entry names
- Extensions

- Version comments
- Link group comments
- Digital signature comments
- Digital signature certificate subject
- Tag comments
- Annotations, including their text (in callouts, sticky notes, and textboxes) and comments

Fields may or may not be indexed. An administrator can determine whether an individual field is indexed when they create or modify the field.

The ~= Operator

To take advantage of full-text search features when doing text searches on entry properties, use the ~= operator. If you use the = operator, you can use wildcards, but not features like fuzzy search, the Within search, logical operators between search phrases, sub-queries, stemming, and ranking of search results. The following examples illustrate the difference between the ~= and = operators.

- **{[General]:[Category] = "my category"}** will return entries with the General template and the Category field where the value of the latter is exactly "my category".
- **{[General]:[Category] ~= "my category"}** is a multi-word text search on Category fields that are part of the template General. This search will work only if the field Category is indexed. It will return entries that have both words "my" and "category" in a Category field that is in a General template (including results that are within the limits set by your fuzzy search settings and results matching related words determined by word stemming).
- To search for phrases using the ~= operator, enclose the phrase in single quotes within the double quotes. **{[General]:[Category] ~= "'my category'"}** returns entries with the General template and the Category field where the value of the latter contains the phrase "my category" (including results that are within the limits set by your fuzzy search settings results matching related words determined by word stemming). This search will work only if the field Category is indexed.

  Note: When you use the ~= operator, dashes (-) in a search term are interpreted as the NOT operator unless the search term is enclosed in single quotes. For example, to search for a value containing the string "plant-based" in the (indexed) Diet field, use the following query: **{[]:[Diet] ~= "'plant-based'"}**.

- **{[General]:[Category] ~= "(classified | unclassified) ^2 record"}** is a full-text search matching any entry that has the words "classified" or "unclassified" within 2 words of the word "record" in the General template's Category field. It uses logical operators between search terms, which is not possible in a property search with other operators. The Category field must be indexed for this search to work.

Note: Unlike the **<>** and **=** operators, the **~=** operator does not support searching for empty (“”) or non-empty (“*”) values. For such searches, you should use the **<>** or **=** operator, even for indexed properties.

Escape Characters

As with regular searches, when using the ~= operator, the following characters:

**[ ] ? **

are interpreted as part of search syntax unless they are preceded by a \. The ~= operator introduces two more characters that need escaping. A single quote (‘) will be interpreted as part of search syntax unless it is preceded by another single quote. For example, to search for the value “O’Leary” in a field, the query should be written as follows:

{[]:[] ~= “O’’Leary”}

where two single quotes, not a double quote, follow the **O**.

The tilde (~) also needs to be escaped when using the ~= operator. You can escape a ~ by adding a \ before it. For example, to search for the value “~42” in a field, the query should be written as follows:

{[]:[] ~= “\~42”}

Document Text Search Examples

The examples in this section are searches for words or phrases in document text. For text searches on indexed document properties (using the ~= operator), see the examples in Property Searches.

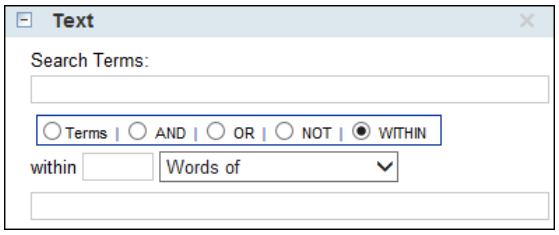
AND and OR Text Searches

To Search For…	Syntax	Example
two terms in the same document • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	***term1* & *term2*** *term1* and *term2* are the words or phrases you want to search for.	**document & imaging** This search will return the following result because it includes both search terms. **"Document imaging helps organizations run smarter."** The search terms do not have to be consecutive. The search will also return the following result. **"Imaging a document allows you to automatically extract text from it."** This search will not return the following result because it is missing one of the search terms. **"Importing a document into Laserfiche is easy."**

To Search For…	Syntax	Example
either term in the same document • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	***term1*** **|** ***term2*** *term1* and *term2* are the words or phrases you want to search for.	**document | images** This search will return the following results because they include at least one of the search terms. **"Document management should be at the heart of every business."** **"Images are stored as TIFF files."** **"Document images help businesses become paperless."** This search will not return the following result because it does not include at least one of the search terms. **"How accessible are your records?"**

Searching for Words that are Close Together

Documents that contain two specific words and/or phrases that are located near each other can be found using the ^ operator. The two search terms must be separated by at most a specific number of words. In the web and Windows clients, you can carry out this kind of text search as a WITHIN text search.



In our examples below, *N* is an integer representing how far apart the two terms are, counted in terms of number of words. Two consecutive words are considered to be 1 word apart. Two words that have a word between them are considered to be 2 words apart, and so on.

The definition of a "word" depends on the language. Generally, a word consists of consecutive non-delimiter characters that are:

1. Prefixed by either a delimiter or the start of a stream, and

2. Followed by a delimiter or the end of a stream.

Delimiter characters consist of all characters that are not alphanumeric, except for the following four: # $ % &.

> **Note**: In Chinese, Korean, and Japanese, each non-delimiter character is considered to be a separate word.

To Search For…	Syntax	Example
words and/or phrases located at most a specific number of words apart • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**(*term1* ^*N* *term2*)** *term1* and *term2* are the words or phrases you want to search for. This search returns results where *term1* and *term2* are at most *N* words apart.	**(document ^2 imag*)** This search will return the following results because each contains the specified search terms within 2 words of each other. • **"Document imaging helps organizations run smarter."** • **"Importing a document or image into Laserfiche is easy."**
two terms that are at most a certain number of words apart and where the first term in the query appears *before* the second term in the query • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**(*term1* ^+*N* *term2*)** Returns results where *term1* appears at most *N* words before *term2*.	**(ham ^+3 sandwich)** Documents with the following text will be returned: • "**ham sandwich**" • "**ham and cheese sandwich**" Documents with the following text will not be returned: • **"To make this sandwich, slice some ham."**

To Search For…	Syntax	Example
two terms that are at most a certain number of words apart and where the first term in the query appears *after* the second term in the query • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**(*term1* ^-*N* *term2*)** Returns results where *term1* appears at most *N* words after *term2*.	**(chips ^-3 "sea salt")** Documents with the following text will be returned: • **"sea salt and vinegar chips"** • **"sea salt chips"** Documents with the following text will not be returned: • **"sprinkle your chips with sea salt"**
two terms that both exist in a document, but which are *not* located within N words of each other • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**(*term1* -^*N* *term2*)** Returns results where *term1* and *term2* both exist in the document, but are not within *N* words of each other. This search is not sensitive to the order in which *term1* and *term2* appear.	**(chocolate -^2 icing)** Documents with the following text will be returned: • **"Chocolate cake with buttercream icing"** Documents with the following text will not be returned: • **"chocolate icing"** • **"icing for chocolate cake"** Any documents that do not contain both the words "chocolate" and "icing" will be omitted.
two terms that both exist in a document, but where the first term in the query does not appear N (or fewer) words *before* the second term. • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**(*term1* -^+*N* *term2*)** Returns results where *term1* and *term2* both exist in the returned document, but *term1* does not appear *N* (or fewer) words before *term2*.	**(chocolate -^2 "buttercream icing")** Documents with the following text will be returned: • **"buttercream icing on chocolate cake"** • **"make a chocolate cake with buttercream icing"** Documents with the following text will not be returned: • **"chocolate buttercream icing"**

To Search For…	Syntax	Example
two terms that both exist in a document, but where the first term in the query does not appear N (or fewer) words *after* the second term. • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**(*term1* -^-*N* *term2*)** Returns results where *term1* and *term2* both exist in the returned document, but *term1* does not appear *N* (or fewer) words after *term2*.	**(technology -^-1 information)** Documents with the following text will be returned: • **"click here for information about this mobile technology"** • **"with this technology, information becomes more accessible"** Documents with the following text will not be returned: • **"information technology department"**

Property Searches

The searches in this section look for entries based on properties such as metadata and annotations. They include both indexed and non-indexed searches.

Resolving Shortcuts

Certain searches allow you to specify the **ResolveShortcuts** option.

Use "y", "yes", "1", or "true" as the value for **ResolveShortcuts** if you want shortcuts to inherit the properties of their parent documents for the purposes of the search. Use "n", "no", "0", or "false" if you do not want shortcuts to inherit their parents' properties for this search. The value specified for **ResolveShortcuts** overrides any setting for resolving shortcuts in the client's options dialog.

If you do not specify the **ResolveShortcuts** parameter, or if **ResolveShortcuts** is set to "default", then whether the search resolves shortcuts depends on the settings in your client's options dialog.

> **Note**: If the **ResolveShortcuts** option is enabled and used in multiple searches linked by logical operators, the shortcuts will be resolved before the search elements are linked by logical operators.

Entry Name Searches

To Search For…	Syntax	Example
an entry by name (i.e., folder, document) • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:Name="*name*", Type="*type*", ResolveShortcuts="*y*/*n*"}** *name* is the name of the entry you want to search for. *type* is the type of entry being searched for. If you do not specify an entry type, the search will only search for *documents* satisfying the specified entry name conditions.	To search for a folder (F) named General Documents: **{LF:Name="General Documents", Type="F"}** To search for documents whose name contains "General", including shortcuts to those documents: **{LF:Name~="General", ResolveShortcuts="yes"}**

To Search For…	Syntax	Example
an entry by the name of its child entry (i.e., a document or folder within a particular folder) • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:ChildName="*name*"}** *name* is the child entry name.	To locate a file, with only the knowledge that it has a child document named District 5: **{LF:ChildName= "District 5", Type="D"}** **Note:** A child entry name search will only return an entry's immediate parent. **Note:** If you do not specify an entry type, the search will only search for child documents.
an entry by the name of its parent entry (i.e., a folder that contains a particular document or folder) • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:ParentName="*name*"}** *name* is the parent entry name.	I want to see all entries contained in the folder named Patient Billing, but I cannot remember the file path to the folder. To search for the entries using this parent entry name: **{LF:ParentName= "Patient Billing"}** **Note:** The parent entry name search will only return an entry's immediate children in the results.

Template/Field Searches

Template and field names must be enclosed in square brackets if the name contains a space or a parenthesis. Field values that contain text, dates, or date/times must be enclosed in quotation marks.

To Search For…	Syntax	Example
an entry not assigned to a template • Can be performed using the clients' search interfaces.	**{LF:TemplateID=0}**	To search for entries not assigned a template: **{LF:TemplateID=0}**
entries assigned a specific template • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:TemplateName= "*name*"}** *name* is the name of the template. **Note**: This search does not allow you to look for documents that do not have templates. Use **{LF:TemplateID=0}** to find documents without templates.	To search for entries assigned to the General template: **{LF:TemplateName= "General"}** To search for entries assigned any template: **{LF:TemplateName="*"}**
entries that have been assigned any template • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:TemplateID<>0}** **Note**: The <> operator with this search cannot return documents without templates.	To search for entries that have been assigned any template: **{LF:TemplateID<>0}**

To Search For…	Syntax	Example
entries by template field values • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{[*name*]:[*fieldname*]= "*value*"}** *name* is the name of the template. *fieldname* is the name of the field. *value* is the value of the field.	To search for entries assigned to the Library Documents template with Shakespeare as the author: **{[Library Documents]:[Author]= "Shakespeare"}** To search for multiple field values, use commas. **{[Library Documents]:[Author]="Shakespeare", [Title]="Romeo and Juliet"}** This will return all documents assigned the following: • Template: Library Documents • Author: Shakespeare • Title: Romeo and Juliet.
entries by field value only • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*]="*value*"}** *fieldname* is the name of the field. *value* is the value of the field.	To search for entries where the Author field is exactly "Shakespeare": **{[]:[Author]="Shakespeare"}** To search for entries where the Author field is indexed and contains "me": **{[]:[Author]~="me"}**

To Search For…	Syntax	Example
entries with a field that is empty • Can be performed using the clients' search interfaces.	**{[]:[*fieldname*]=""}** *fieldname* is the name of the field. There is nothing inside the quotation marks.	To search for entries with an empty Author field: **{[]:[Author]="" }** **Note:** This search will return all entries that have the Author field but have no value in that field.
entries with a field that is not empty • Can be performed using the clients' search interfaces.	**{[]:[*fieldname*]="*"}** *fieldname* is the name of the field.	To search for entries with any value in the Author field (entries with empty values in the Author field are not returned): **{[]:[*fieldname*]="*"}** To search for entries with an Author field that is non-empty and not equal to "Tiptree": **{[]:[Author]<>"Tiptree" }**
entries with a field value within a specified range • Can be performed using the web client's search interface. **Note**: You can also search for a date range using this syntax, as long as the dates are properly formatted and enclosed in double quotes.	**{[]:[*fieldname*]>=*integer1*, <=*integer2*}** *fieldname* is the name of the field. *integer1* is the beginning of the range. *integer2* is the end of the range.	To search for entries where the field Seminar Size has values between 20 and 30: **{[]:[Seminar Size]>=20, <=30}** This search will return all entries with a Seminar Size field value equal to 20 or 30 or between 20 and 30.

To Search For…	Syntax	Example
entries with a time field within a range that includes midnight • Can only be performed using advanced search syntax. **Note**: You can use the search interfaces to search for a time range that does not include midnight.	**{[]:[*fieldname*] between "*time1*" and "*time2*"}** *fieldname* is the name of the field. *time1* is the beginning of the time range. *time2* is the end of the time range. See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To search for a time field with a value between 5 PM and 2 AM: **{[]:[Time] between "5:00:00 PM" and "2:00:00 AM"}**
entries with multi-value fields • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*]="*value*"} | {[]:[*fieldname*]="*value*"}** *fieldname1* and *fieldname2* are the names of the fields. *value1* and *value2* are the values of the fields.	To search for entries with a multi-value Author field containing at least Shakespeare or Hamlet: **{[]:[author]~="Shakespeare"} | {[]:[author]~="Hamlet"}** This search will return all entries assigned the multi-value Author field containing at least one of Shakespeare or Hamlet. If this multi-value field also contains Caesar, the entry will still be returned. Since we use the ~= operator in this search, it means the matching field values only need to contain the words "Shakespeare" or "Hamlet", but do not need to be identical to "Shakespeare" or "Hamlet".

To Search For…	Syntax	Example
entries with specific values in multi-value fields • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*]=(“*value1*”, “*value2*”,”*value3*”)}** *fieldname* is the name of the field. *value1, value2,* and *value3* are the values of the field.	To search for entries with a multiple-value Author field containing the exact values Shakespeare and Hamlet *only*: **{[]:[author]=(“Shakespeare”,” Hamlet”)}** This search will return all entries containing only Shakespeare and Hamlet in the multi-value Author field. If this multi-value Author field also contains Caesar, the entry will not be returned. If it contains only Hamlet and not Shakespeare, it will not be returned. **Note:** If a document has a multi-value field containing multiple instances of a search value, such as “Shakespeare, Shakespeare, Hamlet, Hamlet, Hamlet,” that document will be returned. It must have at least one of each of the specified values, but no other values.

To Search For…	Syntax	Example
entries that do not contain specific values in a multi-value field • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*]<>(“*value1*”, “*value2*”,”*value3*”)}** *fieldname* is the name of the field. *value1, value2,* and *value3* are the values of the field.	To search for entries with a multi-value Author field that does not contain only Shakespeare and Hamlet: **{[]:[Author]<>(“Shakespeare”, ”Hamlet”)}** If a document contains Shakespeare and Hamlet as the only values in the Author field, it will not be returned. If it contains Shakespeare, Hamlet, and Caesar as values in the Author field, it will be returned. If the document has a value in the Author field that is not equal to Shakespeare or Hamlet, it will be returned. If it contains only Shakespeare or only Hamlet in the Author field, it will be returned.
entries with a specific field value in a particular position in a multi-value field on a particular template • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[*name*]:[*fieldname*] (*position)*=“*value*”}** *name* is the name of the template. *fieldname* is the name of the field. *position* is the position in the multi-value field (1 is the top-most value). *value* is the value of the field.	To search for entries assigned the Human Resources template with a multi-value Telephone Number field that contains 123-456-7890 in the 2nd position: **{[Human Resources]: [Telephone Number](2)= “123-456-7890”}** **Note:** You can also use the words **all** or **any** in place of *position*. **all** indicates that the condition must be satisfied by all values in the multi-value field. **any** indicates that the condition must be satisfied by at least one value in the field.

To Search For…	Syntax	Example
entries with a specific field value in a particular position in a multi-value field, regardless of template • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*](*position)= "value"}** *fieldname* is the name of the field. *position* is the position in the multi-value field (1 is the top-most value). *value* is the value of the field.	To search for entries with a value of 100 in the 1st position of the multi-value Test Score field: **{[]:[Test Score](1)=100]** To search for entries where any of the values in the multi-value Test Score field are 100: **{[]:[Test Score](any)=100]** To search for entries where all of the values in the multi-value Test Score field are greater than 80: **{[]:[Test Score](all)>80}**
entries with at least one multi-value field value that matches a value on the list defined in the search • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*](any) in ("value1", "value2", "value3")}** *fieldname* is the name of the field. *value1, value2,* and *value3* are the possible values of the field.	To search for entries with a multi-value Author field that contains at least one of the following values: Romeo, Juliet, Shakespeare: **{[]:[Author](any) in ("Romeo", "Juliet", "Shakespeare")}** If a document's Author field contains only Romeo or only Romeo and Juliet as values, it will be returned. If it contains Romeo and Hamlet, it will be also returned because at least one value, Romeo, is found in the list defined in the search. If the only values in the Author field are Hamlet and Iago, the document will not be returned.

To Search For…	Syntax	Example
entries that have multiple field values where none of the values are found in the list defined in the search • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*](all) not in (“*value1*”, “*value2*”, “*value3*”)}** *fieldname* is the name of the field. *value1, value2, and value3* are the values of the field.	To search for entries with a multi-value Author field that does not contain any of the following values–Romeo, Juliet, or Shakespeare: **{[]:[author](all) not in (“Romeo”, “Juliet”, “Shakespeare”)}**
entries with multiple field values that are all in the list defined in the search • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[*fieldname*](all) in (“*value1*”, “*value2*”, “*value3*”)}** *fieldname* is the name of the field. *value1, value2, and value3* are the values of the field.	To search for entries with a multi-value Author field that contains all or some of the values Romeo, Juliet, and Shakespeare, but no others: **{[]:[author](all) in (“Romeo”, “Juliet”, “Shakespeare”)}** If a document’s field contains only Romeo, or Romeo and Juliet, it will be returned. **Note:** Each of the field’s values must match at least one of the listed values in order for the entry to be returned.
entries with a specific field value anywhere in a particular template • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[*name*]:[]=“*value*”}** *name* is the name of the template. *value* is the value of the field.	To search for entries assigned the Human Resources template with the field value Peter in any of its fields: **{[Human Resources]:[]= “Peter”}** **Note:** The content of the search criteria determines what potential fields are searched. For example, if you search Peter (characters), it will search all character and list fields. If you search 99 (number), it will search all character, list, number, integer, and long integer fields.

To Search For…	Syntax	Example
entries with empty field values anywhere in a particular template • Can only be performed using advanced search syntax.	**{[*name*]:[]=“”}** *name* is the name of the template. There is nothing inside the quotation marks.	To search for entries assigned the Human Resources template with any empty field values: **{[Human Resources]:[]=“”}**
entries with no field values in a specific field • Can be performed using the clients' search interfaces.	**{[]:[*field*]=””}** *field* is the name of the field. There is nothing between the quotation marks.	To search for entries with a blank Author field: **{[]:[Author]=””}**
entries with any value in a specific field • Can only be performed using advanced search syntax.	**{[]:[*field*]=”*”}** *field* is the name of the field.	To search for entries with any value in the Author field: **{[]:[Author]=”*”}**
entries with a specific field value in any field, regardless of template • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{[]:[]=“*value*”}** *value* is the value of the field.	To search for entries assigned with the field value Peter in any of its fields, regardless of template: **{[]:[]=“Peter”}** To search for entries with non-empty fields: **{[]:[]=”*”}** To search for entries with non-empty indexed fields: **{[]:[]~=”*”}**

To Search For…	Syntax	Example
entries with empty field values in any field, regardless of template • Can only be performed using advanced search syntax.	**{[]:[]=“”}** *value* is the value of the field. There is nothing inside the quotation marks.	To search for entries with empty field values for any of its fields, regardless of template: **{[]:[]=“”}** To search for entries with empty indexed fields: **{[]:[]~=“”}** To search for entries with non-empty fields: **{[]:[]<>“”}**
entries in folders that satisfy certain template conditions. • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LF:ParentTemplateName=”*template*”}** Replace *template* with the name of the relevant template. This search does not return the parent entry in the results, unless that entry also has a parent that meets the template specified conditions.	To find all entries that are the immediate children of a folder with the General template: **{LF:ParentTemplateName=”General”}** To find all entries whose parent does not have the General template: **{LF:ParentTemplateName<>”General”}**

To Search For…	Syntax	Example
entries based on field values within multi-value field groups • Can only be performed using advanced search syntax. • Wildcards permitted.	**{[Template]:[*Field1*](r) = "*Value1*", [*Field2*](r)="*Value2*"}** Replace *Field1* and *Field2* with the fields that are in the same multi-value field group you are interested in. More fields in the same group may be appended. Replace *Value1* and *Value2* with the values you are looking for. For more complex searches where you want to specify conditions pertaining to more than one group, use **(r*N*)** to distinguish the different groups. In an example on the right, we look for documents where there is a group containing fields satisfying one set of conditions, *and* another (distinct) group containing fields satisfying another set of conditions. You can add even more conditions on additional groups by using **(r3)**, **(r4)**, and so on.	To search for entries with an individual field with a value "Invoice", and a multi-value field group that contains the multi-value fields Name and Phone, where Name starts with "J" and Phone starts with "(562)": **{[]:[]="Invoice", [Name](r)="J*", [Phone](r)="(562)*"}** To search for entries with a multi-value field group containing the multi-value fields Company Contact, Company/Entity, and Phone, where one field group has a Company Contact starting with "John" and a Company/Entity starting with "Chicago", while a second field group has a Company Contact starting with "Johanna" and a Phone equal to "(555) 111 1234": **{[Contract]:[Company Contact](r1) = "John*", [Company/Entity](r1)="Chicago*"} & ({[Contract]:[Company Contact](r2) = "Johanna*", [Phone](r2) = "(555) 111 1234"})**

Search within Folder

To Search For…	Syntax	Example
entries in a specific folder • Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:LookIn="*path*", Subfolders=0}** *path* is where the item you are looking for is located. **Note**: The path must be a valid folder path that starts with a backslash or contains a letter. If you enter a number in place of *path*, it will be interpreted as a folder ID. **0** means that no subfolders will be searched. **Note**: **N**, **no**, or **false** can also be used to mean that no subfolders will be searched.	To search the folder General Documents for an entry named Romeo and Juliet: **{LF:LookIn="General Demo\General Documents", Subfolders=0} & {LF:Name="Romeo and Juliet", Type="D"}** This will return the following document path in the repository: **General Demo\General Documents\Romeo and Juliet** Entries in subfolders will not appear in the above results.
entries in a folder and all its subfolders • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:LookIn="*path*", Subfolders=y}** *path* is where the item you are looking for is located. **y** indicates that the results should include subfolders. You can also use **yes**, **1**, or **true** instead of **y**. **Note**: If you do omit the Subfolders term, subfolders will be searched. **{LF:LookIn="*path*", Subfolders=y}** returns the same results as **{LF:LookIn="*path*"}**.	To search the General Documents folder and its subfolders for an entry named Romeo and Juliet: **{LF:LookIn="General Demo\General Documents", Subfolders=y} & {LF:Name="Romeo and Juliet", Type="D"}** This will return the following document path in the repository: **General Demo\General Documents\William Shakespeare\ Romeo and Juliet** To search for any document whose name contains "Romeo" and is a descendant of the folder New Folder under the root folder: **{LF:LookIn="New Folder"} & {LF:Name~="Romeo"}**

To Search For…	Syntax	Example
entries with a specific field value in certain folders • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{[*template*]:[*field*]="*value*"} & ({LF:LookIn= "*path1*"}|{LF:LookIn= "*path2*"})** *field* is the name of the field. *template* is the name of the template. *value* is the value of the field. *path1 and path2* are the folder paths.	To search for entries in the Sales or Marketing folders that have the Human Resources template and a Document Type field value equaling Resume: **{[Human Resources]:[Document Type]="Resume"} & ({LF:LookIn="Marketing"} | {LF:LookIn="Sales"})** To search for entries in the Recruiting folder that have a Document Type field value containing "Application": **{LF:LookIn="Recruiting"} & {[]:[Document Type]~="Application"}** **Note:** This search assumes that the Document Type field is indexed.
entries with a specific field value not in a certain folder • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{[*template*]:[*field*]="*value*"} - ({LF:LookIn= "*path*"}** *field* is the name of the field. *template* is the name of the template. *value* is the value of the field. *path* is the folder path.	To search for entries that are *not* located in the Public Relations folder (or its subfolders), but that have the template Human Resources, and where the Document Type field has the value Resume: **{[Human Resources]:[Document Type]="Resume"} - ({LF:LookIn="Public Relations"})**

Search Within Volume

To Search For…	Syntax	Example
entries in a volume by volume name • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:VolName="*name*"}** *name* is the name of the volume.	To search for entries assigned to a volume named Paris: **{LF:VolName="Paris"}** **Note:** If the volume is a logical volume, the search will return the documents in the corresponding physical volumes.

Volume ID Search

To Search For…	Syntax	Example
entries in a volume by volume ID number • This search can only be performed using advanced search syntax. • Not case-sensitive.	**{LF:VolID=*integer*}** *integer* is the entry ID number. This number must be greater than zero.	To search for entries assigned to a volume with volume ID 1: **{LF:VolID=1}** **Note:** If the volume is a logical volume, the search will only support the = and <> operators.

Annotation Text Searches

To Search For…	Syntax	Example
entries containing sticky notes • Can be performed using the web client search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:Sticky="*"}** * is a wildcard and will search for all text, resulting in a search for all non-empty sticky notes. **{LF:Sticky="*text*"}** *text* is the text on the sticky note that is being searched for. **{LF:Sticky=""}** There is nothing inside the quotation marks. This search will return all empty sticky notes.	To search for all entries with any sticky note text: **{LF:Sticky="*"}** To search for entries containing a sticky note that does not have the word "important": **{LF:Sticky ~= "-important"}**

To Search For…	Syntax	Example
entries containing text boxes • Can be performed using the web client search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:Textbox="*"}** * is a wildcard and will search for all text. **{LF: Textbox="*text*"}** *text* is the text in the text box that will be searched for. **{LF:Textbox=""}** There is nothing inside the quotation marks. This will return all empty text boxes.	To search for an entry with text box text stating "Read pages 1-30": **{LF:Textbox="Read pages 1-30"}** To search for an entry with a textbox containing the word "Read": **{LF:Textbox~="Read"}**
entries containing callout text boxes with any text • Can be performed using the web client search interface. • Not case-sensitive. • Wildcards permitted.	**{LF: Callout="*text*"}** *text* is the text in the callout that will be searched for. **{LF:Callout=""}** There is nothing inside the quotation marks. This will return all empty callouts.	To search for all entries with any text in the callout: **{LF:Callout="*"}** To search for an entry with text in the callout stating "Start here" exactly: **{LF:Callout="Start here"}** To search for an entry with text in the callout containing "edit": {**LF:Callout ~= "edit"}**
entries containing annotation text (the text in sticky notes, callouts, and textboxes) with the specified properties • Can be performed using the web client's search interface.	**{LF:anntext="*text*"}** **{LF:anntext~="*text*"}** **{LFAnn:Text="text"}** **{LFAnn:Text~="text"}** Replace *text* with your desired search term. **{LF:anntext="*"}** **{LFAnn:Text~="*"}** * is a wildcard and will search for any annotation that can contain text. **Note:** Callouts, textboxes, and sticky notes are the annotation types that can contain text. The above search will include empty callouts, textboxes, and sticky notes in the results.	To search for all entries with empty sticky notes, callouts, or textboxes: **{LF:anntext=""}** To search for all entries with annotation text that contains a word starting with "Important": **{LF:anntext~="Important*"}** To search for all entries with protected annotations containing text: **{LFAnn:Text="*", Protected="y"}** Read more about how the **LFAnn:** prefix lets you carry out attribute searches.

To Search For…	Syntax	Example
entries with annotations that are attachments, where the attachment contains certain text • Can be performed using the web client search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:attachment~="*text*"}** Replace *text* with your search term. The types of attachments that the text search will cover are the same as the types of electronic documents that are indexed in your repository.	To search for all entries with annotations that are attachments containing either of the words "imaging" or "scanning": **{LF:attachment~="scanning imaging"}**

Annotation Properties Searches

Some of the searches in this section contain an option with the **LF:** prefix and another with the **LFANN:** prefix. When both options are presented, the searches are identical except for the possibility of doing attribute searches with the **LFANN:** prefix. Using the **LFANN:** prefix, you can specify that the search return only entries where multiple conditions specified apply to one annotation, as long as each condition uses a search with the same prefix. For example, you can specify that the search return all documents with a yellow annotation on the first page with the query **{LFANN:Color="65535", Page=1**}. In contrast, if you specify **{LF:AnnColor="65535"} & {LFANN:Page=1}**, the search will also return any document that has a yellow annotation (on any page) and that also has an annotation (of any color) on the first page.

To Search For…	Syntax	Example
annotations by color • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:AnnColor="*color*"}** Alternatively, **{LFANN:Color = "*color*"}** *color* can be expressed in three different ways: The decimal value of the annotation color based on the RGB color model, (e.g., yellow = 65535; red = 255; blue = 16744448), the 3-character hexadecimal value (e.g. red=#f00, yellow=#ff0), or the 6-character hexadecimal value (e.g. red=##FF0000, black=#000000). All parts of an annotation with a color are searched. For example, for a callout, both line color and fill color are searched. If any color property of the annotation matches the search, a result is returned.	To search for entries with yellow annotations: **{LF:AnnColor="65535"}** To search for entries with black annotations: **{LFAnn:Color="#000"}** To search for entries with red annotations: **{LFAnn:Color="##FF0000"}**

To Search For…	Syntax	Example
annotations by creation date and time • Can only be performed using advanced search syntax. • Not case-sensitive.	Possible expressions: **{LF:AnnCreateDate= "*date time*"}** **{LFANN:CreateDate= "*date time*"}** **{LFANN:Created= "*date time*"}** *date* is the date the annotation was created. *time* is the time the annotation was created. **Note**: You can specify date and time or only date. See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To search for an annotation created on April 14, 2012: Possible date formats: **{LF:AnnCreateDate="04/14/2012"}** **{LF:AnnCreateDate="04/14/12"}** **{LF:AnnCreateDate ="04-14-2012"}** **{LF:AnnCreateDate ="04-14-12"}** To search for an annotation created between dates April 10 and April 15, 2012: **{LF:AnnCreateDate >="4/10/2012", AnnCreateDate<="04/15/2012"}** **Note**: If the year is a two-digit number less than 60, Laserfiche will assume that the year is in the 21st century. For example, 59 will be treated as 2059. If the two-digit year is greater than or equal to 60, the server will assume that the year is in the 20th century. For example, 60 will be treated as 1960.
annotations by creator • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:AnnCreator= "*creator*"}** Alternatively, **{LFANN:Creator= "*creator*"}** *creator* is the user who created the annotation. **Note:** The only valid format for searching for entries created by a Windows-authenticated user is: **"*domain\user*"** You can search for Windows-authenticated users across all domains with ***\user***.	To search for annotations created by Jeff: **{LF:AnnCreator="Jeff"}**

To Search For…	Syntax	Example
annotations by comments • Can be performed using the web client search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:AnnComment= "*comment*"}** Alternatively, **{LFANN:Comment = "*comment*"}** *comment* is the text added to the annotation in the **Metadata** tab of the annotation properties dialog box.	To search for annotations with "Please do not delete" as a comment: **{LF:AnnComment= "Please do not delete"}** To search for annotations with a comment containing a word starting with "dino": **{LFann:Comment~="dino*"}**
annotations by type • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LF:AnnType="*kind*"}** **or** **{LF:AnnType <> "*kind*"}** *kind* is the type of annotation. Possible values for *kind*: • attachment • bitmap • callout • freehand • highlight (or H) • line • polyline • rectangle • redaction (or R) • stamp (or S) • note (or N) • strikeout • textbox • underline	To search for all documents with highlights: **{LF:AnnType="H"}** To search for all documents with any annotation that is not a sticky note: **{LF:AnnType<>"note"}** **Note:** Documents with sticky notes will be included in the results if they also have another kind of annotation.

To Search For…	Syntax	Example
annotations that are or are not the specified annotation type • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LFANN:Type <> "*kind*"}** or **{LFANN:Type = "*kind*"}** This search syntax differs from **LF:AnnType** in that it is in the **LFANN** namespace, so it can be used for attribute searches. The possible values for *kind* are the same as those for **LF:AnnType**.	To search for all stamps that are on the first page of a document: **{LFANN:Type = "s", Page=1}** To search for all polylines created by the ADMIN user: **{LFANN:Type="polyline", Creator="ADMIN"}**
annotations by page • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:AnnPage=*page*}** Alternatively, **{LFANN:Page=*page*}** *page* is an integer indicating the page that the annotation is on.	To search for an annotation on page 100 of a document: **{LF:AnnPage="100"}** **Note**: Page numbers start at page 1. To search for an annotation on page 1 of a document containing the text "read": **{LFANN:Page=1, text~="read"}**
annotations that are attached files • Can only be performed using advanced search syntax.	**{LFANN:FileName="*document*"}** *document* is the name of the file that is attached as an annotation.	To search for annotations that are attached files with a name "form.png", and that have a comment containing "note": **{LFANN:FileName="form.png",Comment~="note"}**

To Search For…	Syntax	Example
annotations by ID • Can only be performed using advanced search syntax.	**{LFANN:ID = "*id*"}** Replace *id* with the annotation's ID. Annotation IDs are unique to each annotation in a repository, and are not reused if the annotation is deleted.	This search is designed for use in the Laserfiche SDK, not in the Laserfiche clients.
annotations by item ID • Can only be performed using advanced search syntax.	**{LFANN:ItemID = *id*}** Replace *id* with the annotation's item ID. Item IDs are unique to a document page. They may be reused if an annotation is deleted and another is added on the same page.	This search is designed for use in the Laserfiche SDK, not in the Laserfiche clients.
annotations based on date of last modification • Can only be performed using advanced search syntax.	**{LFANN:Modified = "*date*"}** The syntax follows that of the **LFANN:Created** search. =, <>, <, <=, >, >= operators are supported.	To find annotations modified after March 1, 2018: **{LFANN:Modified>"03/01/18"}**

To Search For…	Syntax	Example
annotations based on visibility setting • Can only be performed using advanced search syntax.	**{LFANN:Visibility=*integer*}** **Replace *integer* with one of the following:** • **0**: Visible to the entry creator and entry owner only. • **1**: Visible to all users with the See Annotations entry access right, or the See Through Redactions entry access right for redactions. • **2**: Visible to all users with Read access to the document, even if they lack the See Annotations or See Through Redactions access right.	To find all entries with annotations visible to the entry owners and creators only: **{LFANN:Visibility=0}** To find all entries that are visible to all users and that contain the text "emergency": **{LFANN:Visbility=2, Text ~= "emergency"}**
annotations based on Z-order (how an annotation is layered with respect to other annotations) • Can only be performed using advanced search syntax.	**{LFANN:ZOrder = *integer*}** When the first annotation is created on a page, it has a Z-order of 0. When there are multiple annotations on a page and the user brings one annotation to the front, the Z-order of that annotation is set to an integer that is one more than the highest Z-order on the page. When the user sends an annotation to the back, the Z-order of that annotation is set to an integer that is one less than the lowest Z-order on the page. Z-orders can be negative.	To find all annotations with a Z-order of 0: **{LFANN:ZOrder = 0}**

To Search For…	Syntax	Example
annotations based on whether the annotations are protected. • Can only be performed using advanced search syntax.	**{LFANN:Protected = "*y*"}** Use "y", "yes", "1", or "true" to search for protected annotations. Use "n", "no", "0", or "false" to search for unprotected annotations.	To search for unprotected annotations: **{LFANN:Protected = "n"}** To search for protected annotations on the first page of a document: **{LFANN:Protected="0", Page=1}**
annotations based on the ID of the annotation reason • Can only be performed using advanced search syntax.	**{LFANN:ReasonID = "*id*"}**	This search is designed for use in the Laserfiche SDK, not in the Laserfiche clients.
public/personal stamps by name • Can be performed using the web client search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:AnnStampName= "*name*"}** *name* is the name of the stamp. **Note:** The only valid format for searching for entries created by a Windows-authenticated user is: **"*domain\user*"**. You can search for Windows-authenticated users across all domains with ***\user.***	To search for a document with a stamp (public or personal) named "faxed": **{LF:AnnStampName= "faxed"}** You can restrict the search to a specific user's personal stamp. To search for the faxed stamp owned by Sue: **{LF: AnnStampName = "faxed", owner="Sue"}**
public stamps only by name • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:AnnPubStampName= "*name*"}** *name* is the name of the public stamp.	To search for a public stamp named "denied": **{LF:AnnPubStampName="denied"}**

Electronic Document Searches

To Search For…	Syntax	Example
entries that are a specific type of electronic document (i.e., Word, Excel, PDF, PowerPoint, etc.) • Can be performed using the clients' search interface, for some extensions. • Not case-sensitive. • Wildcards permitted.	**{LF:Ext="*ext*"}** *ext* is the extension of the electronic document. **Note:** You do not need to add a period (.) before the extension. Use **doc** instead of **.doc**.	To search for all entries that are Microsoft Word documents: **{LF:Ext="*docx*"}**
entries that are electronic documents • Can only be performed in the Windows client.	**{LF:Ext="*"}** ***** (asterisk) is a wildcard and will search for all extensions.	To search for all entries that are electronic documents: **{LF:Ext="*"}**
entries based on the MIME types of their electronic documents • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.Can only be performed using advanced search syntax. • Wildcards permitted.	**{LF:mimetype="*type*"}** Order-based operators (<, >, >=, <, <=) are supported, in addition to = and <>.	To search for Microsoft Excel electronic documents: **{LF:mimetype="application/vnd.ms-excel"}**
entries that are a specific size of electronic document • Can be performed through the clients' search interface. • Wildcards permitted.	**{LF:EDocSize=*integer*}** *integer* is the entry size in bytes.	To search for an electronic document that is 100 KB: **{LF:EDocSize=102,400}** **Note:** If you're not using the clients' search interface, you must specify the entry size in bytes. 1 KB = 1024 bytes.

To Search For…	Syntax	Example
entries that contain an image file satisfying certain size conditions • Can be performed using the clients' search interfaces.	**{LF:ImageSize=*integer}*** *integer* is the entry size in bytes.	To search for an image that is less than 35 KB: **{LF:ImageSize<35840}** **Note:** If you are not using the clients' search interfaces, then you must specify the entry size in bytes. 1 KB = 1024 bytes.
entries that contain a text file satisfying certain size conditions • Can be performed using the clients' search interfaces.	**{LF:TextSize=*integer}*** *integer* is the entry size in bytes.	To search for a text file larger than 35 KB: **{LF:ImageSize>35840}** **Note:** If you are not using the clients' search interfaces, then you must specify the entry size in bytes. 1 KB = 1024 bytes.
entries that contain a thumbnail satisfying certain size conditions • Can only be performed using advanced search syntax.	**{LF:ThumbSize=*integer}*** *integer* is the entry size in bytes.	To search for a thumbnail that is larger than 35 KB: **{LF:ThumbSize>35840}** **Note:** If you are not using the clients' search interfaces, then you must specify the entry size in bytes. 1 KB = 1024 bytes.
entries that contain a location data of a specific size • Can only be performed using advanced search syntax.	**{LF:LocDataSize=*integer}*** *integer* is the entry size in bytes.	To search for a location data file that is smaller than 35 KB: **{LF:ImageSize<35840}** **Note:** If you are not using the clients' search interfaces, then you must specify the entry size in bytes. 1 KB = 1024 bytes.

To Search For…	Syntax	Example
entries for which the total size of a specific item type satisfies certain conditions • Can only be performed using advanced search syntax.	**{LF:Total*item*Size=*integer}*** *item* is the object being searched for. Input one of the following: **text**, **thumb**, **image**, or **locdata**. *integer* is the entry size in bytes.	To search entries whose total image size is larger than 35 KB: **{LF:TotalImageSize>35840}** **Note:** If you are not using the clients' search interfaces, then you must specify the entry size in bytes. 1 KB = 1024 bytes.
electronic documents based on when they were last modified • Can only be performed using advanced search syntax.	**{LF:EdocModified="*date*"}** See the Date, Time, and Date/Time Formats section for more information on how to format dates. =, <>, <, <=, >, >= operators are supported.	To search for entries whose electronic documents were last modified before 2018: **{LF:EdocModified<"01/01/18" }**

Date and Time Searches

See Date, Time, and Date/Time Formats for more details on possible date formats.

To Search For…	Syntax	Example
entries by creation date and time • Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:Created="*date time*"}** *date* is the date the entry was created. *time* is the time the entry was created. See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To search for a document created on April 20, 2012 at 2:30:45 PM: **{LF:Created= "4/20/12 2:30:45 pm"}** To search for a document created between April 20 and April 25: **{LF:Created>="4/20/2012", Created<="04/25/2012"}**

To Search For…	Syntax	Example
entries by modification date and/or time • Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:Modified="*date time*"}** *date* is the date the entry was modified. *time* is the time the entry was modified. See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To search for a document modified on a certain date (April 22, 2012): **{LF:Modified ="04/22/2012"}** **{LF:Modified ="04/22/12"}** **{LF:Modified="04/22/2012"}** **{LF:Modified ="04-22-12"}** **{LF:Modified ="04-22-2012"}** **Note:** More than one date format is possible (see above).

Page Searches

A "page" is defined as an entry with associated text (generated by OCR) or images.

To Search For…	Syntax	Example
entries with/without pages • Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:AssociatedPages= "Y"}** **{LF:AssociatedPages= "N"}** "Y", "Yes", "1", or "true" mean the document has pages. "N", "No", "0", or "false" mean the document does not have pages.	To search for documents that do not have pages (associated text or images): **{LF:AssociatedPages= "N"}** To search for documents that have pages (associated text or images): **{LF:AssociatedPages= "Y"}**
entries with a certain number of pages • Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:PageCount=*integer*}** *integer* is the number of pages in the document. =, <>, <, <=, >, >= operators are supported.	To search for entries that have more than 10 pages**:** **{LF:PageCount>10}**
entries that have been OCRed • Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:AssociatedPages= "Y"} & {LF:OCR="all"}** **all** means all the pages in the document that have been OCRed.	To search for entries that have text associated with all their pages (have been OCRed): **{LF:OCR="all"}**

To Search For…	Syntax	Example
entries that have not been OCRed • Can be performed using the clients' search interface. • Not case-sensitive.	**{LF:AssociatedPages = "Y"} & {LF:OCR="none"}** **none** means no pages have text associated with them.	To search for entries that do not have text associated with them (have not been OCRed): **({LF:OCR="none"}**
entries where some pages have been OCRed and some have not • Can be performed using the clients' search interface. • Not case-sensitive.	**({LF:AssociatedPages = "Y"} & {LF:OCR="some"})** **some** means at least one associated page has text, but not all associated pages have text.	To search for entries that have at least one page of text associated with it (has been OCRed) and at least one page that does not have text associated with it (has not been OCRed): **{LF:OCR="SOME"}**
entries where some or all pages have been OCRed • Can be performed using the clients' search interface. • Not case-sensitive.	**({LF:AssociatedPages = "Y"} & {LF:OCR<>"none"})**	To search for entries that have at least one page of text associated with it: **{LF:OCR<>"none"}**
entries that do or do not contain image pages • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LF:Img="Y"}** "Y", "Yes", "1", or "true", mean the document has image pages. **{LF:Img="N"}** "N", "No", "0", or "false" mean the document has no image pages.	To search for an entry with image pages: **{LF:Img="Y"}** To search for an entry without image pages: **{LF:Img="N"}**

Indexed and Indexable Searches

To Search For…	Syntax	Example
entries that have been indexed • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LF:Indexed=Y}** **Y**, **Yes**, **1**, or **true** mean the entry is indexed.	To search for pages that have been indexed (made full-text searchable): **{LF:Indexed=Y}**
entries that have not been indexed • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LF:Indexed=*N*}** **N**, **No**, **0**, or **false** mean the entry has not been indexed.	To search for pages that have not been indexed (not made full-text searchable): **{LF:Indexed=N}**
entries that are not indexable • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LF:NotIndexable=Y}** **Y**, **Yes**, **1**, or **true** mean the entry is not indexable.	To search for all entries that are not indexable (any file that cannot contain text, such as an mp3 audio file): **{LF:NotIndexable=Y}**
entries that are indexable • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LF:NotIndexable=*N*}** **N**, **No**, **0**, or **false** mean the entry is indexable.	To search for all entries that are indexable (any file that contains text): **{LF:NotIndexable=N}**

Tag Searches

To Search For…	Syntax	Example
entries based on their tags • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:Tags="*tagname*"}** *tagname* is the name of the tag assigned to the entry.	To search all entries assigned the Classified tag: **{LF:Tags="Classified"}** To search for all entries assigned the Confidential tag with an Author field containing "Reality": **{LF:Tags="Confidential"} & {[]:[Author] ~= "Reality"}**
entries based on their tag comments • Can be performed using the web client search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:TagComment="*text*"}** **{LF:TagComment="*text*"}** Replace *text* with your search term.	**{LF:TagComment="*"}** * is a wildcard and will search for all tag comments. **{LF:TagComment=""}** Finds all entries that are tagged but do not have tag comments associated with them. **{LF:TagComment~="HR"}** Searches for all tag comments that contain "HR".
entries that have any tag besides, or in addition, to a specified tag • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:Tags <> "*tagname*"}** *tagname* is the name of the tag assigned to the entries you do not want returned.	To search for all entries that have any tag besides, or in addition to, the Pending tag: **{LF:Tags <> "Pending"}** **Note**: This search will return a document with the tag Classified or a document with both the tags Classified and Pending. It will not return a document that only has the Pending tag. To return entries that have any tag besides Pending and do not have the Pending tag use the following syntax: **{LF:Tags <> "Pending} – {LF:Tags="Pending"}**

To Search For…	Syntax	Example
entries with a combination of a tag and a tag comment • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**({LF:Tags="*tagname*"}) & {LF:TagComment="*text*"}** *tagname* is the name of the tag. *text* is the text in the tag comment.	To search for an entry assigned the Poetry tag and that has the comment "read by Thursday": **({LF:Tags="*Poetry*"}) & {LF:TagComment="*read by Thursday*"}**

Entry Creator Search

To Search For…	Syntax	Example
entries created by a specific user • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:Creator="*user*"}** *user* is the name of the user who created the entry. **Note:** The only valid format for searching for entries created by a Windows-authenticated user is: **"*domain\user*".** You can search for Windows-authenticated users across all domains with ***\user***.	To search for all entries created by Misty: **{LF:Creator="Misty"}** To search for all entries not created by Misty and modified on or after May 1, 2018: **{LF:Creator <> "Misty"} & {LF:Modified >= "05/01/18"}**

Entry Owner Search

To Search For	Syntax	Example
entries owned by a specific user • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:Owner="*user*"}** *user* is the name of the user who owns the entry. **Note:** The only valid format for searching for entries owned by a Windows-authenticated user is: **"*domain\user*"** You can search for Windows-authenticated users across all domains with ***\user***.	To search for all entries owned by Roger: **{LF:Owner="Roger"}** To search for all entries owned by Roger and all entries that contain the text "Roger" in an indexed field: **{LF:Owner="Roger"} | {[]:[] ~= "Roger"}**

To Search For	Syntax	Example
entries with no owner • Can only be performed using advanced search syntax.	**{LF:ID>0} - {LF:owner="*"}**	This search will return all entries that do not have an owner. The syntax subtracts all documents with an owner (**owner="*"**) from all documents in your repository (**ID>0**).

Last Modified By Search

To Search For	Syntax	Example
entries that were most recently modified by a specific user • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:LastModifiedBy= "*user*"}** *user* is the name of the user who last modified the entry. **Note:** The only valid format for searching for entries owned by a Windows-authenticated user is: **"*domain\user*"** You can search for Windows-authenticated users across all domains with ***user*.**	To find all the documents last modified by Eric: **{LF:LastModifiedBy="Eric"}** To find all documents that were last modified by Eric and not created by ADMIN: **{LF:LastModifiedBy="Eric"} – {LF:Creator="ADMIN"}**

Entry ID Searches

To Search For…	Syntax	Example
entry with a specific entry ID number • Can be performed through the clients' search interface. • Wildcards permitted.	**{LF:ID=*integer*}** *integer* is the entry ID number.	To find the document with the entry ID number 187: **{LF:ID=187}**
entries created after a specific entry • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LF:ID>*integer*}** *integer* is the entry ID number.	To find all entries created after the document with the entry ID number 187 was created: **{LF:ID>187}** **Note:** Use **>=** to include the entry with the specified ID number in the search results.

To Search For…	Syntax	Example
entries created before a specific entry • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LF:ID<*integer*}** *integer* is the entry ID number.	To find all entries created before the document with the entry ID number 187 was created: **{LF:ID<187}** **Note:** Use **<=** to include the entry with the specified ID number in the search results.

Document Links Searches

To Search For…	Syntax	Example
entries linked by a specific document relationship • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:Relation= "*relationshipname*"}** *relationshipname* is the name of the document relationship. **Note**: If you type in a number for ***relationshipname***, Laserfiche will search by the link's ID number.	To search for all entries linked by the Message relationship: **{LF:Relation="message"}** **Note:** Inverse document relationships (e.g., "Attachment" and "Message") will return the same results.
entries that are the source documents for a specific document relationship (e.g., the "supersedes" document in the "supersedes/superseded by" relationship) • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:RelationSRC= "*relationshipname*"}** *relationshipname* is the name of the document relationship. **Note**: If you type in a number for *relationshipname*, Laserfiche will search by the link's ID number.	To find all entries that are source documents in the "supersedes/superseded by" relationship: **{LF:RelationSRC= "supersedes"}**

To Search For…	Syntax	Example
entries that are the destination documents for a specific document relationship (e.g., the "superseded by" document in the "supersedes/ superseded by" relationship) • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:RelationDST= "*relationshipname*"}** *relationshipname* is the name of the document relationship. **Note**: If you type in a number for *relationshipname*, Laserfiche will search by the link's ID number.	To find all entries that are destination documents in the "supersedes/superseded by" relationship: **{LF:RelationDST= "superseded by"}**
entries that have a link group comment • Can be performed using the clients' search interfaces. **Note**: This search is an alias of **{LF:vercomment}**—not to be confused with **{LFVER:Comment}**.	**{LF:LinkGroupComment=""}** **{LF:LinkGroupComment~=""}** Insert your search term inside the quotation marks.	**{LF:LinkGroupComment=** **"June Invoice"}** This search will return any entries with link group comments that are *exactly* "June Invoice". **{LF:LinkGroupComment ~= "'personnel file'"}** This search will return any link group comments that *contain* the phrase "personnel file".
entries in a document relationship, based on the custom properties of that relationship • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LFLINKPROP: [*Your Property*]: "*value*"}** Replace *Property* with the name of the custom property, and *value* with the value of that property. **Note**: The property name does not have to be enclosed in square brackets if it contains no punctuation or whitespace and begins with a letter.	To find entries in a relationship with the custom property Size and a value 2 for that property: **{LFLINKPROP: [Size]= "2"}** To find entries in a relationship with the custom property Size and any value for that property: **{LFLINKPROP: [Size]= "*"}**

Version Searches

To Search For	Syntax	Example
entries that have a version comment containing specific text • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFVER:Comment="*text*"}** *text* is the text of the comment. **{LFVER:Comment="*"}** * is a wildcard and will search for all version comments. **{LFVER:Comment=""}** There is nothing inside the quotation marks. This will return all empty version comments.	Multiple documents in a repository contain new versions of their originals, and the comment "read on Thursday" was added to one of the new versions. To search for the version with this comment: **{LFVER:Comment ="read on Thursday"}**
entries by version number • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LFVER:Version=*integer*}** *integer* is a version number that the document has. The number specified does not have to be the document's current version; it can be an older version.	To search for documents that have a version number 4: **{LFVER:Version =4}** To search for documents that have a version number 2 or later and were last modified by the user Editor: **{LF:LastModifiedBy="Editor" } & {LFVER:Version>=2}**
entries by version label • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFVER:Label="*text*"}** *text* is the text of the comment. **{LFVER:Label ="*"}** * is a wildcard and will search for all tag comments. **{LFVER:Label =""}** There is nothing inside the quotation marks. This will return all empty tag comments.	To search for a document version with the label Updated: **{LFVER:Label ="Updated"}**

To Search For…	Syntax	Example
entries whose version was created or modified by a specific user • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFVER:ModifiedBy="*user*"}** *user* is the name of the user who modified or created the version. **{LFVER:ModifiedBy="*"}** * is a wildcard and will search for all modified versions.	To search for a document version that was last modified by Tabitha: **{LFVER:ModifiedBy ="Tabitha"}**
entries whose version was created or modified on a specific date • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFVER:Modified="*date*"}** *date* is the date when the version was modified or created. You can also enter a date/time. See Date, Time, and Date/Time Formats for more information on acceptable formats.	To search for a document version that was modified or created before May 1, 2012: **{LFVER:Modified ="5/1/2012 12:00:00 AM"}**

Shortcut Searches

To Search For…	Syntax	Example
all shortcuts that link to a specific entry, using the latter's entry ID • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LF:ShortcutID=*integer*}** *integer* is the entry ID number.	To find shortcuts to a document that has entry ID number 2021: **{LF:ShortcutID=2021}**
all shortcuts that link to a specific entry using the latter's entry name • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted.	**{LF:ShortcutName= "*entryname*", Type = "*type*"}** *entryname* is the entry name. *type* refers to the type of entry the shortcut links to. This parameter may be omitted from the search string.	To find all shortcuts that link to a folder that is named Performance Reviews: **{LF:ShortcutName= "Performance Reviews"}** **Note:** If you have changed a shortcut's name, it will still show up when using this search.

Checked Out Documents Searches

To Search For…	Syntax	Example
entries that are currently checked out by a specific Laserfiche user • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:CheckOutUser= "*user*"}** *user* is the name assigned to a user. **Note:** The only valid format for searching for entries modified by a Windows-authenticated user is: **"*domain\user*"** You can search for Windows-authenticated users across all domains with ***user***.	To conduct a search that will return all documents currently checked out by Misty: **{LF:CheckOutUser= "Misty"}** To conduct a search that will return all documents that a Windows user named Mattie on the Laserfiche domain has checked out: **{LF:CheckOutUser= "Laserfiche\Mattie"}**
entries that are currently checked out by a specific Laserfiche user using his or her user ID • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LF:CheckOutUserID= *integer*}** *integer* is the user ID number. Only the = and <> operators are supported.	To conduct a search that will return all documents the user with user ID number 107 has checked out: **{LF:CheckOutUserID= 107}**
checked out electronic documents based on when they were checked out • Can only be performed using advanced search syntax.	**{LF:CheckoutDate="*date*"}** See the Date, Time, and Date/Time Formats section for more information on how to format dates. =, <>, <, <=, >, >= operators are supported.	To search for entries whose electronic documents were checked out before 2018: **{LF:CheckoutDate<"01/01/18" }**

UUID Search

To Search For…	Syntax	Example
documents with the specified UUID (Universally Unique Identifier) • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LF:UUID="*uuid*"}** *uuid* is the Universally Unique Identifier.	To search for an entry with UUID number 550e8400-e29b-41d4-a716-446655440000: **{LF:UUID="550e8400-e29b-41d4-a716-446655440000"}**

Digital Signature Searches

To Search For…	Syntax	Example
entries signed by a digital certificate with a certain subject • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFSIG:CertSubject= "user"}** *user* is the name of the subject of the certificate (the user or organization to whom the certificate was issued).	To conduct a search that will return all documents signed by certificates with the subject Misty: **{LF:certsubject = "Misty"}**
entries that are signed by a Windows user • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFSIG:signedby= "*user*"}** *user* is the user whose signature is being searched for . **Note:** The only valid format for searching for entries created by a Windows-authenticated user is: **"*domain\user*"** You can search for Windows-authenticated users across all domains with ***user***.	To conduct a search that will return all documents signed by Walt: **{LF:signedby = "Walt"}**

To Search For…	Syntax	Example
entries that are signed at, within, before, or after certain dates • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:signeddate="*date*"}** *date* is the date the entry was signed. See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To conduct a search that will return all documents signed on or after June 15, 2012: **{LF:signeddate >= "6/15/2012"}**
entries with a specific digital signature comment • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFSIG:comment= "*comment*"}** *comment* is the word or phrase being searched for.	To conduct a search that will return all documents with a signature comment of "Tentatively Approved": **{LFSIG:comment = "Tentatively Approved"}**

Classification Level Searches

To Search For…	Syntax	Example
entries with classification levels equal to a certain number or within a certain range • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:ClassificationLevel= "*level*"}** *level* is an integer representing the classification level.	To find all documents with classification level 5: **{LF:ClassificationLevel= "5"}** To find all documents with a classification level lower than 5: **{LF:ClassificationLevel<"5"}**

Business Process Searches

To Search For…	Syntax	Example
entries associated with a business process with a certain name • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:bpname="*name*"}** *name* is the name of the business process.	To find all entries associated with the business process named Continuum: **{LF:bpname="Continuum"}**

To Search For…	Syntax	Example
entries associated with a business process entity that has a certain status • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LF:bpestatus="*status*"}** *status* is the status of the business process.	To find all entries associated with business processes that have the status Under Review: **{LF:bpestatus="Under Review"}**
entities associated with a business process entity with a specific start date or range of start dates • Can be performed using the clients' search interfaces.	**{LF:bpestartdate="*startdate*"}** See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To find all entries started after April 2, 2018: **{LF:bpestartdate>"04/02/18"}**
entities associated with a business process entity with a specific completion date or range of completion dates • Can only be performed using advanced search syntax.	**{LF:bpecompleteddate="*start date*"}** See Date, Time, and Date/Time Formats for more information on how to format dates.	To find all entries completed before April 2, 2018: **{LF:bpecompleteddate<"04/02/18"}**

Records Management Searches

Note: Shortcuts cannot be records, so searches on record management properties will not return any shortcuts that are in record folders. However, shortcuts that point to entries that match the search criteria will be returned.

To Search For…	Syntax	Example
record series based on their codes • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFRM:Code =*"code"*}** = and <> are the only supported operators.	To find a record series with the series code "1-3-4-7": **{LF:bpname="1-3-4-7"}**
record series based on their description • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.	**{LFRM:seriesdesc =*"description"*}** Supported operators are = and <>.	To find record series with a description of "California State Registrations": **{LFRM:SeriesDesc="Californi a State Registrations"}**
records based on whether they are permanent • Can be performed using the clients' search interface. • Not case-sensitive.	**{LFRM:PermanentRecord="*Y* "}** Use "y", "yes", "1", or "true" to search for permanent records. Use "n", "no", "0", or "false" to search for non-permanent records.	To find all non-permanent records: **{LFRM:PermanentRecord="0 "}**
records based on whether they are vital • Can be performed using the clients' search interface. • Not case-sensitive.	**{LFRM:vitalrecord="*y*"}** Use "y", "yes", "1", or "true" to search for vital records. Use "n", "no", "0", or "false" to search for non-vital records.	To find non-vital records: **{LFRM:vitalrecord="false"}**

To Search For…	Syntax	Example
records based on review dates • Can be performed using the clients' search interfaces.	**{LFRM:NextReviewDate="*mm/dd/yyyy*"}** **{LFRM:LastReviewDate="*mm/dd/yyyy*"}** Date/times can also be used. See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To find records that are eligible for review between October 1, 2028 and October 8, 2028: **{LFRM:NextReviewDate>="10/01/2028 12:00:00 AM",NextReviewDate<="10/08/2028 11:59:59 PM"}** To find records that were last reviewed before Jan 1, 2015: **{LFRM:LastReviewDate<"01-01-2015"}**
records based on type of review cycle • Can be performed using the clients' search interface. • Not case-sensitive.	**{LFRM:VitalRecordReviewCycle="*Value*"}** For *Value*, use the codes of cycle definitions.	To find all records that have a monthly review cycle: **{LFRM:VitalRecordReviewCycle="mon"}**
record series, record folders, and records based on whether they are frozen • Can be performed using the clients' search interface. • Not case-sensitive.	**{LFRM:FrozenDocument="*N*"}** Use "y", "yes", "1", or "true" to search for frozen records management entries. Use "n", "no", "0", or "false" to search for non-frozen records management entries.	To find all frozen records management entries: **{LFRM:FrozenDocument="1"}**
frozen records management entries based on their reason for being frozen • Can be performed through the clients' search interface. • Wildcards permitted.	**{LFRM:FrozenReason="*Value*"}**	To find all records management entries that were frozen for the reason "Court Order": **{LFRM:FrozenReason="Court Order"}**

To Search For…	Syntax	Example
records management entries based on when they were cutoff • Can be performed using the clients' search interfaces.	**{LFRM:ActualCutoffDate="*mm/dd/yyyy*"}** See the Date, Time, and Date/Time Formats section for more information on how to format dates. **Note**: **LFRM:CutoffDate** carries out the same search.	To find all records that have been cutoff: **{LFRM:ActualCutoffDate="*"}**
records management entries based on when they become eligible for cutoff • Can be performed using the clients' search interfaces.	**{LFRM:EligibleForCutoff<="*mm/dd/yyyy*"}** Objects that have already been cutoff are not returned by this search. Use **LFRM:ActualCutoffDate** to find such objects. See the Date, Time, and Date/Time Formats section for more information on how to format dates.	To find all records and records folders that became eligible for cutoff between January 1, 2008, and June 1, 2008, but have not actually been cutoff: **{LFRM:EligibleForCutoff>="01/01/2008"} & {LFRM:EligibleForCutoff<="06/01/2008"}**
records management entries based on their type of cutoff instruction • Can be performed using the clients' search interfaces.	**{LFRM:RetentionType="*Value*"}** For *Value*, use the following codes. If you want to include entries that satisfy at least one of multiple cutoff instructions, separate the codes with a comma. **E**: Event **T**: Time **N**: Time + Event **S**: Superseded **I**: Interval **R**: Interval + Event **L**: Disposition Action **NULL**: No cutoff instruction	To find all records that have either an Event-based cutoff instruction or a Time-based cutoff instruction: **{LFRM:RetentionType="E,T"}** To find all records with no cutoff instruction: **{LFRM:RetentionType="NULL"}**

To Search For…	Syntax	Example
records management entries based on their cutoff instruction IDs • Can be performed using the clients' search interfaces.	**{LFRM:CoffCritID="*ID*"}**	To find records with no cutoff instruction: **{LFRM:CoffCritID=""}**
records management entries based on the type of their final disposition • Can be performed using the clients' search interfaces.	**{LFRM:FinalDispAction="*DispositionAction*"}** Use the following codes in place of *DispositionAction*. **A**: Accession **D**: Destruction **N**: No final disposition action.	To find all records with no assigned final disposition action: **{LFRM:FinalDispAction="N"}**
records management entries based on whether final disposition has been confirmed • Can only be performed using advanced search syntax.	**{LFRM:FinalDispDate="*date*"}** Use * in place of *date* to find all records that have had final disposition confirmed. See the Date, Time, and Date/Time Formats section for more information on how to format dates. =, <>, <, <=, >, >= operators are supported.	To find all records that have had final disposition confirmed: **{LFRM:FinalDispDate="*"}** To find all records that have had final disposition confirmed before August 12, 2017: **{LFRM:FinalDispDate<"08/12/17"}**

To Search For…	Syntax	Example
records management entries based on their disposition eligibility • Can be performed using the clients' search interfaces.	**{LFRM:DispEligibilityAction="*Value*"}** **{LFRM:DispEligibilityAction="*Value*", EligibilityDate="*mm/dd/yyyy*"}** Use the following codes in place of *Value*: **U**: All records that have a calculated eligibility date for any disposition action. **T**: All eligible records with a projected transfer eligibility date. **A**: All records with an Accession final disposition and a final disposition eligibility date. **D**: All records with a Destruction final disposition and a final disposition eligibility date. **Note**: If you use the <> operator, the search will not return records that lack a disposition action.	To find all records with either a destruction or accession final disposition and a final disposition eligibility date: **{LFRM:DispEligibilityAction="D"} | {LFRM:DispEligibilityAction="A"}** To find all records that became eligible for destruction between June 1, 2008, and December 31, 2008: **{LFRM:DispEligibilityAction="D", EligibilityDate>="06/01/2008", EligibilityDate<="12/31/2008"}** **Note:** See the Date, Time, and Date/Time Formats section for more information on how to format dates.
records management entries based on whether they are *currently* available for disposition • Can be performed using the clients' search interfaces.	**{LFRM:ImmediateDispEligibility="*Value*"}** Use the following codes in place of *Value*: **U**: All records currently eligible for any disposition action. **T**: All records currently eligible for a transfer. **A**: All records currently eligible for accession. **D**: All records currently eligible for destruction.	To find all records currently eligible for transfer that have not yet been transferred: **{LFRM:ImmediateDispEligibility="T"}** To find all records that are either currently eligible for destruction or currently eligible for accession: **{LFRM:ImmediateDispEligibility="D"} | {LFRM:ImmediateDispEligibility="A"}**

To Search For…	Syntax	Example
records management entries based on their current location • Can be performed through the clients' search interface. • Wildcards permitted.	**{LFRM:CurrentLocation="*Value*"}** *Value* is the name of the location of interest.	To find all records currently located at Archive A: **{LFRM:CurrentLocation="Archive A"}**
record series and children of record series based on their disposition authority • Can be performed through the clients' search interface. • Wildcards permitted.	**{LFRM:DispositionAuthority="*Value*"}** *Value* is the name of the disposition authority.	To find all records with no disposition authority: **{LFRM:DispositionAuthority=""}** To find all records where the disposition authority is the National Archives: **{LFRM:DispositionAuthority="National Archives "}**
records management entries based on the IDs of their retention schedules • Can be performed using the clients' search interfaces.	**{LFRM:RetSchedID = "*id*"}**	To find all records with a retention schedule ID of "0": **{LFRM:RetSchedID = "0"}** To find all records without a retention schedule: **{LFRM:RetSchedID = ""}**
record folders based on whether they have been closed • Can only be performed using advanced search syntax.	**{LFRM:Closed="*boolean*"}** In place of *boolean*, • Use "y", "yes", "1", or "true" to search for closed record folders • Use "n", "no", "0", or "false" to search for record folders that have not been closed.	To find all closed record folders: **{LFRM:Closed="1"}**

To Search For…	Syntax	Example
records based on whether they must be superseded before undergoing final disposition (meaning that the **Records must be superseded** option was chosen in their final disposition instructions) • Can only be performed using advanced search syntax.	**{LFRM:reqsuperseded="*boolean*"}** In place of *boolean*, • Use "y", "yes", "1", or "true" to search for records that must be superseded before undergoing final disposition. • Use "n", "no", "0", or "false" to search for records that do not have to be superseded before undergoing final disposition.	To find all records for which the **Records must be superseded** option was chosen: **{LFRM:reqsuperseded="y"}**
records based on events assigned to them • Can be performed using the clients' search interfaces.	**{LFRM:Event="*name*"}** In place of *name*, specify the name of the event of interest.	To find all records that have the Termination event assigned to them: **{LFRM:Event="Termination"}**
records based on set events • Can be performed using the clients' search interfaces.	**{LFRM:SetEvent="*name*"}** In place of *name*, specify the name of the event of interest.	To find all records that have the End of Lease event set: **{LFRM:SetEvent="End of Lease"}**
records based on how often they need to be reviewed • Can only be performed using advanced search syntax.	**{LFRM:vitalrecordreviewbyday="*n*"}** **{LFRM:vitalrecordreviewbymo="*n*"}** *n* is an integer indicating the interval in days/months between reviews. *n* should be a positive integer.	To find all records that need to be reviewed daily: **{LFRM:vitalrecordreviewbyday="1"}** To find all records that need to be reviewed every 3 months: **{LFRM:vitalrecordreviewbymo="3"}**

Entry Access Rights Searches

These searches find entries that have certain entry access rights directly assigned to them. They do not return entries with the equivalent effective rights or that inherit the specified entry access rights. You can carry out an attribute search on

entry access rights: To specify that a single entry access right matches multiple conditions, enclose all the conditions within the same pair of curly braces and separate the conditions with commas.

To Search For…	Syntax	Example
entries with entry access rights that are explicitly assigned (whether allowed or denied) to a trustee • Can only be performed using advanced search syntax. • Wildcards permitted.	**{LFACE:trustee = "*Name*"}** Replace *Name* with the name of the trustee. To search the Everyone group, use **\everyone** in place of *Name*. **Note:** The only valid format for searching for entries created by a Windows-authenticated user is: **"domain\user"** You can search for Windows-authenticated users across all domains with **\user**.	To search for entries that have entry access rights explicitly denied to any trustee: **{LFACE:trustee = "*", type="Deny")**
entries explicitly assigned an entry access right set to Allow or Deny. • Can be performed using the clients' search interface. • Not case-sensitive. • Wildcards permitted.Can only be performed using advanced search syntax.	**{LFACE:type = "*type*"}** For type, specify **Allow**, **Deny**, or *****.	To find all entries where an entry access right of either Allow or Deny is *explicitly* granted: **{LFACE:type = "*"}**

To Search For…	Syntax	Example
entries explicitly assigned an entry access right (whether allowed or denied) with a certain type of scope. • Can only be performed using advanced search syntax. • Not case-sensitive.	**{LFACE:scope = "*scopeType*"}** For *scopeType*, specify one of the following bolded strings: • **documentsonly**: Documents only • **thisentry**: This entry only • **folders**: This folder and subfolders • **foldersonly**: Subfolders only • **immediate**: This folder and its immediate children • **immediatechildren**: Immediate children only • **immediatedocuments**: Documents that are immediate children only • **notthisentry**: Subfolders and documents only • **all**: This folder, subfolders, and documents **Note**: If the scope term is omitted from an LFACE query, all scopes will be included.	To find all entries with explicitly assigned entry access rights that have a scope of either Documents that are immediate children only or Subfolders only: **{LFACE:scope IN ("foldersonly", "immediatedocuments")}** **Note:** * cannot be used with the **IN** or **NOT IN** operators.

To Search For…	Syntax	Example
entries that have any combination of entry access rights explicitly assigned • Can only be performed using advanced search syntax. • Not case-sensitive. • Wildcards permitted. **Note**: You can use a "*" wildcard to indicate the existence of any rights in addition to those specified. * cannot be used with the **IN** and **NOT IN** operators.	**{LFACE:rights = "*rightsString*"}** Replace *rightsString* with a string composed of any combination of the following three-character strings: • **brs**: Browse • **rea**: Read • **mcn**: Modify Contents • **ada**: Append Data • **del**: Delete Entry • **ren**: Rename • **dpg**: Delete Document Pages • **san**: See Annotations • **ann**: Annotate • **red**: See Through Redactions • **wme**: Write Metadata • **crd**: Create Documents • **crf**: Create Folders • **rac**: Read Entry Security • **wac**: Write Entry Security • **cow**: Change Entry Owner • **srd**: Set Last Review Date • **frz**: Freeze/Unfreeze • **evt**: Set Event Date • **cls**: Close/Reopen Folder The string **---** (three dashes) indicates that no rights have been explicitly assigned.	To find entries that are explicitly assigned both the Browse right and the Read right, but no other rights: **{LFACE:rights = "BrsRea"}** To find entries that are assigned both the Browse right and the Read right, and that may also have other rights: **{LFACE:rights = "BrsRea*"}** To find entries where the explicitly assigned entry access rights are anything other than exactly and only the Read and Browse rights (entries with explicitly assigned rights in addition to both the Read and Browse rights will be included in the results): **{LFACE:rights <> "BrsRea"}** To find entries where users who are not the ADMIN user have not been explicitly assigned any rights: **{LFACE:rights = "---", trustee<>"ADMIN"}**

Geography Searches

In Laserfiche 10.3 or later, you can use the advanced search syntax to look for entries that have associated geographic objects meeting certain conditions. Attribute searches, where you specify multiple conditions that a single geographic object must meet, are possible. The only restrictions on specifying multiple conditions are as follows:

- The **Intersects** condition cannot appear more than once.
- The **Object** clause must appear exactly once if there are any **Intersects** or **Distance** conditions. If there are no **Intersects** or **Distance** conditions, the **Object** clause cannot appear.

For the **LFGEO:Length**, **LFGEO:Distance** and **LFGEO:Area** searches, the units used are defined by the spatial reference identifier (SRID) of the geographical objects.

All geography searches can be performed only through advanced search syntax.

To Search For…	Syntax	Example
entries associated with a specific geographic object	**{LF:Geography="Extended*WKTrepresentation*"}** Replace *ExtendedWKTrepresentation* with the Extended Well-known text (EWKT) string for the object. **Note**: This search is an exact search for the object, supporting only the = and <> operators. It does not support finding nearby objects.	To search for a point with the coordinates (-118.189753 33.821306) in the WGS 84 world geodetic system (the reference system used by GPS), specify the system's SRID of 4326: **{LF:Geography="SRID=4326; POINT(-118.189753 33.821306)"}**
geographical objects based on their type	**{LFGEO:Type="*type*"}** Replace *type* with one of the following: • POINT • MULTIPOINT • LINESTRING • MULTILINESTRING • POLYGON • MULTIPOLYGON	To find all point objects: **{LFGEO:Type="point"}**

To Search For…	Syntax	Example
geographical objects based on their dimension	**{LFGEO:Dimension=*integer*}** Replace *integer* with the dimension of the object. Points have 0 dimensions, lines have 1 dimension, and polygons have 2 dimensions. **Note:** Only the = and <> operators are supported.	To find all polygons: **{LFGEO:Dimension=2}**
geographical objects based on their length	**{LFGEO:Length > *number*}** *number* must be a non-negative number. =, <>, <, <=, >, >= operators are supported.	To find all geographical objects with a length 5 or less (in units defined by objects' SRIDs): **{LFGEO:Length <= 5}**
geographical objects based on their area	**{LFGEO:Area > *number*}** *number* must be non-negative. Points and lines have an area of 0. =, <>, <, <=, >, >= operators are supported.	To search for geographical objects with no area, including points and lines: **{LFGEO:Area = 0}**

To Search For…	Syntax	Example
geographical objects based on whether they intersect an object defined in the query	**{LFGEO:Intersects=*boolean*,Object="*ExtendedWKTDescription*"}** In place of *boolean*: • Use "y", "yes", "1", or "true" to search for objects that intersect with the specified object. • Use "n", "no", "0", or "false" to search for objects that *do not* intersect with the specified object. In place of *ExtendedWKTDescription*: • Use the same kind of object used in **LF:Geography** queries, i.e. an extended WKT string.	**{LFGEO:Intersects=1,Object="POLYGON((30 10, 40 40, 20 40, 10 20, 30 10))"}**
geographical objects based on their distance from another object specified in the query	**{LFGEO:Distance<*number*,Object="*ExtendedWKTDescription*"}** In place of *number*, use any non-negative number. In place of *ExtendedWKTDescription*, use an extended WKT object.	**{LFGEO:Distance<100,Object="POLYGON((30 10, 40 40, 20 40, 10 20, 30 10))"}**

Using Advanced Search Syntax with Custom Quick Searches

The Quick Search tool allows you to perform a general search without having to open the Search or Advanced Search pane and configure search types. By default, you can search document text, entry names, fields, annotation text, or any combination of these by selecting the searches you want to use from the Quick Search drop-down menu. You can also create custom quick searches using any of the advanced search syntax strings detailed above.

Custom quick searches are especially useful for making frequently used searches easily accessible when searching.

> **Example**: Wendy works in her company's Human Resources department and every week she searches for all entries that have blank field values for the Employee Name, Social Security Number, and Home Address fields, then fills in the missing information. Rather than opening the Search or Advanced Search pane and typing the advanced search syntax for these searches, she uses the syntax to create a custom quick search that she can easily access every week.

To create a custom Quick Search in the Windows client:

1. Click the down arrow to the right of the Search bar, and select **Customize Quick Searches**.
2. In the **Customize Quick Searches** dialog box, click **Create**.
3. Type a name for your custom quick search in the text box that appears in the **Custom Searches** list.
4. Make sure your new search is selected on the **Custom Searches** list. In the **Search Query** textbox below, type the advanced search syntax for your search, using the token **%(SearchTerm)** to stand in for the search terms users will eventually provide in the Quick Search bar.

> **Example:** Wendy needs to search for blank field values in different fields. The advanced search syntax for searching for empty values in a particular field is **{[]:[*fieldname*]=""}**. Because she will provide

the field name as her search term when she searches, she replaces *fieldname* with **%(SearchTerm)** in the syntax. With this substitution, the syntax reads **{[]:[“%(SearchTerm)”]=“”}**.

5. Click **OK**. Your custom search will now appear in the Quick Search drop-down menu.

In the **Customize Quick Searches** dialog of the Windows client, you can manage your custom Quick Searches as follows:

- Change the order in which custom searches appear in the Quick Search drop-down menu by using the **Move Up** and **Move Down** buttons to the left of the **Custom Searches** list.
- Remove a custom search from the menu by clicking **Remove**.
- Rename a custom search by selecting the search, clicking on it again, and typing in the new name.

To create a custom Quick Search in the web client:

1. Click the downward-pointing arrow to the left of the search bar.
2. Select **Manage custom quick search**.
3. Select **Add** to open the **Create Custom Quick Search** dialog box.
4. In the **Name** option, specify a name for your custom quick search.
4. In the **Search query** option, type the advanced search syntax for your new search type. The token **%(SearchTerm)** will be replaced with the user's actual search term when the search is performed.

   Example: The advanced search syntax for a **Created By** search is: **{LF:Creator="*username*"}**. To search for documents created by a specific user, you would type the syntax for a **Created By** search, with the **%(SearchTerm)** token in place of the creator's name. The syntax would therefore be: **{LF:Creator="%(SearchTerm)"}**.

5. Select either **Apply to myself** to save the custom quick search only for yourself, or **Apply to the following group** to share the custom quick search with a particular group. You can select the Everyone group to share the search with all users.

   Note: You must have the **Manage Repository Configuration** privilege to share custom quick searches with a group. To configure a custom quick search that is available to a particular group, you have to use the web client for the configuration.

6. Select **Save** to save the custom quick search.

In the **Manage Custom Quick Search** dialog of the web client, you can manage your custom Quick Searches as follows:

- Change the order in which custom searches appear in the Quick Search drop-down menu by dragging your custom searches into the order you want.
- Remove a custom search from the drop-down menu by clicking **Delete** next to its name.
- Change the properties (including name) of a custom search by clicking **Edit** beside its name and making the appropriate changes.

Laserfiche®

Advanced Search Syntax 10.3
October 2018

Authors: Leif Hancox-Li, Misty Kalousek, Gordon Wong, Dave Haas, Sarah Seene, and Sierra Jahoda

Description:
Advanced search syntax allows you to carry out complicated searches in a Laserfiche repository, incorporating conditions that may be difficult or impossible to specify using the graphical search box. This paper describes the search syntax and provides numerous examples. Updated for Laserfiche 10.3.

Laserfiche
3545 Long Beach Blvd.
Long Beach, CA 90807
U.S.A

Phone: +1.562.988.1688
www.laserfiche.com