

NAME

notmuch-search-terms – syntax for notmuch queries

SYNOPSIS

notmuch count [option ...] <search-term> ...

notmuch dump [--gzip] [--format=(batch-tag|sup)] [--output=<file>] [--] [<search-term> ...]

notmuch reindex [option ...] <search-term> ...

notmuch search [option ...] <search-term> ...

notmuch show [option ...] <search-term> ...

notmuch tag +<tag> ... -<tag> [--] <search-term> ...

DESCRIPTION

Several notmuch commands accept a common syntax for search terms.

The search terms can consist of free-form text (and quoted phrases) which will match all messages that contain all of the given terms/phrases in the body, the subject, or any of the sender or recipient headers.

As a special case, a search string consisting of exactly a single asterisk ("*") will match all messages.

Search prefixes

In addition to free text, the following prefixes can be used to force terms to match against specific portions of an email, (where <brackets> indicate user-supplied values).

Some of the prefixes with <regex> forms can be also used to restrict the results to those whose value matches a regular expression (see **regex(7)**) delimited with //, for example:

```
notmuch search 'from: "/bob@.*[.]example[.]com/" '
```

body:<word-or-quoted-phrase>

Match terms in the body of messages.

from:<name-or-address> or from:/<regex>/

The **from:** prefix is used to match the name or address of the sender of an email message.

to:<name-or-address>

The **to:** prefix is used to match the names or addresses of any recipient of an email message, (whether To, Cc, or Bcc).

subject:<word-or-quoted-phrase> or subject:/<regex>/

Any term prefixed with **subject:** will match only text from the subject of an email. Searching for a phrase in the subject is supported by including quotation marks around the phrase, immediately following **subject:**.

attachment:<word>

The **attachment:** prefix can be used to search for specific filenames (or extensions) of attachments to email messages.

mimetype:<word>

The **mimetype:** prefix will be used to match text from the content-types of MIME parts within email messages (as specified by the sender).

tag:<tag> or tag:/<regex>/ or is:<tag> or is:/<regex>/

For **tag:** and **is:** valid tag values include **inbox** and **unread** by default for new messages added by **notmuch-new(1)** as well as any other tag values added manually with **notmuch-tag(1)**.

id:<message-id> or **mid:**<message-id> or **mid:**<regex>/

For **id:** and **mid:**, message ID values are the literal contents of the Message-ID: header of email messages, but without the '<', '>' delimiters.

thread:<thread-id>

The **thread:** prefix can be used with the thread ID values that are generated internally by notmuch (and do not appear in email messages). These thread ID values can be seen in the first column of output from **notmuch-search(1)**

thread:{<notmuch query>}

Threads may be searched for indirectly by providing an arbitrary notmuch query in {}. For example, the following returns threads containing a message from mallory and one (not necessarily the same message) with Subject containing the word "crypto".

```
% notmuch search 'thread:"{from:mallory}" and thread:"{subject:crypto}"'
```

The performance of such queries can vary wildly. To understand this, the user should think of the query **thread:**{<something>} as expanding to all of the thread IDs which match <something>; notmuch then performs a second search using the expanded query.

path:<directory-path> or **path:**<directory-path>/** or **path:**<regex>/

The **path:** prefix searches for email messages that are in particular directories within the mail store. The directory must be specified relative to the top-level maildir (and without the leading slash). By default, **path:** matches messages in the specified directory only. The "/" suffix can be used to match messages in the specified directory and all its subdirectories recursively. **path:""** matches messages in the root of the mail store and, likewise, **path:**** matches all messages.

path: will find a message if *any* copy of that message is in the specific directory.

folder:<maildir-folder> or **folder:**<regex>/

The **folder:** prefix searches for email messages by maildir or MH folder. For MH-style folders, this is equivalent to **path:**. For maildir, this includes messages in the "new" and "cur" subdirectories. The exact syntax for maildir folders depends on your mail configuration. For maildir++, **folder:""** matches the inbox folder (which is the root in maildir++), other folder names always start with ".", and nested folders are separated by "."s, such as **folder:.classes.topology**. For "file system" maildir, the inbox is typically **folder:INBOX** and nested folders are separated by slashes, such as **folder:classes/topology**.

folder: will find a message if *any* copy of that message is in the specific folder.

date:<since>..**until**> or **date:**<date>

The **date:** prefix can be used to restrict the results to only messages within a particular time range (based on the Date: header).

See **DATE AND TIME SEARCH** below for details on the range expression, and supported syntax for <since> and <until> date and time expressions.

The time range can also be specified using timestamps without including the date prefix using a syntax of:

```
<initial-timestamp>..final-timestamp>
```

Each timestamp is a number representing the number of seconds since 1970-01-01 00:00:00 UTC. Specifying a time range this way is considered legacy and predates the date prefix.

lastmod:<initial-revision>..**final-revision**>

The **lastmod:** prefix can be used to restrict the result by the database revision number of when messages were last modified (tags were added/removed or filenames changed). This is usually

used in conjunction with the **--uuid** argument to **notmuch-search(1)** to find messages that have changed since an earlier query.

query:<name>

The **query:** prefix allows queries to refer to previously saved queries added with **notmuch-config(1)**.

property:<key>=<value>

The **property:** prefix searches for messages with a particular <key>=<value> property pair. Properties are used internally by notmuch (and extensions) to add metadata to messages. A given key can be present on a given message with several different values. See **notmuch-properties(7)** for more details.

User defined prefixes are also supported, see **notmuch-config(1)** for details.

Operators

In addition to individual terms, multiple terms can be combined with Boolean operators (**and**, **or**, **not**, and **xor**). Each term in the query will be implicitly connected by a logical AND if no explicit operator is provided (except that terms with a common prefix will be implicitly combined with OR). The shorthand '**!<term>**' can be used for '**not <term>**' but unfortunately this does not work at the start of an expression. Parentheses can also be used to control the combination of the Boolean operators, but will have to be protected from interpretation by the shell, (such as by putting quotation marks around any parenthesized expression).

In addition to the standard boolean operators, Xapian provides several operators specific to text searching.

```
notmuch search term1 NEAR term2
```

will return results where term1 is within 10 words of term2. The threshold can be set like this:

```
notmuch search term1 NEAR/2 term2
```

The search

```
notmuch search term1 ADJ term2
```

will return results where term1 is within 10 words of term2, but in the same order as in the query. The threshold can be set the same as with NEAR:

```
notmuch search term1 ADJ/7 term2
```

Stemming

Stemming in notmuch means that these searches

```
notmuch search detailed
notmuch search details
notmuch search detail
```

will all return identical results, because Xapian first "reduces" the term to the common stem (here 'detail') and then performs the search.

There are two ways to turn this off: a search for a capitalized word will be performed unstemmed, so that one can search for "John" and not get results for "Johnson"; phrase searches are also unstemmed (see below for details). Stemming is currently only supported for English. Searches for words in other languages will be performed unstemmed.

Wildcards

It is possible to use a trailing '*' as a wildcard. A search for 'wildc*' will match 'wildcard', 'wildcat', etc.

Boolean and Probabilistic Prefixes

Xapian (and hence notmuch) prefixes are either **boolean**, supporting exact matches like *tag:inbox* or **probabilistic**, supporting a more flexible **term** based searching. Certain **special** prefixes are processed by notmuch in a way not strictly fitting either of Xapian's built in styles. The prefixes currently supported by notmuch are as follows.

Boolean

tag:, id:, thread:, folder:, path:, property:

Probabilistic

body:, to:, attachment:, mimetype:

Special from:, query:, subject:

Terms and phrases

In general Xapian distinguishes between lists of terms and **phrases**. Phrases are indicated by double quotes (but beware you probably need to protect those from your shell) and insist that those unstemmed words occur in that order. One useful, but initially surprising feature is that the following are equivalent ways to write the same phrase.

- "a list of words"
- a-list-of-words
- a/list/of/words
- a.list.of.words

Both parenthesised lists of terms and quoted phrases are ok with probabilistic prefixes such as **to:**, **from:**, and **subject:**. In particular

```
subject:(pizza free)
```

is equivalent to

```
subject:pizza and subject:free
```

Both of these will match a subject "Free Delicious Pizza" while

```
subject:"pizza free"
```

will not.

Quoting

Double quotes are also used by the notmuch query parser to protect boolean terms, regular expressions, or subqueries containing spaces or other special characters, e.g.

```
tag:"a tag"
```

```
folder:"/^.*(Junk|Spam)$/"
```

```
thread:"{from:mallory and date:2009}"
```

As with phrases, you need to protect the double quotes from the shell e.g.

```
% notmuch search 'folder:"/^.*(Junk|Spam)$/'
```

```
% notmuch search 'thread:"{from:mallory and date:2009}" and thread:{to:mallory
```

DATE AND TIME SEARCH

notmuch understands a variety of standard and natural ways of expressing dates and times, both in absolute terms ("2012-10-24") and in relative terms ("yesterday"). Any number of relative terms can be combined ("1 hour 25 minutes") and an absolute date/time can be combined with relative terms to further adjust it. A non-exhaustive description of the syntax supported for absolute and relative terms is given below.

The range expression

date:<since>..<>until>

The above expression restricts the results to only messages from <since> to <until>, based on the Date: header.

<since> and <until> can describe imprecise times, such as "yesterday". In this case, <since> is taken as the earliest time it could describe (the beginning of yesterday) and <until> is taken as the latest time it could describe (the end of yesterday). Similarly, date:janeury..february matches from the beginning of January to the end of February.

If specifying a time range using timestamps in conjunction with the date prefix, each timestamp must be preceded by @ (ASCII hex 40). As above, each timestamp is a number representing the number of seconds since 1970-01-01 00:00:00 UTC. For example:

date:@<initial-timestamp>..<>final-timestamp>

Currently, spaces in range expressions are not supported. You can replace the spaces with '_', or (in most cases) '-', or (in some cases) leave the spaces out altogether. Examples in this man page use spaces for clarity.

Open-ended ranges are supported. I.e. it's possible to specify date:..<>until> or date:<since>.. to not limit the start or end time, respectively.

Single expression

date:<expr> works as a shorthand for date:<expr>..<>expr>. For example, date:monday matches from the beginning of Monday until the end of Monday.

Relative date and time

[N]number] (years|months|weeks|days|hours|hrs|minutes|mins|seconds|secs) [...]

All refer to past, can be repeated and will be accumulated.

Units can be abbreviated to any length, with the otherwise ambiguous single m being m for minutes and M for months.

Number can also be written out one, two, ..., ten, dozen, hundred. Additionally, the unit may be preceded by "last" or "this" (e.g., "last week" or "this month").

When combined with absolute date and time, the relative date and time specification will be relative from the specified absolute date and time.

Examples: 5M2d, two weeks

Supported absolute time formats

- H[H]:MM[:SS] [(am|a.m.|pm|p.m.)]
- H[H] (am|a.m.|pm|p.m.)
- HHMMSS
- now

- noon
- midnight
- Examples: 17:05, 5pm

Supported absolute date formats

- YYYY-MM[-DD]
- DD-MM[-[YY]YY]
- MM-YYYY
- M[M]/D[D]/[YY]YY
- M[M]/YYYY
- D[D].M[M].[YY]YY
- D[D][[(st|nd|rd|th)] Mon[thname] [YYYY]
- Mon[thname] D[D][[(st|nd|rd|th)] [YYYY]
- Wee[kday]

Month names can be abbreviated at three or more characters.

Weekday names can be abbreviated at three or more characters.

Examples: 2012-07-31, 31-07-2012, 7/31/2012, August 3

Time zones

- (+|-)HH:MM
- (+|-)HH[MM]

Some time zone codes, e.g. UTC, EET.

SEE ALSO

notmuch(1), **notmuch-config(1)**, **notmuch-count(1)**, **notmuch-dump(1)**, **notmuch-hooks(5)**, **notmuch-insert(1)**, **notmuch-new(1)**, **notmuch-properties(7)**, **notmuch-reindex(1)**, **notmuch-reply(1)**, **notmuch-restore(1)**, **notmuch-search(1)**, **notmuch-show(1)**, **notmuch-tag(1)**

AUTHOR

Carl Worth and many others

COPYRIGHT

2009-2022, Carl Worth and many others