

NAME

notmuch-show – show messages matching the given search terms

SYNOPSIS

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

DESCRIPTION

Shows all messages matching the search terms.

See **notmuch-search-terms(7)** for details of the supported syntax for <search-terms>.

The messages will be grouped and sorted based on the threading (all replies to a particular message will appear immediately after that message in date order). The output is not indented by default, but depth tags are printed so that proper indentation can be performed by a post-processor (such as the emacs interface to notmuch).

Supported options for **show** include

--entire-thread=(true|false)

If true, **notmuch show** outputs all messages in the thread of any message matching the search terms; if false, it outputs only the matching messages. For **--format=json** and **--format=sexp** this defaults to true. For other formats, this defaults to false.

--format=(text|json|sexp|mbox|raw)

text (default for messages)

The default plain-text format has all text-content MIME parts decoded. Various components in the output, (**message**, **header**, **body**, **attachment**, and MIME **part**), will be delimited by easily-parsed markers. Each marker consists of a Control-L character (ASCII decimal 12), the name of the marker, and then either an opening or closing brace, ('{' or '}'), to either open or close the component. For a multipart MIME message, these parts will be nested.

json The output is formatted with Javascript Object Notation (JSON). This format is more robust than the text format for automated processing. The nested structure of multipart MIME messages is reflected in nested JSON output. By default JSON output includes all messages in a matching thread; that is, by default, **--format=json** sets **--entire-thread**. The caller can disable this behaviour by setting **--entire-thread=false**. The JSON output is always encoded as UTF-8 and any message content included in the output will be charset-converted to UTF-8.

sexp The output is formatted as the Lisp s-expression (sexp) equivalent of the JSON format above. Objects are formatted as property lists whose keys are keywords (symbols preceded by a colon). True is formatted as **t** and both false and null are formatted as **nil**. As for JSON, the s-expression output is always encoded as UTF-8.

mbox All matching messages are output in the traditional, Unix mbox format with each message being prefixed by a line beginning with "From " and a blank line separating each message. Lines in the message content beginning with "From " (preceded by zero or more '>' characters) have an additional '>' character added. This reversible escaping is termed "mboxrd" format and described in detail here:

<http://homepage.ntlworld.com/jonathan.deboynepollard/FGA/mail-mbox-formats.html>

raw (default if --part is given)

Write the raw bytes of the given MIME part of a message to standard out. For this format, it is an error to specify a query that matches more than one message.

If the specified part is a leaf part, this outputs the body of the part after performing content transfer decoding (but no charset conversion). This is suitable for saving attachments, for example.

For a multipart or message part, the output includes the part headers as well as the body (including all child parts). No decoding is performed because multipart and message parts cannot have non-trivial content transfer encoding. Consumers of this may need to implement MIME decoding and similar functions.

—format=version=N

Use the specified structured output format version. This is intended for programs that invoke **notmuch(1)** internally. If omitted, the latest supported version will be used.

—part=N

Output the single decoded MIME part N of a single message. The search terms must match only a single message. Message parts are numbered in a depth-first walk of the message MIME structure, and are identified in the 'json', 'sexp' or 'text' output formats.

Note that even a message with no MIME structure or a single body part still has two MIME parts: part 0 is the whole message (headers and body) and part 1 is just the body.

—sort=(newest-first|oldest-first)

This option can be used to present results in either chronological order (**oldest-first**) or reverse chronological order (**newest-first**).

Only threads as a whole are reordered. Ordering of messages within each thread will not be affected by this flag, since that order is always determined by the thread's replies.

By default, results will be displayed in reverse chronological order, (that is, the newest results will be displayed first).

—verify

Compute and report the validity of any MIME cryptographic signatures found in the selected content (e.g., "multipart/signed" parts). Status of the signature will be reported (currently only supported with **—format=json** and **—format=sexp**), and the multipart/signed part will be replaced by the signed data.

—decrypt=(false|auto|true|stash)

If **true**, decrypt any MIME encrypted parts found in the selected content (e.g., "multipart/encrypted" parts). Status of the decryption will be reported (currently only supported with **—format=json** and **—format=sexp**) and on successful decryption the multipart/encrypted part will be replaced by the decrypted content.

stash behaves like **true**, but upon successful decryption it will also stash the message's session key in the database, and index the cleartext of the message, enabling automatic decryption in the future.

If **auto**, and a session key is already known for the message, then it will be decrypted, but notmuch will not try to access the user's keys.

Use **false** to avoid even automatic decryption.

Non-automatic decryption (**stash** or **true**, in the absence of a stashed session key) expects a functioning **gpg-agent(1)** to provide any needed credentials. Without one, the decryption will fail.

Note: setting either **true** or **stash** here implies **—verify**.

Here is a table that summarizes each of these policies:

	false	auto	true	stash
--	-------	------	------	-------

Show cleartext if session key is already known		X	X	X
Use secret keys to show cleartext			X	X
Stash any newly recovered session keys, reindexing message if found				X

Note: **---decrypt=stash** requires write access to the database. Otherwise, **notmuch show** operates entirely in read-only mode.

Default: **auto**

---exclude=(true|false)

Specify whether to omit threads only matching `search.exclude_tags` from the search results (the default) or not. In either case the excluded message will be marked with the exclude flag (except when `output=mbox` when there is nowhere to put the flag).

If **---entire-thread** is specified then complete threads are returned regardless (with the excluded flag being set when appropriate) but threads that only match in an excluded message are not returned when **---exclude=true**.

The default is **---exclude=true**.

---body=(true|false)

If true (the default) **notmuch show** includes the bodies of the messages in the output; if false, bodies are omitted. **---body=false** is only implemented for the `text`, `json` and `sexp` formats and it is incompatible with **---part > 0**.

This is useful if the caller only needs the headers as body-less output is much faster and substantially smaller.

---include-html

Include "text/html" parts as part of the output (currently only supported with **---format=text**, **---format=json** and **---format=sexp**). By default, unless **---part=N** is used to select a specific part or **---include-html** is used to include all "text/html" parts, no part with content type "text/html" is included in the output.

A common use of **notmuch show** is to display a single thread of email messages. For this, use a search term of "thread:<thread-id>" as can be seen in the first column of output from the **notmuch-search(1)** command.

CONFIGURATION

Structured output (`json` / `sexp`) is influenced by the configuration option `show.extra_headers`. See **notmuch-config(1)** for details.

EXIT STATUS

This command supports the following special exit status codes

- 20** The requested format version is too old.
- 21** The requested format version is too new.

SEE ALSO

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

AUTHOR

Carl Worth and many others

COPYRIGHT

2009-2022, Carl Worth and many others