NAME

notmuch-properties − notmuch message property conventions and documentation

SYNOPSIS

notmuch count property:<*key*>=<*value*>

notmuch search property:<*key*>=<*value*>

notmuch show property:<*key*>=<*value*>

notmuch reindex property:<*key*>=<*value*>

notmuch tag +<*tag*> **property:**<*key*>=<*value*>

notmuch dump −−include=properties

notmuch restore −−include=properties

DESCRIPTION

Several notmuch commands can search for, modify, add or remove properties associated with specific messages. Properties are key/value pairs, and a message can have more than one key/value pair for the same key.

While users can select based on a specific property in their search terms with the prefix **property:**, the notmuch command−line interface does not provide mechanisms for modifying properties directly to the user.

Instead, message properties are expected to be set and used programmatically, according to logic in notmuch itself, or in extensions to it.

Extensions to notmuch which make use of properties are encouraged to report the specific properties used to the upstream notmuch project, as a way of avoiding collisions in the property namespace.

CONVENTIONS

Any property with a key that starts with "index." will be removed (and possibly re−set) upon reindexing (see **notmuch−reindex(1)**).

MESSAGE PROPERTIES

The following properties are set by notmuch internally in the course of its normal activity.

index.decryption

If a message contains encrypted content, and notmuch tries to decrypt that content during indexing, it will add the property **index.decryption=success** when the cleartext was successfully indexed. If notmuch attempts to decrypt any part of a message during indexing and that decryption attempt fails, it will add the property **index.decryption=failure** to the message.

Note that it's possible for a single message to have both **index.decryption=success** and **index.decryption=failure**. Consider an encrypted e−mail message that contains another encrypted e−mail message as an attachment −− if the outer message can be decrypted, but the attached part cannot, then both properties will be set on the message as a whole.

If notmuch never tried to decrypt an encrypted message during indexing (which is the default, see **index.decrypt** in **notmuch−config(1)**), then this property will not be set on that message.

session−key

When **notmuch−show(1)** or **notmuch−reply(1)** encounters a message with an encrypted part, if notmuch finds a **session−key** property associated with the message, it will try that stashed session key for decryption.

If you do not want to use any stashed session keys that might be present, you should pass those programs **−−decrypt=false**.

Using a stashed session key with "notmuch show" will speed up rendering of long encrypted threads. It also allows the user to destroy the secret part of any expired encryption−capable subkey while still being able to read any retained messages for which they have stashed the session key. This enables truly deletable e−mail, since (once the session key and asymmetric subkey are both destroyed) there are no keys left that can be used to decrypt any copy of the original message previously stored by an adversary.

However, access to the stashed session key for an encrypted message permits full byte−for−byte reconstruction of the cleartext message. This includes attachments, cryptographic signatures, and other material that cannot be reconstructed from the index alone.

See **index.decrypt** in **notmuch−config(1)** for more details about how to set notmuch's policy on when to store session keys.

The session key should be in the ASCII text form produced by GnuPG. For OpenPGP, that consists of a decimal representation of the hash algorithm used (identified by number from RFC 4880, e.g. 9 means AES−256) followed by a colon, followed by a hexadecimal representation of the algorithm−specific key. For example, an AES−128 key might be stashed in a notmuch property as: **session−key=7:14B16AF65536C28AF209828DFE34C9E0**.

index.repaired

Some messages arrive in forms that are confusing to view; they can be mangled by mail transport agents, or the sending mail user agent may structure them in a way that is confusing. If notmuch knows how to both detect and repair such a problematic message, it will do so during indexing.

If it applies a message repair during indexing, it will use the **index.repaired** property to note the type of repair(s) it performed.

index.repaired=skip−protected−headers−legacy−display indicates that when indexing the cleartext of an encrypted message, notmuch skipped over a "legacy−display" text/rfc822−headers part that it found in that message, since it was able to index the built−in protected headers directly.

index.repaired=mixedup indicates the repair of a "Mixed Up" encrypted PGP/MIME message, a mangling typically produced by Microsoft's Exchange MTA. See *https://tools.ietf.org/html/draft−dkg−openpgp−pgpmime−message−mangling* for more information.

SEE ALSO

notmuch(1), **notmuch−config(1)**, **notmuch−dump(1)**, **notmuch−insert(1)**, **notmuch−new(1)**, **notmuch−reindex(1)**, **notmuch−reply(1)**, **notmuch−restore(1)**, **notmuch−search−terms(7)**, **notmuch−show(1)**

AUTHOR

Carl Worth and many others

COPYRIGHT