NAME

lmdb_table - Postfix LMDB adapter

SYNOPSIS

postmap lmdb:/etc/postfix/*filename*
postmap -i lmdb:/etc/postfix/*filename* <*inputfile*

postmap -d *key* **lmdb:/etc/postfix/***filename*
postmap -d - lmdb:/etc/postfix/*filename* <*inputfile*

postmap -q *key* **lmdb:/etc/postfix/***filename*
postmap -q - lmdb:/etc/postfix/*filename* <*inputfile*

DESCRIPTION

The Postfix LMDB adapter provides access to a persistent, memory-mapped, key-value store. The database size is limited only by the size of the memory address space (typically 31 or 47 bits on 32-bit or 64-bit CPUs, respectively) and by the available file system space.

REQUESTS

The LMDB adapter supports all Postfix lookup table operations. This makes LMDB suitable for Postfix address rewriting, routing, access policies, caches, or any information that can be stored under a fixed lookup key.

When a transaction fails due to a full database, Postfix resizes the database and retries the transaction.

Postfix table lookups may generate partial search keys such as domain names without one or more subdomains, network addresses without one or more least-significant octets, or email addresses without the localpart, address extension or domain portion. This behavior is also found with, for example, btree:, hash:, or ldap: tables.

Unlike other flat-file Postfix databases, changes to an LMDB database do not trigger automatic daemon program restart, and do not require **postfix reload**.

RELIABILITY

LMDB's copy-on-write architecture provides safe updates, at the cost of using more space than some other flat-file databases. Read operations are memory-mapped for speed. Write operations are not memory-mapped to avoid silent curruption due to stray pointer bugs.

Multiple processes can safely update an LMDB database without serializing requests through the proxymap(8) service. This makes LMDB suitable as a shared cache for verify(8) or postscreen(8) services.

SYNCHRONIZATION

The Postfix LMDB adapter does not use LMDB's built-in locking scheme, because that would require world-writable lockfiles and would violate the Postfix security model. Instead, Postfix uses fcntl(2) locks with whole-file granularity. Programs that use LMDB's built-in locking protocol will corrupt a Postfix LMDB database or will read garbage.

Every Postfix LMDB database read or write transaction must be protected from start to end with a shared or exclusive fcntl(2) lock. A writer may atomically downgrade an exclusive lock to a shared lock, but it must acquire an exclusive lock between updating the database and starting another write transaction.

Note that fcntl(2) locks do not protect transactions within the same process against each other. If a program cannot avoid making simultaneous database requests, then it must protect its transactions with in-process locks, in addition to the per-process fcntl(2) locks.

CONFIGURATION PARAMETERS

Short-lived programs automatically pick up changes to main.cf. With long-running daemon programs, Use the command **postfix reload** after a configuration change.

lmdb_map_size (default: 16777216)
The initial LMDB database size limit in bytes.

SEE ALSO

postconf(1),
Postfix supported lookup tables
postmap(1),
Postfix lookup table maintenance
postconf(5),
configuration parameters

README FILES

Use **postconf readme_directory** or **postconf html_directory** to locate this information.
DATABASE_README, Postfix lookup table overview
LMDB_README, Postfix OpenLDAP LMDB howto

LICENSE

HISTORY

LMDB support was introduced with Postfix version 2.11.

AUTHOR(S)

Howard Chu
Symas Corporation

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA