NAME
> memcache_table – Postfix memcache client configuration

SYNOPSIS
> **postmap −q "***string***" memcache:/etc/postfix/***filename*
>
> **postmap −q − memcache:/etc/postfix/***filename* <*inputfile*

DESCRIPTION
> The Postfix mail system uses optional tables for address rewriting or mail routing. These tables are usually in **dbm** or **db** format.
>
> Alternatively, lookup tables can be specified as memcache instances. To use memcache lookups, define a memcache source as a lookup table in main.cf, for example:
>
> > virtual_alias_maps = memcache:/etc/postfix/memcache−aliases.cf
>
> The file /etc/postfix/memcache−aliases.cf has the same format as the Postfix main.cf file, and specifies the parameters described below.
>
> The Postfix memcache client supports the lookup, update, delete and sequence (first/next) operations. The sequence operation requires a backup database that supports the operation.

MEMCACHE MAIN PARAMETERS
> **memcache (default: inet:localhost:11211)**
> > The memcache server (note: singular) that Postfix will try to connect to. For a TCP server specify "inet:" followed by a hostname or address, ":", and a port name or number. Specify an IPv6 address inside "[]". For a UNIX−domain server specify "unix:" followed by the socket pathname. Examples:
> >
> > > memcache = inet:memcache.example.com:11211
> > > memcache = inet:127.0.0.1:11211
> > > memcache = inet:[fc00:8d00:189::3]:11211
> > > memcache = unix:/path/to/socket
> >
> > NOTE: to access a UNIX−domain socket with the proxymap(8) server, the socket must be accessible by the unprivileged postfix user.
>
> **backup (default: undefined)**
> > An optional Postfix database that provides persistent backup for the memcache database. The Postfix memcache client will update the memcache database whenever it looks up or changes information in the persistent database. Specify a Postfix "type:table" database. Examples:
> >
> > > # Non−shared postscreen cache.
> > > backup = btree:/var/lib/postfix/postscreen_cache_map
> > >
> > > # Shared postscreen cache for processes on the same host.
> > > backup = proxy:btree:/var/lib/postfix/postscreen_cache_map
> >
> > Access to remote proxymap servers is under development.
> >
> > NOTE 1: When sharing a persistent **postscreen**(8) or **verify**(8) cache, disable automatic cache cleanup (set *_cache_cleanup_interval = 0) except with one Postfix instance that will be responsible for cache cleanup.
> >
> > NOTE 2: When multiple tables share the same memcache database, each table should use the **key_format** feature (see below) to prepend its own unique string to the lookup key. Otherwise,

automatic **postscreen**(8) or **verify**(8) cache cleanup may not work.

NOTE 3: When the backup database is accessed with "proxy:" lookups, the full backup database name (including the "proxy:" prefix) must be specified in the proxymap server's proxy_read_maps or proxy_write_maps setting (depending on whether the access is read−only or read−write).

flags (default: 0)

Optional flags that should be stored along with a memcache update. The flags are ignored when looking up information.

ttl (default: 3600)

The expiration time in seconds of memcache updates.

NOTE 1: When using a memcache table as **postscreen**(8) or **verify**(8) cache without persistent backup, specify a zero *_cache_cleanup_interval value with all Postfix instances that use the memcache, and specify the largest **postscreen**(8) *_ttl value or **verify**(8) *_expire_time value as the memcache table's **ttl** value.

NOTE 2: According to memcache protocol documentation, a value greater than 30 days (2592000 seconds) specifies absolute UNIX time. Smaller values are relative to the time of the update.

MEMCACHE KEY PARAMETERS

key_format (default: %s)

Format of the lookup and update keys that the Postfix memcache client sends to the memcache server. By default, these are the same as the lookup and update keys that the memcache client receives from Postfix applications.

NOTE 1: The **key_format** feature is not used for **backup** database requests.

NOTE 2: When multiple tables share the same memcache database, each table should prepend its own unique string to the lookup key. Otherwise, automatic **postscreen**(8) or **verify**(8) cache cleanup may not work.

Examples:

    key_format = aliases:%s
    key_format = verify:%s
    key_format = postscreen:%s

The **key_format** parameter supports the following '%' expansions:

%%   This is replaced by a literal '%' character.

%s   This is replaced by the memcache client input key.

%u   When the input key is an address of the form user@domain, **%u** is replaced by the SQL quoted local part of the address. Otherwise, **%u** is replaced by the entire search string. If the localpart is empty, a lookup is silently suppressed and returns no results (an update is skipped with a warning).

%d   When the input key is an address of the form user@domain, **%d** is replaced by the domain part of the address. Otherwise, a lookup is silently suppressed and returns no results (an update is skipped with a warning).

%[SUD]

The upper−case equivalents of the above expansions behave in the **key_format** parameter identically to their lower−case counter−parts.

%[1−9]

>The patterns %1, %2, ... %9 are replaced by the corresponding most significant component of the input key's domain. If the input key is *user@mail.example.com*, then %1 is **com**, %2 is **example** and %3 is **mail**. If the input key is unqualified or does not have enough domain components to satisfy all the specified patterns, a lookup is silently suppressed and returns no results (an update is skipped with a warning).

domain (default: no domain list)

>This feature can significantly reduce database server load. Specify a list of domain names, paths to files, or "type:table" databases. When specified, only fully qualified search keys with a *non−empty* localpart and a matching domain are eligible for lookup or update: bare 'user' lookups, bare domain lookups and "@domain" lookups are silently skipped (updates are skipped with a warning). Example:

>domain = example.com, hash:/etc/postfix/searchdomains

MEMCACHE ERROR CONTROLS

data_size_limit (default: 10240)

>The maximal memcache reply data length in bytes.

line_size_limit (default: 1024)

>The maximal memcache reply line length in bytes.

max_try (default: 2)

>The number of times to try a memcache command before giving up. The memcache client does not retry a command when the memcache server accepts no connection.

retry_pause (default: 1)

>The time in seconds before retrying a failed memcache command.

timeout (default: 2)

>The time limit for sending a memcache command and for receiving a memcache reply.

BUGS

The Postfix memcache client cannot be used for security−sensitive tables such as **alias_maps** (these may contain "/*command* and "*/file/name*" destinations), or **virtual_uid_maps**, **virtual_gid_maps** and **virtual_mailbox_maps** (these specify UNIX process privileges or "*/file/name*" destinations). In a typical deployment a memcache database is writable by any process that can talk to the memcache server; in contrast, security−sensitive tables must never be writable by the unprivileged Postfix user.

The Postfix memcache client requires additional configuration when used as **postscreen**(8) or **verify**(8) cache. For details see the **backup** and **ttl** parameter discussions in the MEMCACHE MAIN PARAMETERS section above.

SEE ALSO

postmap(1), Postfix lookup table manager
postconf(5), configuration parameters

README FILES

Use "**postconf readme_directory**" or "**postconf html_directory**" to locate this information.
DATABASE_README, Postfix lookup table overview
MEMCACHE_README, Postfix memcache client guide

LICENSE

The Secure Mailer license must be distributed with this software.

HISTORY

Memcache support was introduced with Postfix version 2.9.

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research

P.O. Box 704
Yorktown Heights, NY 10598, USA