ConfigurationVariables

Table of Contents

ConfigurationVariables

(NOTE: This page is auto-generated, do not edit by hand)

TMDA Configuration Variables

- Set TMDA configuration variables in /etc/tmdarc or ~/.tmda/config only. Settings in ~/.tmda/config override those in /etc/tmdarc.
- Each variable has a default value, and in most cases that will suffice. You only need to set a variable in your configuration file if you want to override the default value.
- /etc/tmdarc and ~/.tmda/config are interpreted as executable Python code. Thus, variable declarations must adhere to proper Python syntax. Each variable definition below contains one or more examples.

Index

ACTION_EXPIRED_DATED

Specifies how incoming messages should be disposed of if they are sent to an expired dated address. You can specify a single action to take by setting this variable to a string, in which case possible values include:

`"bounce"`

bounce the message - uses the bounce_expired_dated.txt template.

`"drop"`

silently drop the message

`"ok"`

deliver the message

`"confirm"`

request confirmation for the message

`"hold"`

silently hold message in pending queue

This can also be a dictionary if you want to handle different ages of expired dated messages in different ways.

Examples:

```
ACTION_EXPIRED_DATED = {

      'default':'confirm', # default is to confirm, unless
       '1w': 'bounce', # ...it expired more than 1w ago, then
      bounce
       '30d': 'hold', # ...it expired more than 30d ago, then
      hold
       '1Y': 'drop'} # ...it expired more than 1Y ago, then
      drop
```

Default is "confirm"

ACTION_FAIL_DATED

Specifies how incoming messages should be disposed of if they are sent to a dated address that does not properly verify. Possible values include:

`"bounce"`

bounce the message - uses the bounce_fail_dated.txt template.

`"drop"`

silently drop the message

`"ok"`

deliver the message

`"confirm"`

request confirmation for the message

`"hold"`

silently hold message in pending queue

Default is confirm

ACTION_FAIL_KEYWORD

Specifies how incoming messages should be disposed of if they are sent to a keyword address that fails to verify. Possible values include:

`"bounce"`

bounce the message - uses the bounce_fail_keyword.txt template.

`"drop"`

silently drop the message

`"ok"`

deliver the message

`"confirm"`

request confirmation for the message

`"hold"`

silently hold message in pending queue

Default is confirm

ACTION_FAIL_SENDER

Specifies how incoming messages should be disposed of if they are sent to a sender address, but were not sent from the correct sender, and fail to verify. Possible values include:

`"bounce"`

bounce the message - uses the bounce_fail_sender.txt template.

`"drop"`

silently drop the message

`"ok"`

deliver the message

`"confirm"`

request confirmation for the message

`"hold"`

silently hold message in pending queue

Default is confirm

ACTION_HEADER_INCOMING

Set this variable to True if you want TMDA to add an `X-TMDA-Action' header to your delivered incoming messages. The value of this header is the same as the "Actn:" field in a LOGFILE_INCOMING entry. e.g,

X-TMDA-Action: OK (from johndoe* ok)

NOTE: This will not work if you are running qmail and have not set the DELIVERY variable.

Default is False (turned off)

ACTION_INCOMING

Specifies how incoming messages should be disposed of by default if they didn't match FILTER_INCOMING and were not sent to a tagged address. Possible values include:

`"bounce"`

bounce the message - uses the bounce_incoming.txt template.

`"drop"`

silently drop the message

`"ok"`

deliver the message

`"confirm"`

request confirmation for the message

`"hold"`

silently hold message in pending queue

Default is confirm

ACTION_INVALID_CONFIRMATION

Specifies how confirmation messages should be disposed of if they are sent to a confirmation address that fails to verify. Possible values include:

`"bounce"`

bounce the message - uses the bounce_invalid_confirmation.txt template.

`"drop"`

silently drop the message

`"ok"`

deliver the message

`"confirm"`

request confirmation for the message

`"hold"`

silently hold message in pending queue

Default is bounce

ACTION_MISSING_PENDING

Specifies how confirmation messages should be disposed of if the message to be confirmed can not be located on disk. This might be the case if you hand-released the message before the sender tried to confirm it by e-mail, or if the sender sent multiple confirmation replies. Possible values include:

`"bounce"`

bounce the message - uses the bounce_missing_pending.txt template.

`"drop"`

silently drop the message

`"ok"`

deliver the message

`"confirm"`

request confirmation for the message

`"hold"`

silently hold message in pending queue

Default is bounce

ACTION_OUTGOING

Specifies how outgoing messages should be tagged by default if there are no matches in FILTER_OUTGOING and no X-TMDA header. Possible values include:

`"bare[=append]"`

don't tag

`"dated[=timeout_interval]"`

tag with a dated address

`"sender[=sender_address]"`

tag with a sender address

`"exp=full_address"`

tag with explicit address

```
"kw=keyword"
```

tag with a keyword address

Default is dated

ADDED_HEADERS_CLIENT

A Python dictionary containing one or more header:value string pairs that should be added to _all_ outgoing client-side messages (i.e, messages sent with tmda-sendmail) prior to injection. Listed headers and their values are case-sensitive.

As the examples below illustrate, the full power of Python is available to create these headers -- just make sure both the header and its value end up as strings.

Examples:

```
ADDED_HEADERS_CLIENT = {"X-Fact" : "Father Hennepin Discovered
Niagra Falls"}
import time, os, random
ADDED_HEADERS_CLIENT = {

     "X-Localtime" : time.asctime(),
      "X-Favorite-Author" : "James Joyce",
      "X-OperatingSystem" : os.uname()[0],
      "Organization" : os.environ.get("ORGANIZATION"),
      "X-Uptime" : os.popen("/usr/bin/uptime").read().strip(),
      "X-Now-Reading" : random.choice(open
      ("/home/jasonrm/.now-reading").
      readlines()).strip()
      }
```

No default

ADDED_HEADERS_SERVER

A Python dictionary containing one or more header:value string pairs that should be added to _all_ outgoing server-side messages (i.e, messages sent with tmda-filter) prior to injection. Listed headers and their values are case-sensitive.

See ADDED_HEADERS_CLIENT (above) for some examples.

No default

ALLOW_MODE_640

Set this variable to True if you want to allow a mode 640 CRYPT_KEY_FILE.

Default is False (turned off)

AUTORESPONSE_INCLUDE_SENDER_COPY

An integer which controls whether a copy of the sender's message is included or not when sending an auto response. Available options:

0 - do not include any portion of the sender's message. 1 - include only the headers from the sender's message. 2 - include the sender's entire message.

2 is *highly* recommended. Not only to give the sender a clear indication of which message is being responded to, but also to preserve the message for the sender in case she didn't save a copy when sending it.

Default is 2

BARE_APPEND

Filename to which a recipient's e-mail address should be automatically appended if the outgoing <action> is in the form 'bare=append'.

Examples:

```
BARE_APPEND = "/full/path/to/whitelist"
BARE_APPEND = "~/.tmda/lists/whitelist"
```

No default

BOUNCE_ENV_SENDER

The envelope sender of a bounce message.

Example:

```
BOUNCE_ENV_SENDER = "devnull@domain.dom"
```

Default is "<>", a null envelope sender.

CGI_SETTINGS

Filename for saving various tmda-cgi settings. This filename can include a full path, otherwise it is relative the user's config file.

Examples:

```
CGI_SETTINGS = "/home/jim/.tmda/MyCGISettings"
CGI_SETTINGS = "~/.tmda/MyCGISettings"
```

Default is "tmda-cgi.ini".

CGI_URL

Absolute URL of your tmda-cgi installation.

Example:

```
CGI_URL = "http://www.domain.dom/cgi-bin/tmda.cgi"
```

No default.

CGI_VIRTUALUSER

Set this variable to True if you wish to enable tmda-cgi's "virtual user" support (http://tmda.net/tmda-cgi/virtual.html).

Default is False (disabled)

CONFIG_EXEC

If set to False in GLOBAL_TMDARC, the user's TMDARC file will be parsed using ConfigParser, otherwise it will evaluated as a sequence of Python statements using execfile().

Default is True (use execfile())

CONFIRM_ACCEPT_CC

An optional e-mail address which will be sent a copy of the confirmation acceptance messages people send you.

Example:

```
CONFIRM_ACCEPT_CC = "jdoe-confirm-replies@domain.dom"
```

No default.

CONFIRM_ACCEPT_NOTIFY

Set this variable to False if you do not want to generate any confirmation acceptance notices. These are the notices returned to senders when they confirm their original message by e-mail. Their content is based on the confirm_accept.txt template.

Default is True (turned on)

CONFIRM_ADDRESS

An optional e-mail address to use for creating confirmation addresses. Normally, the recipient address is used, but in some virtual domain or forwarding scenarios, you may wish to have confirmation address be based on a fixed address regardless of the recipient address.

Example:

```
CONFIRM_ADDRESS = "webmaster@domain.dom"
```

No default

CONFIRM_APPEND

Filename to which a sender's e-mail address should be automatically appended once they confirm a message. This can be used to implement "auto-whitelisting" functionality.

Examples:

```
CONFIRM_APPEND = "/full/path/to/whitelist"
CONFIRM_APPEND = "~/.tmda/lists/whitelist"
```

No default

CONFIRM_CC

An optional e-mail address which will be sent a copy of any message that triggers a confirmation request.

Example:

```
CONFIRM_CC = "jdoe-confirms@domain.dom"
```

No default.

CONFIRM_MAX_MESSAGE_SIZE

This is the largest size (in bytes) that a message can be before the its body is excluded from the confirmation request/acceptance notice. Set this to None to allow any size message.

Default is 50000

CRYPT_KEY_FILE

File which contains your unique TMDA secret key generated by the `tmda-keygen' program. The key should be unquoted in the file. This file must be chmod 400 or 600, unless ALLOW_MODE_640 is on.

Default is ~/.tmda/crypt_key

DATADIR

Top-level directory which TMDA uses to store its files and directories. TMDA should be free to create files and directories under DATADIR if need be.

Examples:

```
DATADIR = "/full/path/to/.tmda"
DATADIR = "~/.tmda"
```

Default is ~/.tmda

DATED_TEMPLATE_VARS

Set this variable to True if you want to use 'dated' address variables in your templates.

Default is False (turned off)

DATED_TIMEOUT

The timeout interval for 'dated' addresses. The available units are (Y=years, M=months, w=weeks, d=days, h=hours, m=minutes, s=seconds).

Default is 5d (5 days).

DB_BARE_APPEND

SQL INSERT statement to be used to insert recipient addresses into a SQL database if the outgoing <action> was 'bare=append'. The Python DB API will take care of properly quoting parameters that are strings. Requires a valid DB_CONNECTION object.

Available substition parameters are:

%(recipient)s - recipient's email address %(username)s - USERNAME (of TMDA user) %(hostname)s - HOSTNAME (of TMDA user) %(sender)s - USERNAME@HOSTNAME (address of TMDA user) %(fromheader)s - address of TMDA user in From: header field

Examples:

```
DB_BARE_APPEND = """

      INSERT INTO whitelist (user_email, address)
       VALUES (%(sender)s, %(recipient)s)"""
```

Default is None

DB_CONFIRM_APPEND

SQL INSERT statement to be used to insert confirmed sender addresses into a SQL database. The Python DB API will take care of properly quoting parameters that are strings. Requires a valid DB_CONNECTION object.

Available substition parameters are:

%(recipient)s - USERNAME@HOSTNAME %(username)s - USERNAME %(hostname)s - HOSTNAME %(sender)s - sender's address (envelope sender or X-Primary-Address)

Examples:

```
DB_CONFIRM_APPEND = """

      INSERT INTO whitelist (user_email, address)
       VALUES (%(recipient)s, %(sender)s)"""
```

DB_CONFIRM_APPEND = """

INSERT INTO wildcard_list (uid, address, action) SELECT uid, %(sender)s, 'accept' FROM users WHERE users.email = %(recipient)s"""

Default is None

DB_CONNECTION

Python DB API Connection object. This is specific to the database and should be created in the ~/.tmda/config file. This is typically created by importing the appropriate module and calling the connect() function.

Example:

```
import MySQLdb
DB_CONNECTION = MySQLdb.connect("...")
```

Default is None

DELIVERY

The default delivery instruction for successful messages, or TMDA's final delivery location. Only required if you are NOT running qmail. The default value for qmail users is "_qok_" which means exit(0) and proceed to the next instruction in the dot-qmail file.

Delivery to qmail-style Maildirs, mboxrd-format mboxes, programs (pipe), and forward to an e-mail addresses are supported.

Acceptable syntax and restrictions for delivery instructions are discussed in the ``action'' section of the TMDA Filter Specification (config-filter.html). Please read this documentation.

Examples:

```
DELIVERY = "~/Maildir/"
DELIVERY = "~/Mailbox"
DELIVERY = "/var/mail/jasonrm"
DELIVERY = ":~/Mailbox"
DELIVERY = ":/var/spool/mail/jasonrm"
DELIVERY = "|/usr/bin/maildrop"
DELIVERY = "|/usr/bin/procmail ~/.procmailrc-tmda"
DELIVERY = "me@new.job.com"
```

No default for non-qmail users.

FILTER_BOUNCE_CC

An optional e-mail address which will be sent a copy of any message that bounces because of a match in FILTER_INCOMING.

Example:

```
FILTER_BOUNCE_CC = "jdoe-bounces@domain.dom"
```

No default.

FILTER_DROP_CC

An optional e-mail address which will be sent a copy of any message that is dropped because of a match in FILTER_INCOMING.

Example:

```
FILTER_DROP_CC = "jdoe-drops@domain.dom"
```

No default.

FILTER_INCOMING

Filter file which controls how incoming messages are tagged. Look for the filter-file in the environment first.

Default is ~/.tmda/filters/incoming

FILTER_OUTGOING

Filter file which controls how outgoing messages are tagged. Look for the filter-file in the environment first.

Default is ~/.tmda/filters/outgoing

FINGERPRINT

A list containing one or more message headers whose values should be used to create a "fingerprint" for the message. If the header value is 'body' (all-lowercase), the message body content is used instead of a header value. The fingerprint is a SHA-1 HMAC digest represented as a base64-encoded string. This fingerprint will be added to your outgoing client-side messages (i.e, messages sent with tmda-sendmail) in an `X-TMDA-Fingerprint' header prior to injection.

Examples:

```
FINGERPRINT = ["message-id"]
FINGERPRINT = ["message-id", "from", "date"]
FINGERPRINT = ["message-id", "body"]
```

Things to keep in mind, especially if verifying these fingerprints with non-TMDA code.

- ◊ CRYPT_KEY is converted from hex into raw binary before it is used to create the HMAC object.
- ◊ The order of header names in the FINGERPRINT list is important; updates are made in the order listed.
- ◊ If a listed header doesn't exist in the message, no update attempt will be made for that header.
- ◊ For a listed header, only the header value (i.e, text after the colon) is used. Additionally, leading and trailing whitespace (but not internal whitespace) is stripped from that value.

For example, if the listed headers consisted of:

Message-ID: <20011212192455.A7060@nightshade.la.mastaler.com>
From: "Jason R. Mastaler" <jason-dated-1008901496.5356ec@mastaler.com>
Date: Wed, 12 Dec 2001 19:24:55 -0700

The strings which are used to create the HMAC digest are the header values only (right hand side of Header:).

The following header would then be added to the outgoing message:
X-TMDA-Fingerprint: vDBoOHtIUE6VniJguxJ+w2fR5bU

No default

FULLNAME

Your full name.

Example:

```
FULLNAME = "John Doe"
```

Default comes from your environment or the password file.

HMAC_BYTES

An integer which determines the length of the HMACs used in TMDA's "cookies". Read the `CRYPTO' file for more information. Changing this value will will invalidate all previously generated HMACs.

Default is 3 (24-bit HMACs)

HOSTNAME

The right-hand side of your email address (after `@'). Used only in cases where TMDA can't determine this itself.

Example:

```
HOSTNAME = "domain.dom"
```

Defaults to the fully qualified domain name of the localhost.

LOGFILE_DEBUG

Filename which uncaught exceptions should be written to.

Examples:

```
LOGFILE_DEBUG = "/path/to/logs/tmda.debug"
LOGFILE_DEBUG = "~/.tmda/logs/debug"
```

No default.

LOGFILE_INCOMING

Filename which incoming delivery (i.e, tmda-filter) summaries should be written to.

Examples:

```
LOGFILE_INCOMING = "/path/to/logs/tmda.in"
LOGFILE_INCOMING = "~/.tmda/logs/incoming"
```

No default.

LOGFILE_OUTGOING

Filename which outgoing message (i.e, tmda-sendmail) summaries should be written to.

Examples:

```
LOGFILE_OUTGOING = "/path/to/logs/tmda.out"
LOGFILE_OUTGOING = "~/.tmda/logs/outgoing"
```

No default.

MAIL_FOLLOWUP_TO

Inspired by qmail-inject's QMAILMFTFILE feature, MAIL_FOLLOWUP_TO automatically adds a Mail-Followup-To field for messages sent to mailing lists. It works for both tmda-sendmail and tmda-ofmipd.

MAIL_FOLLOWUP_TO can either be a list of mailing list addresses, or a string pointing to a file containing mailing list addresses, one per line. If To or Cc in the message includes one of those addresses (without regard to case), tmda-inject adds a Mail-Followup-To field with all the To+Cc addresses. tmda-inject does not add Mail-Followup-To to a message that already has one, or if the address file does not exist.

See http://cr.yp.to/proto/replyto.html for more on Mail-Followup-To.

Examples:

```
MAIL_FOLLOWUP_TO = "/path/to/lists.txt"
MAIL_FOLLOWUP_TO = "~/.lists"
MAIL_FOLLOWUP_TO = ["tmda-users@tmda.net",
"postfix-users@postfix.org"]
```

No default.

MAIL_TRANSFER_AGENT

Defines which mail transfer agent (MTA) software you are running. Possible choices are "exim", "postfix", "qmail", or "sendmail".

Default is qmail

MAIL_TRANSPORT

Final delivery method for all outgoing mail (server and client). Possible values include:

`"smtp"`

> Deliver messages by handing them off to an SMTP server (i.e, smarthost). Be sure that the "SMTPHOST" variable (see below) is set correctly.

`"sendmail"`

> Deliver messages via the command line interface to the sendmail program (e.g, /usr/sbin/sendmail).

Default is "sendmail"

MAX_AUTORESPONSES_PER_DAY

An integer specifying the maximum number of automatic responses sent to a given sender address in a day. This includes _all_ auto-responses sent by TMDA (confirmation requests, confirmation acceptance notices, failure notices, etc.)

This limit prevents response loops between TMDA and misconfigured remote auto-responders. TMDA already inhibits automatic replies to any message that looks like a mailing list message or a bounce message. This is a fallback safety valve so it should be set fairly high. Set to 0 for no limit.

Default is 50

MESSAGE_FROM_STYLE

Specifies how `From' and `Resent-From' headers should look when tagging outgoing messages with tmda-sendmail. There are two valid values:

```
"address"
```

Just the address - king@grassland.com

```
"angles"
```

Elvis Parsley <king@grassland.com>

Default is "angles".

MESSAGE_TAG_HEADER_STYLE

Specifies how headers (other than `From' and `Resent-From') should look when tagging outgoing messages with tmda-sendmail. These are the headers such as Reply-To which are defined using the `tag' action in your FILTER_OUTGOING file.

The valid values and default value is identical to that of MESSAGE_FROM_STYLE.

PENDING_BLACKLIST_APPEND

Filename to which a sender's e-mail address should be appended when a message is "blacklisted" by tmda-pending.

Examples:

```
PENDING_BLACKLIST_APPEND = "/full/path/to/blacklist"
PENDING_BLACKLIST_APPEND = "~/.tmda/lists/blacklist"
```

No default

PENDING_CACHE

Path to the cache file used when tmda-pending is invoked with the --cache option.

Default is ~/.tmda/.pendingcache

PENDING_CACHE_LEN

An integer which specifies the maximum number of entries held by PENDING_CACHE. Make sure this is greater than the number of messages kept in your pending queue, or else you'll start seeing previously viewed messages again.

Default is 5000

PENDING_CLEANUP_ODDS

A floating point number which describes the odds that tmda-filter will automatically clean the pending queue of expired messages upon receipt of an incoming message. The lifetime of a message in the pending queue is controlled by the PENDING_LIFETIME setting.

If you wish to disable this feature in order to clean the queue by hand, or through cron, set this value to 0.0. If you wish to trigger a cleanup every time a message arrives, set it to 1.0.

The closer this value gets to 1.0, the fewer messages you'll have in your pending queue beyond PENDING_LIFETIME, but at some additional expense upon delivery.

Examples:

```
PENDING_CLEANUP_ODDS = 0.0 # 0% chance
PENDING_CLEANUP_ODDS = 1.0 # 100% chance
PENDING_CLEANUP_ODDS = 0.025 # 2.5% chance
```

Default is 0.01, or 1% chance of cleanup for every message received, or cleanup approximately once per 100 messages received.

PENDING_DELETE_APPEND

Filename to which a sender's e-mail address should be automatically appended when a message is "deleted" by tmda-pending. tmda-filter's automated pending queue cleanup feature (see PENDING_CLEANUP_ODDS) also respects this setting.

Examples:

```
PENDING_DELETE_APPEND = "/full/path/to/blacklist"
PENDING_DELETE_APPEND = "~/.tmda/lists/blacklist"
```

No default

PENDING_DIR

Full path to the directory containing messages pending confirmation (aka, the "pending queue"). If this directory doesn't exist, it will automatically be created by TMDA with 0700 permissions when the first message arrives.

Default is ~/.tmda/pending/

PENDING_LIFETIME

A time interval describing how long a message can live in the pending queue before it's subject to automated deletion by tmda-filter (a feature controlled by the PENDING_CLEANUP_ODDS setting).

Time intervals can be expressed in seconds (s), minutes (m), hours (h), days (d), weeks (w), months (M), or years (Y).

Examples:

```
PENDING_LIFETIME = "24h" # messages can live for 24 hours
PENDING_LIFETIME = "1M" # messages can live for 1 month
```

Default is 14d (14 day lifetime)

PENDING_QUEUE_FORMAT

A string specifying the format of TMDA's pending queue where unconfirmed messages are stored. Possible values include:

`"original"`

> A custom TMDA format where messages are stored one per file in a directory (PENDING_DIR). Offers high performance, but can only be browsed with TMDA tools like 'tmda-pending' and 'tmda-cgi'.

`"maildir"`

> Maildir is a specific one-file-per-message organization that was introduced with the qmail system by D.J. Bernstein. For more information, see http://wiki.tmda.net/TmdaPendingAsMaildir

Default is "original".

PENDING_RELEASE_APPEND

Filename to which a sender's e-mail address should be automatically appended when a message is "released" by tmda-pending.

Examples:

```
PENDING_RELEASE_APPEND = "/full/path/to/whitelist"
PENDING_RELEASE_APPEND = "~/.tmda/lists/whitelist"
```

No default

PENDING_WHITELIST_APPEND

Filename to which a sender's e-mail address should be appended when a message is "whitelisted" by tmda-pending.

Examples:

```
PENDING_WHITELIST_APPEND = "/full/path/to/whitelist"
PENDING_WHITELIST_APPEND = "~/.tmda/lists/whitelist"
```

No default

PENDING_WHITELIST_RELEASE

An option detailing the action taken when 'Whitelist' is the current action in tmda-pending or tmda-cgi

Available options:

0 - 'Whitelist' does not release any messages, only appends

> the envelope sender to PENDING_WHITELIST_APPEND

1 - 'Whitelist' releases the current message and also appends

> the envelope sender to PENDING_WHITELIST_APPEND

Default is 1

PRIMARY_ADDRESS_MATCH

An integer which controls how closely the address in the ``X-Primary-Address'' header of an incoming messages must match the envelope sender address before it's honored.

If the match is close enough, this address will be used for CONFIRM_APPEND instead of the envelope sender, and also added to the list of addresses checked against FILTER_INCOMING.

This option is available to limit cases of abuse where a sender attempts to "whitelist" an address not his own by using an external address in an ``X-Primary-Address'' header.

Available options:

0 - Never a match. Equivalent to disabling X-Primary-Address recognition.

1 - Identical addresses match. e.g, king@grassland.com and king@grassland.com.

2 - Usernames and hostnames must match. e.g, king@grassland.com and king-dated-1037839131.65d080@grassland.com.

3 - Usernames and domains must match. e.g, king@grassland.com and king-dated-1037839131.65d080@memphis.grassland.com.

4 - Hostnames must match. e.g, king@grassland.com and elvis@grassland.com.

5 - Domains must match. e.g, king@grassland.com and elvis@memphis.grassland.com.

6 - Always a match. e.g, king@grassland.com and elvis@parsely.com.

Default is 5

PURGED_HEADERS_CLIENT

A list containing one or more message headers that should be removed from outgoing client-side messages (i.e, messages sent with tmda-sendmail) prior to injection. Listed headers are case-insensitive, and the purging will only be attempted if the header actually exists.

NOTE: PURGED_HEADERS_CLIENT is run _after_ ADDED_HEADERS (see above).

Examples:

```
PURGED_HEADERS_CLIENT = ["bcc", "resent-bcc", "x-mailer"]
PURGED_HEADERS_CLIENT = ["bcc", "resent-bcc", "x-mailer",
"user-agent"]
```

Default is "Bcc:" and "Resent-Bcc"

PURGED_HEADERS_DELIVERY

A list containing one or more message headers that should be removed from _all_ delivered messages (i.e, messages stored in an mbox or a maildir, forwarded or passed to a program by TMDA). Listed headers are case-insensitive and the purging will only be attempted if the header actually exists.

See PURGED_HEADERS_CLIENT (above) for some examples.

No default

PURGED_HEADERS_SERVER

A list containing one or more message headers that should be removed from _all_ outgoing server-side messages (i.e, messages sent with tmda-filter) prior to injection. Listed headers are case-insensitive and the purging will only be attempted if the header actually exists.

See PURGED_HEADERS_CLIENT (above) for some examples.

No default

RECIPIENT_DELIMITER

A single character which specifies the separator between user names and address extensions (e.g, user-ext). The default under qmail is `-', while the default for Sendmail and friends is likely `+'. The default for MMDF is '='.

Default is "-"

RECIPIENT_HEADER

A string containing the name of a header (case-insensitive) whose contents will be taken as the envelope recipient of the message. The header should contain one fully-qualified e-mail address. This can be used in rare cases when you need to override the RECIPIENT environment variable, which is how TMDA normally determines the envelope recipient address. If EXT isn't set, the address extension will also be extracted from RECIPIENT_HEADER.

Example:

```
RECIPIENT_HEADER = 'x-originally-to'
```

Then, if the message contains the following header:

X-Originally-To: webmaster@domain.dom

TMDA will take webmaster@domain.dom as the envelope recipient of the message rather than the value of RECIPIENT.

WARNING: If you enable this feature, make sure that the method you are using to add the header will overwrite or replace an existing header of the same name (such as reformail/formail's -i and -I options do). Otherwise the envelope recipient will be invalid which will break the confirmation process.

No default

RESPONSE_DIR

Full path to a directory containing auto-response rate-limiting information. Only consulted if MAX_AUTORESPONSES_PER_DAY != 0

Default is ~/.tmda/responses

SENDER_TEMPLATE_VARS

Set this variable to True if you want to use 'sender' address variables in your templates.

Default is False (turned off)

SENDMAIL_PROGRAM

The path to the sendmail program, or sendmail compatibility interface when MAIL_TRANSPORT is "sendmail".

Defaults to one of the two standard locations (/usr/lib/sendmail, /usr/sbin/sendmail).

SMTPAUTH_PASSWORD

The password to authenticate with if your SMTP server requires authentication. You must also define SMTPAUTH_USERNAME if you use this option.

Examples:

```
SMTPAUTH_PASSWORD = "6Yu_9iKzs"
```

No default.

SMTPAUTH_USERNAME

The username to authenticate with if your SMTP server requires authentication. You must also define SMTPAUTH_PASSWORD if you use this option.

Examples:

```
SMTPAUTH_USERNAME = "johndoe"
```

No default.

SMTPHOST

SMTP host and optional port, when MAIL_TRANSPORT is "smtp". If the hostname or IP address ends with a colon (":") followed by a number, that suffix will be stripped off and the number interpreted as the port number to use. Otherwise, the standard SMTP port (25) will be used.

Examples:

```
SMTPHOST = "localhost"
SMTPHOST = "172.16.78.20:2525"
SMTPHOST = "mailhost.company.com"
SMTPHOST = "mailhost.company.com:1234"
```

Default is "localhost" (port 25 on the local host)

SMTPSSL

Set this variable to True to put the SMTP connection in TLS (Transport Layer Security) mode. All SMTP commands that follow will be encrypted. Your Python's socket module must be compiled with SSL support.

Default is False (turned off)

SMTPSSL_CERTFILE

When SMTPSSL is enabled, this is the optional name of a PEM formatted certificate chain file. Warning: This does not do any certificate verification.

No default.

SMTPSSL_KEYFILE

When SMTPSSL is enabled, this is the optional name of a PEM formatted file that contains your private key.

No default.

SMTP_MAX_SESSIONS_PER_CONNECTION

An integer specifying a ceiling on the number of SMTP sessions to perform on a single socket connection, when MAIL_TRANSPORT is "smtp". Some MTAs have limits. Set this to 0 to do as many as we like (i.e. your MTA has no limits). Set this to some number great than 0 and TMDA will close the SMTP connection and re-open it after this number of consecutive sessions.

Default is 0

SUMMARY_HEADERS

A list containing one or more message headers that should be displayed by tmda-pending's interactive mode. Listed headers are case-insensitive.

Examples:

```
SUMMARY_HEADERS = ['from', 'subject', 'x-spam-status']
SUMMARY_HEADERS = ['date', 'from', 'to', 'subject']
```

Default is Date, From, To, and Subject.

TAGS_CONFIRM

A list of one or more strings used to identify a confirmation address. The first element in the list will be used when generating a new confirmation address.

Example:

```
TAGS_CONFIRM = ['confirm', 'c']
```

Default is 'confirm'

TAGS_DATED

A list of one or more strings used to identify a dated address. The first element in the list will be used by `tmda-address' when generating a new dated address.

Example:

```
TAGS_DATED = ['dated', 'd', 'exp', 'expires']
```

Default is 'dated'

TAGS_KEYWORD

A list of one or more strings used to identify a keyword address. The first element in the list will be used by `tmda-address' when generating a new keyword address.

Example:

```
TAGS_KEYWORD = ['keyword', 'key', 'kw']
```

Default is 'keyword'

TAGS_SENDER

A list of one or more strings to identify a sender address. The first element in the list will be used by `tmda-address' when generating a new sender address.

Example:

```
TAGS_SENDER = ['sender', 's']
```

Default is 'sender'

TEMPLATE_DIR

Full path to a directory containing custom TMDA templates. Any templates found in this directory will be used, otherwise the default templates will be used.

Examples:

```
TEMPLATE_DIR = "/full/path/to/templates/"
TEMPLATE_DIR = "~/.tmda/templates/"
```

No default.

TEMPLATE_DIR_MATCH_RECIPIENT

Set this variable to True if you want to use specific templates for different recipient addresses. Be sure to set TEMPLATE_DIR as well.

The TEMPLATE_DIR_MATCH_RECIPIENT feature enables TMDA to search for customized template files based on the recipient address. To continue with the example given for TEMPLATE_DIR_MATCH_SENDER, if a message arrives for foo-lists-guitar@bar.baz.de then TMDA will search for template files in the following directories:

```
foo-lists-guitars@bar.baz.de/

foo-lists@bar.baz.de/

foo@bar.baz.de/

bar.baz.de/

baz.de/

de/
```

This example assumes that RECIPIENT_DELIMITER is set to `-'. This feature also works for *-confirm-* addresses.

When both TEMPLATE_DIR_MATCH_RECIPIENT and TEMPLATE_DIR_MATCH_SENDER are enabled, the TEMPLATE_DIR_MATCH_RECIPIENT directories are searched after the sender directories.

Default is False (turned off)

TEMPLATE_DIR_MATCH_SENDER

Set this variable to True if you want to use sender specific template directory matching. Make sure you also have TEMPLATE_DIR set.

With this feature enabled, TMDA will look for templates in a subdirectory of TEMPLATE_DIR that matches the sender address, and then increasingly general portions of the domain part of the address.

For example, if mail arrives from foo@bar.baz.de, TMDA will look for templates in these subdirectories of TEMPLATE_DIR, in this order:

```
foo@bar.baz.de/

bar.baz.de/

baz.de/

de/
```

If no sender based templates can be found, TEMPLATE_DIR itself and then the default locations will be tried.

Default is False (turned off)

TEMPLATE_EMAIL_HEADERS

A list containing the names of headers in your templates that contain an e-mail address. This is necessary so that the e-mail address will avoid being RFC 2047 encoded when handling internationalized headers.

Example:

```
TEMPLATE_EMAIL_HEADERS = ["from", "reply-to"]
```

Default is "From:" and "Reply-To:".

TEMPLATE_ENCODED_HEADERS

A list containing the names of headers in your templates that might contain an RFC 2047 encoded string. This is necessary so that they can be decoded first when handling internationalized headers.

Example:

```
TEMPLATE_ENCODED_HEADERS = ["subject"]
```

Default is "Subject:".

TERSE_SUMMARY_HEADERS

A list containing one or more message headers that should be displayed by tmda-pending's `--terse option'. Listed headers are case-insensitive. 'from_name' and 'from_address' can be used to specify the Fullname and e-mail address from the message's "From:" header.

Examples:

```
TERSE_SUMMARY_HEADERS = ["return-path"]
TERSE_SUMMARY_HEADERS = ["from_name", "from_address", "subject"]
```

Default is the Fullname followed by Subject.

TIMEOUT_UNITS

Dictionary that contains translations of timeout unit strings. This dictionary must contains the keys Y,M,w,d,h,m,s with respective internationalized strings as values.

Examples:

```
TIMEOUT_UNITS = {

     'Y' : "anos",
      'M' : "meses",
      'w' : "semanas",
      'd' : "dias",
      'h' : "horas",
      'm' : "minutos",
      's' : "segundos"}
```

Default is English values.

TMDAINJECT

A string containing one or more of the following letters which toggle on certain behaviors in tmda-inject (which is used by both tmda-sendmail and tmda-ofmipd):

'd' - Add a new Date field to the message, clobbering any

> existing Date fields. Normally tmda-inject only adds a Date field if the incoming message lacks one.

'i' - Add a new Message-ID field to the message, clobbering any

> existing Message-ID fields. Normally tmda-inject only adds a Message-ID field if the incoming message lacks one.

Examples:

```
TMDAINJECT = "i"
TMDAINJECT = "di"
```

No default.

USERNAME

The left-hand side of your e-mail address (before `@').

Example:

```
USERNAME = "jdoe"
```

Defaults to your UNIX username.

USEVIRTUALDOMAINS

Set this variable to False if want to turn off TMDA's qmail virtualdomains support. This should obviously only be done if you are not running any qmail virtualdomains, but it will improve performance.

Default is True (turned on)

VIRTUALDOMAINS

virtualdomains defaults to /var/qmail/control/virtualdomains, but this lets you override it in case it is installed elsewhere. Used for virtualdomain processing in tmda-filter.

Default is /var/qmail/control/virtualdomains

X_TMDA_IN_SUBJECT

With this variable set to True, tmda-inject will parse the Subject header looking for `X-TMDA' actions, and then remove them before sending the message. This is useful for users that desire the `X-TMDA' override behavior, but don't use an MUA which easily allows addition of arbitrary headers (e.g, Outlook).

The Subject header should contain `X-TMDA' followed by whitespace followed by the desired action followed by the real subject. Case is insensitive.

Examples:

```
Subject: X-TMDA dated Re: You're fired!
Subject: X-TMDA dated=5M Re: You're fired!
Subject: X-TMDA sender Re: You're fired!
```

In all cases, the resulting subject will simply be:
Subject: Re: You're fired!

Default is False (turned off)

2007-07-30 15:45