



## TABLE OF CONTENTS

- [1 Introduction](#)
- [2 PowerMTA Installation](#)
- [3 PowerMTA Configuration](#)
- [4 PowerMTA Management](#)
- [5 Web-Based Monitoring](#)
- [6 Command Line Tool](#)
- [7 Application Programming Interfaces](#)
- [8 VirtualMTA Support](#)
- [9 Mailmerge Support](#)
- [10 Advanced Features](#)
- [11 Accounting Files and Statistics](#)
- [12 Processing Inbound email](#)
- [13 Acknowledgments](#)
- [14 Directives index](#)

SparkPost, Inc.

PowerMTA™ v5.0

User's Guide

Copyright © 1999-2020, Port25 Solutions, Inc. All Rights Reserved. PowerMTA is a trademark of Port25 Solutions, Inc. All other brands and names are property of their respective owners.

# 1 Introduction

## 1.1 General Description and Basic Functionality

PowerMTA is specialized, high performance Message Transfer Agent (MTA) software that intelligently and efficiently delivers large volumes of email, allowing for maximum delivery and response. While all-purpose MTAs in use today perform a whole variety of tasks including delivering email, these general solutions fall short both in regards to scalability and relevant feature sets, negatively affecting delivery rates and subsequent ROI. In contrast, PowerMTA was developed for this particular task, helping legitimate, permission based email marketers, publishers, and service providers overcome the business and technology challenges of email message delivery.

PowerMTA's basic functionality consists of receiving inbound or outbound messages via either standard email protocols (SMTP) or programming interfaces, routing them based on either name services (DNS) or pre-configured information and delivering them using SMTP or locally to a program.

## 1.2 Minimum Requirements

PowerMTA currently runs on:

- Microsoft Windows 2008/2012/2016/2019 on Intel and compatible processors
- Red Hat Enterprise Linux 6.0 or later on Intel and compatible processors (32 and 64 bit)
- Debian based Linux on Intel and compatible processors (32 and 64 bit)

Prerequisites:

- a name server (DNS), not necessarily on the same host,
- a static IP address and a domain name,
- a license activation key (LAK) from SparkPost support,
- minimum hardware requirements, depending on the size of the workload and mail queues, and the desired throughput rate.
- NTP (Network Time Protocol) – Clock must be synced to internet time server

Contact SparkPost if you need hardware recommendations for your specific workload and throughput requirements.

## 1.3 Supported interfaces

PowerMTA supports the following submission and delivery interfaces:

- Submission via standard (extended) SMTP
- Submission via C/C++ API
- Submission via .NET API
- Submission via Perl API
- Submission via Java API
- Submission via "pickup" directory
- Delivery via standard (extended) SMTP
- Delivery to a local program

and will be supporting others in the future. Unless otherwise noted, these interfaces are available on all supported operating systems. You will find detailed information about the submission APIs [here](#). Sample applications for the various APIs are also distributed along with PowerMTA.

## 1.4 Monitoring/Management Tools Summary

PowerMTA was designed to require only minimal configuration and very little management. However, for those who want to actively monitor and manage, a variety of flexible tools are available to do just that, both on a manual or automated basis. Full details and examples are provided in this Guide, but here is a quick list of the resources available:

- a built-in web-based monitor that allows you to view in real time what is currently taking place within PowerMTA. This monitor also includes a view and top level break down of the current mail queues, a view and breakdown of each VirtualMTA queue, a view of each job/campaign in the queue, as well as the last error received for each domain.
- a command line tool supporting a variety of monitoring and management commands, with four output formats (text, XML, JSON and DOM-style), which allow for easy integration into existing monitoring infrastructures or custom parsing and monitoring applications.
- detailed logging, both for incoming and outgoing connections, configurable both globally and for specific destination domains.
- detailed accounting data in CSV format on each successful and unsuccessful transfer/delivery that can be post processed for delivery verification/accountability, for performance and throughput analysis, etc.
- a powerful accounting file search and reporting tool, allowing one to search for records in the accounting file and then to produce a custom output based on specific fields in the matching records.
- an accounting statistics application for extracting the most commonly needed information for monitoring and capacity planning.
- the [SparkPost Signals](#) service that analyzes your email sending and response data to help you identify potential problems.

## 2 PowerMTA Installation

If you are upgrading to a newer version of PowerMTA, please make sure that you have obtained a compatible PowerMTA license for the new version that you are installing. For instance, if you are upgrading from 4.5 to 5.0, you should ensure that a valid 5.0 license also exists in PowerMTA's license file before attempting to upgrade. If your license has the "SupportExpires" field in it, please ensure that the current date is not past the "SupportExpires" date. The pre-install checks will fail the installation process if a compatible license for the version you are installing is not found.

### 2.1 Installing on Microsoft Windows

How to install:

1. Download the kit.
2. Log in as Administrator.
3. Open a command prompt as an administrator (right click and select "*Run as Administrator*").
4. Run the PMTA MSI from within this window.
5. If you are installing PowerMTA for the first time or upgrading from a different version, copy the license key received from SparkPost to a file named `license.dat` in the directory where PowerMTA was installed (e.g., `C:\pmta\license.dat`). Please ensure that the file is stored as ASCII, *not* in Unicode.
6. Check that the configuration (e.g., `C:\pmta\config.dat`) suits your needs; the file can be edited with any ASCII-editor such as Notepad. See [Initial Configuration](#) for details.
7. Start the mailer from the Services item in the Control Panel or by typing `net start pmta` in a Command Prompt window.
8. If you have an anti-virus and/or indexing service, they should not scan or handle the spool files, accounting files, or log files (to prevent file locking issues with PowerMTA).

### 2.2 Installing on Linux

How to install:

1. Download the kit.
2. Log in as root.
3. If installing for the first time, optionally create a group called `pmta`. Membership to this group authorizes non-root users to execute PowerMTA commands and access the submission APIs. The group is created automatically by the installation procedures, but creating it manually allows you to also choose its numeric ID.
4. Install the package (substituting the file name as appropriate):

```
# rpm -Uvh PowerMTA-4.*.rpm
```

Or on Debian:

```
# dpkg -i PowerMTA-4.*.deb
```

5. If installing for the first time or upgrading from a different version of PowerMTA, copy the license key received from SparkPost:

```
# cp mykey.txt /etc/pmta/license
```

6. Check that the initial configuration (`/etc/pmta/config`) suits your needs, as described [below](#).

7. Start the mailer (only necessary immediately after the install: it will be started automatically upon system startup):

```
# /etc/rc.d/init.d/pmta start
```

8. Optionally, install `gdb`, which PowerMTA can use in order to obtain useful debugging information.

### 2.3 Initial Configuration

There are only two things that *must* be manually defined in the configuration file, in addition to the supplied default entries, in order to have PowerMTA up and running: the postmaster address(es) and the relaying control configuration.

Besides being the traditional address for which external parties can reach the maintainer of the mail server, the postmaster address is used by PowerMTA and its watchdog monitoring facility, `pmtawatch`, to send crash reports and other information to upon successful automatic restart of the mailer (after a crash).

For security reasons, PowerMTA does not allow external relaying by default; the default configuration only allows messages to be submitted from the local `127.0.0.1` IP address. In order to submit messages for delivery from a different host, you will need to enter its IP address(es) into the configuration file by creating one or more source entries and specifying the `always-allow-relaying` directive.

Note that the `relay-domain` and `relay-address` directives in the configuration file are really only useful when PowerMTA is used to handle inbound traffic. These allow you to specify the recipient domains or email addresses of messages which PowerMTA is to accept from any sources. When configuring PowerMTA for outbound, you want it to accept email from your feeders for all domains by using `always-allow-relaying`.

See [PowerMTA Configuration](#) for detailed information on the configuration file.

### 2.4 Uninstalling

On each platform PowerMTA is installed with the platform's standard packaging software. You can uninstall PowerMTA using the platform's existing uninstall option:

- On Windows, open the Control Panel, click "Programs", click "Uninstall or change a program", select the PowerMTA item, and click on "Uninstall" or "Uninstall/Change" at the top of the program list. After confirming that you would like to proceed, that will initiate the uninstall process.
- On Linux, log in as root and execute `rpm -e PowerMTA` or `dpkg -r PowerMTA`

Any configuration, logging, accounting, and message files found in PowerMTA's directories will not be automatically removed, and neither will the directories themselves. On Linux the uninstall software will display a list of the remaining files and the commands you need to execute to remove them. On Windows, you can simply remove the complete directory tree into which you had installed PowerMTA.

## 2.5 Running PowerMTA on a Virtual Server

While this is technically possible, it is not a recommended setup unless the virtual server has adequate hardware and the virtual server software is configured correctly. The sharing of resources on a misconfigured virtual machine may create an environment under load that could cause PowerMTA to become unresponsive.

If PowerMTA is going to be run on VMware®, please take the following steps to ensure that PowerMTA gets the needed resources:

- Open your vSphere client
- Right click on the PowerMTA server and select "Edit Settings"
- Select the "Resources" tab
- Change the shares value for CPU/Memory/Disk to the following values:

```
CPU = 1000000
Memory = 1000000
Disk = 4000
```

- For CPU, Memory, and Disk also make sure the "Unlimited" checkbox is checked for each in the same section
- Restart the PowerMTA server

These can be changed again later if needed or the settings can be turned down to find the optimal setting, but we have found that this helps ensure PowerMTA runs properly in a virtual environment.

## 3 PowerMTA Configuration

### 3.1 Working with the configuration file

PowerMTA's configuration files are simple UTF-8 encoded text files that can be edited by most editors, such as Notepad, vim and Emacs. Most settings are entered one per line and are formed by a keyword on the left identifying a directive and a value on the right, as in the example below:

```
host-id 5
```

Certain directives, like those applying to a specific destination domain, are grouped based on what they apply to:

```
<domain test.port25.com> # 'test' host
    smtp-hosts [127.0.0.1]:11111
    log-commands yes
</domain>
```

As indicated in the examples above, comments can be introduced by prefixing the comment with a hash mark "#". Like other file formats using the hash mark for comments, PowerMTA ignores the "#" and any characters following it until the end of the line.

Since the files are UTF-8 encoded, internationalized domain names and email addresses can be entered directly:

```
<domain bücher.de>
...
</domain>
```

PowerMTA reads the configuration upon startup. After making changes, you must either execute the `pmta reload` command or restart PowerMTA. Reloading the configuration is more efficient. Most configuration changes can be reloaded, but changes to some directives require PowerMTA to be restarted; this is indicated in each such directive.

PowerMTA supports the ability to remotely modify/upload a configuration file via the web monitor (and thus via HTTP). To push the configuration programmatically, one needs to issue a HTTP POST, passing the configuration file in the field named "file" to the URL <http://127.0.0.1:8080/configFile>. One also needs to be authorized for admin access via the `http-access` directive.

### 3.2 Syntax definitions

#### BOOLEAN

Valid values are yes, no, true, or false.

#### DOMAIN pattern

Domain patterns are used to match domain names in various parts of PowerMTA and can be specified in one of the following formats:

- `*` - Matches all domains.
- `domain` - A single, fully qualified domain name. Matches that specific domain only.
- `*.domain` - Matches any subdomains of `domain`, but *not* the domain itself.
- `/*.domain` - Matches `domain` as well as its subdomains.
- `[ip/mask]` - CIDR format, matches domains in the dotted decimal format, like [127.0.0.1]. The actual IP address in the domain must match the given CIDR specification (*ip* prefix and (optional) *mask* number of significant bits). [0/0] matches all domains in dotted decimal format.

#### REGEX

A perl-compatible regular expression (PCRE). For more information, please refer to the [Perl documentation](#) or the [PCRE library package](#).

Examples:

```
/^s*eval/
/unknown user/
/^421 .* too many errors/
```

#### TIME-INTERVAL

A string like 1w2d12h or 300s where w is weeks, d is days, h is hours, m is minutes, and s is seconds.

### 3.3 Configuration Directives

### 3.3.1 Global Directives

These directives are "global" and affect PowerMTA as a whole.

#### 3.3.1.1 total-max-smtp-out

Used to configure the limit on the number of simultaneous outbound connections that PowerMTA can make at any point in time.

Standard PowerMTA license is always limited to 200 simultaneous outbound connections. Any `total-max-smtp-out` setting above 200 is ignored by PowerMTA.

Enterprise PowerMTA license defaults to 1200 simultaneous outbound connections. There is no upper limit on the `total-max-smtp-out` setting when running with an Enterprise PowerMTA license.

**Note: While PowerMTA does not enforce any upper limit on the setting for this directive, please note that the limit is still subject to the hardware and OS limits on the machine that PowerMTA is installed on. For best results, it is recommended that `total-max-smtp-out` should not be set to a number higher than 7500. Please consult your OS and hardware manuals for arriving at the best value for this limit on the system that PowerMTA has been installed on.**

For the per-queue version of this directive, see [max-smtp-out](#).

For the per-Virtual MTA version of this directive, see [max-smtp-out](#).

**Default:** 1200 for Enterprise license, 200 for Standard license.

**Example:**

```
total-max-smtp-out 5000
```

#### 3.3.1.2 total-max-smtp-in

Used to configure the limit on the number of simultaneous inbound connections that PowerMTA should support at any point in time.

**Note: While PowerMTA does not enforce any upper limit on the setting for this directive, please note that the limit is still subject to the hardware and OS limits on the machine that PowerMTA is installed on. The best value for this limit depends on the number of connections that you expect will be made to PowerMTA by your submissions applications that submit messages to PowerMTA. We recommend that the limit should not be set above the default limit of 100 in practice. Please consult your OS and hardware manuals for arriving at the best value for this limit on the system that PowerMTA has been installed on.**

**Default:** 100

**Example:**

```
total-max-smtp-in 110
```

#### 3.3.1.3 postmaster

Specifies an email address for the person responsible for PowerMTA's operation. This should *always* be specified because it is used by the internal watchdog facility for crash and LAK expiry notifications. Messages sent to the local postmaster and abuse addresses are also by default forwarded to the given addresses. You can specify more than one address by entering `postmaster` multiple times.

**Example:**

```
postmaster you@your.domain
postmaster you@your.domain.at.home
```

#### 3.3.1.4 spool

Specifies a directory for the message spool, where PowerMTA stores the message files while queued for delivery. While at least one spool directory is required, additional entries can be added for greater performance, for example by distributing the I/O load over several disks.

**Deprecated** Use `<spool>` instead.

**Example:**

```
spool c:\pmta\spool
```

#### 3.3.1.5 spool-delete-corrupted

Determines whether PowerMTA should delete any message files in the spool found to contain encoding errors, such as those caused by disk data corruption. If set to `false`, corrupted spool files, if any, are ignored and remain on disk.

**Default:** false

**Example:**

```
spool-delete-corrupted true
```

#### 3.3.1.6 spool-min-free-space

Instructs PowerMTA to stop accepting email into a spool directory if the space available on its file system falls below the amount specified. When available disk space increases above that level, messages are accepted again.

On Unix, this is based on the space available to non-root users.

**Deprecated** Use `min-free-space` within `<spool>` instead.

**Syntax:**

```
spool-min-free-space N
```

where N is a positive integer specifying a size in MB.

**Default:** 0

**Example:**

```
spool-min-free-space 100 # MB
```

---

**3.3.1.7 spool-max-recipients**

Specifies the maximum number of recipients to allow across all spool directories, to help prevent overwhelming PowerMTA with more messages than it has the resources to handle. When this limit is reached, feeding new messages into PowerMTA fails with a transient error message.

If set to auto, the maximum number is determined automatically based on system resources.

**Syntax:**

```
spool-max-recipients { N | auto }
```

where N is a positive integer representing the maximum number of recipients allowed.

**Default:** auto

**Example:**

```
spool-max-recipients 10000000
```

---

**3.3.1.8 run-as-root**

Specifies whether PowerMTA should run as root or as user pmta. (Only supported on Unix.) You may want to not run PowerMTA as root for security reasons.

**Note:** This option requires a restart of PowerMTA service whenever changed.

**Default:** true

**Example:**

```
run-as-root false
```

---

**3.3.1.9 smtp-server-tls-certificate**

Specifies the X.509 certificate that PowerMTA should present when TLS is started on inbound SMTP connections (e.g., when STARTTLS is used). The certificate must be in PEM format.

**Syntax:**

```
smtp-server-tls-certificate FILENAME PASSWORD
```

**Example:**

```
smtp-server-tls-certificate /etc/pmta/yourcert.pem SkJkemI0zffkez
```

---

**3.3.1.10 smtp-server-tls-ca-file**

Specifies a "CA file" containing Certification Authority certificates to allow verifying any certificates presented by clients on inbound SMTP connections. The certificates must be in PEM format.

**Example:**

```
smtp-server-tls-ca-file /etc/pmta/certchain.pem
```

---

**3.3.1.11 smtp-server-tls-allow-ssl3**

This directive is no longer valid as PowerMTA no longer supports SSLv3.

---

**3.3.1.12 smtp-server-tls-allow-tls1**

Allows PowerMTA to use TLSv1 when accepting a secure connection from a client during STARTTLS.

**Default:** false

**Example:**

```
smtp-server-tls-allow-tls1 true
```

---

**3.3.1.13 smtp-server-tls-allow-tls1.1**

Allows PowerMTA to use TLSv1.1 when accepting a secure connection from a client during STARTTLS.

**Default:** true

**Example:**

```
smtp-server-tls-allow-tls1.1 false
```

---

**3.3.1.14 smtp-server-tls-allow-tls1.2**

Allows PowerMTA to use TLSv1.2 when accepting a secure connection from a client during STARTTLS.

**Default:** true

**Example:**

```
smtp-server-tls-allow-tls1.2 false
```

---

```
smtp-server-tls-allow-tlsv1.2 false
```

### 3.3.1.15 smtp-server-tls-allow-tlsv1.3

Allows PowerMTA to use TLSv1.3 when accepting a secure connection from a client during STARTTLS.

**Default:** true

**Example:**

```
smtp-server-tls-allow-tlsv1.3 false
```

### 3.3.1.16 smtp-server-tls-ciphers

Specifies the ciphers that PowerMTA is allowed to use when negotiating a session using a TLSv1.2 or earlier protocol with a client during STARTTLS. Please see documentation for OpenSSL command 'ciphers' for valid cipher names that can be used with this.

**Default:** HIGH:!ADH:@STRENGTH

**Example:**

```
smtp-server-tls-ciphers HIGH:!ADH:@STRENGTH
```

### 3.3.1.17 smtp-server-tls-v1.3-ciphersuites

Specifies the ciphersuites that PowerMTA is allowed to use when negotiating a session using the TLSv1.3 protocol with a client during STARTTLS. Please see documentation for OpenSSL command 'ciphers' for valid ciphersuite names that can be used with this.

**Default:**

TLS\_AES\_256\_GCM\_SHA384:TLS\_CHACHA20\_POLY1305\_SHA256:TLS\_AES\_128\_GCM\_SHA256:TLS\_AES\_128\_CCM\_SHA256:TLS\_AES\_128\_CCM\_8\_SHA256

**Example:**

```
smtp-server-tls-v1.3-ciphersuites TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
```

### 3.3.1.18 smtp-source-host

On a multi-homed host, this directive specifies an IP address (or IP address range) and host-name from which PowerMTA is to initiate outgoing SMTP connections. You can enter this directive several times and/or specify a CIDR IP address range along with the host name, instructing PowerMTA to use the various source IP addresses in round-robin fashion. All local IP addresses matching the IP range specified are added to the list of source IP addresses, except for 127.0.0.1, which is automatically excluded since it can't be used to connect to remote hosts.

When specified within a VirtualMTA, this directive overrides the global source IPs for connections established to send messages through that VirtualMTA.

This directive may also designate the hostname used for SMTP greeting messages. See [host-name](#) definition in [Global Directives](#) for more information.

**Syntax:**

```
smtp-source-host CIDR HOST-NAME
smtp-source-host IP HOST-NAME
```

**Example:**

```
smtp-source-host 10/8 baz.quux.com
smtp-source-host 1.2.3.4 foo.bar.com
```

### 3.3.1.19 alias

This directive allows a list of email addresses to be defined for which PowerMTA should forward messages to different addresses. For example, if a message is received by PowerMTA for example@domain1abc.com, then forward the message on to support@port25.com. See [Aliases / Forwarding](#) for more information.

**Syntax:**

```
alias SOURCE TARGET
```

**Example:**

```
alias example@domain1abc.com support@port25.com
```

### 3.3.1.20 smtp-ip

**Deprecated** Use smtp-listener instead.

### 3.3.1.21 smtp-port

**Deprecated** Use smtp-listener instead.

### 3.3.1.22 smtp-listener

Specifies an IP address and port to listen for incoming SMTP connections. Multiple smtp-listener entries can be specified in a configuration file, each of which specifying a different IP address and/or port. 0.0.0.0 may be used more than once to set multiple ports for all IP addresses on the system. A CIDR IP address range can also be specified. If any smtp-listener directives are specified, PowerMTA ignores the smtp-ip and smtp-port directives. However, if no smtp-listener directives are specified, one listener is implicitly started for 0.0.0.0:25.

smtp-listener accepts an optional source= parameter which allows you to specify a <source> entry with settings to override those obtained from the normal matching by source IP address.

**Syntax:**

```
smtp-listener CIDR:PORT [source=SOURCE-NAME]
```

**Example:**

```
smtp-listener 1.2.3.4:25 source=mta1
smtp-listener 0.0.0.0:26

<source mta1>
  default-virtual-mta mta1
</source>

<source 10.0.0.0/8>
  allow-mailmerge yes
  always-allow-relaying yes
</source>

<source 0/0>
  allow-mailmerge no
  always-allow-relaying no
</source>
```

In the example above, VirtualMTA mta1 is selected by default for any messages received through 1.2.3.4. Permission for relaying and for using the mailmerge extensions are granted to connections coming from 10.\*, whether they connect to 1.2.3.4 or not.

**Warning** If you specify `always-allow-relaying` on a source selected from a `smtp-listener` directive, anybody who can connect to the given IP address will be allowed to relay. In doing so, you might be creating an "open SMTP relay".

**3.3.1.23 custom-dsn-from-header**

Used to configure the content of From header in DSN reports generated by PowerMTA.

**Example:**

```
custom-dsn-from-header "Postmaster <foobar@there.com>"
```

**3.3.1.24 smtp-await-slot**

Specifies whether PowerMTA should wait for a new connection slot when a new (incoming) SMTP connection request is received but no more slots are available. If set to false, PowerMTA responds with a 421 greeting, indicating that the service is not available, and closes the connection.

**Default:** false

**Example:**

```
smtp-await-slot true
```

**3.3.1.25 host-id**

When using `add-message-id-header`, `host-id` specifies a per server unique identifier that helps prevent the same message ID from being used on two separate instances of PowerMTA (multiple servers).

**Default:** 0

**Example:**

```
host-id 5
```

**3.3.1.26 domain-suffix**

Specifies the domain name to append when an email address is received whose domain is not fully qualified.

**Example:**

```
domain-suffix bounce.port25.com
```

**3.3.1.27 host-name**

This directive specifies a (fully qualified) host name for the local host. If the machine on which PowerMTA is running is known under several names, these should all be listed by adding various `host-name` directives. The first name specified is picked as the "main" name and may be used for the SMTP greeting messages, EHLO/HELO commands, etc. When specified within a VirtualMTA, this directive overrides the global host name for messages sent through that VirtualMTA. When used, this directive behaves like `relay-domain`, meaning that it will accept all messages for the defined domain.

For the hostname used for SMTP greeting messages, PowerMTA first looks for a `default-virtual-mta` setting in the matching `<source>` definition of the connecting client's IP address. If the `default-virtual-mta` is a VMTA name, the hostname displayed will be the hostname of the matching source IP defined in the VMTA. If the VMTA name is a pool, PowerMTA will look at the first VMTA in the pool (note that there is no rotation performed here - the hostname will always belong to the first VMTA). If the `default-virtual-mta` is set to the special value `by-smtp-source-ip`, then PowerMTA looks for a VMTA which has an `smtp-source-host` that matches the IP on the PowerMTA machine that the client connected to. If none of these options matched, PowerMTA will display the hostname of the first global `smtp-source-host` that matches the IP the client connected to or else the first global `host-name`.

**Example:**

```
host-name mail.foo.com
```

**3.3.1.28 name-server**

This directive specifies the IP address of a name server (DNS server) for PowerMTA to use. If none is entered, PowerMTA uses the name servers configured in the operating system. If more than one `name-server` is defined, PowerMTA will round robin through all defined name servers in a load balancing fashion.

**Example:**

```
name-server 10.1.2.3
name-server 8.8.8.8
```

#### 3.3.1.29 thread-min-priority, thread-max-priority

**Unix specific directive.** Specifies range of process priorities to use for the various threads within PowerMTA. Normally these directives should be left at their defaults, however it may be necessary to weight process priorities differently when PowerMTA shares a host with other server software. `thread-min-priority` specifies the "slower" priority and `thread-max-priority` specifies the "faster" priority.

**Default:** 15 for `thread-min-priority` 0 for `thread-max-priority`

**Example:**

```
thread-min-priority 5
thread-max-priority 50
```

#### 3.3.1.30 thread-reuse

Normally, once a thread is finished with its work (like, for example, handling a connection), PowerMTA retains it for later reuse. Setting this directive to false causes PowerMTA to discard threads as soon as they are no longer needed, avoiding certain external problems such as slowing down the top command on Linux. However, this costs some performance and also prevents PowerMTA from working around a bug in glibc versions prior to 2.2.2 that severely reduces the amount of virtual memory available, possibly causing PowerMTA to crash if the queue grows very large.

**Default:** true

**Example:**

```
thread-reuse false
```

#### 3.3.1.31 mailmerge-expands-undefined-variables

When set to "false", mail merge inserts like "[foo]" remain as "[foo]" in the message if the "foo" variable isn't defined, rather than getting expanded to the empty string.

**Default:** true

**Example:**

```
mailmerge-expands-undefined-variables false
```

#### 3.3.1.32 pmc-acct-min-free-space

If you use PowerMTA's companion product, the PowerMTA Management Console (PMC), this directive controls when to start deleting old CSV files written for the PMC. If the amount of free space on the disk drops below this number, PowerMTA will start to delete the oldest PMC accounting files in order to make space for new accounting files.

If the PMC runs properly, you should not see accounting files building up on PowerMTA's side, but if there is a problem (e.g., PMC machine down or too busy, firewall rules preventing the PMC to connect to PowerMTA, or PMC just turned off but not properly de-registered from PowerMTA), this directive can limit the amount of disk space PowerMTA uses. When PowerMTA runs out of disk space and discards accounting files, that data is lost forever, and you'll see "holes" in the PMC's reports.

**Default:** unlimited

**Example:**

```
pmc-acct-min-free-space 5G
```

#### 3.3.1.33 track-time-first-attempt

This directive tells PowerMTA to keep track of the time it first attempted to deliver an email. An attempt may mean a successful delivery, bounce, transient failure, or transient queue failure. This time is available in successfully delivered, bounced, and transient failure accounting records.

**Default:** false

**Example:**

```
track-time-first-attempt true
```

#### 3.3.1.34 include

This directive can be used at global level to specify an additional configuration file to process. This can be used, for example, to facilitate maintenance of the configuration files across multiple hosts, by storing those settings which differ from host to host in a separate file and including it from the main (common) file. Wildcards may be used. An include may be used at global level and within `<source>`, `<domain>`, `<virtual-mta>` or `<smtp-pattern-list>` tag.

**Example:**

```
include /etc/pmta/vmtas.txt
```

#### 3.3.1.35 Relaying Control Directives

##### 3.3.1.35.1 always-allow-relaying

Specifies whether SMTP senders from the given source should be allowed to relay email through PowerMTA, even if the recipients are not for one of the relay-domains. Typically you should enable this in the source entries of "internal" hosts feeding PowerMTA through SMTP.

**Default:** false

**Example:**

---

```
always-allow-relaying false
```

---

#### 3.3.1.35.2 relay-domain

Recipient domain name (or pattern) for which PowerMTA will accept email. Any host is allowed to relay email for these domains, independent of its source IP address. This directive may be specified more than once.

This directive is mostly useful for handling inbound email. When setting up PowerMTA for relaying inbound email to some other host, you should enter the corresponding domain name(s) into the relay domains list. Conversely, if in your installation PowerMTA is handling outbound email only, there is no need to configure any relay domains.

##### Syntax:

```
relay-domain DOMAIN
```

where DOMAIN is a domain name pattern.

**Default:** local domains

##### Example:

```
# allow relaying of mail to some.domain
relay-domain some.domain
```

---

#### 3.3.1.35.3 relay-address

Recipient address for which PowerMTA will accept email. Any host is allowed to relay email to these addresses, independent of its source IP address. This directive may be specified more than once.

This directive is mostly useful for handling inbound email such as setting up PowerMTA for relaying inbound email to some other host. Conversely, if in your installation PowerMTA is handling outbound email only, there is no need to configure any relay addresses.

##### Syntax:

```
relay-address { EMAIL-ADDRESS | REGEX }
```

##### Example:

```
relay-address bounce@bounce.yourdomain.com
relay-address /@foo(bar)?\.com$/
```

---

#### 3.3.1.35.4 relay-debug

Specifies that relaying control is to be performed in "debugging" mode. Normally, recipients to which relaying is not allowed are rejected with persistent (5XX) SMTP error codes, causing immediate bounces. In debugging mode, a transient (4XX) error is returned instead to the sending mailer, allowing it to continue trying to deliver to that recipient. This directive comes handy when making changes to the relaying control configuration since it gives you a chance to review the log for any unintended rejections and correct the configuration before messages are bounced.

**Default:** false

##### Example:

```
relay-debug true
```

---

### 3.3.1.36 General Queuing and Delivery Directives

---

#### 3.3.1.36.1 domain-key

Specifies a domain key to use for signing messages delivered through the given VirtualMTA or for any messages if configured at the global level. You can specify multiple domain-key entries. PowerMTA determines the key to use by going through the keys sequentially and picking the first key whose domain either equals or is a parent of the message's *sending domain*. If there are matching domain keys at both the global level and the VirtualMTA level, the VirtualMTA's domain keys take precedence. The sending domain is that of the Sender header, or, if Sender is not present, of the From header. If no keys match the sending domain, the message is not signed. The wildcard \* may be used in place of a domain. This will cause all messages to be signed using the sending domain (extracted from the headers). We recommend that all wildcard entries be placed last when multiple domain-key entries are specified, as a wildcard will be matched before a non-wildcard entry. For more information on how to set up DKIM, please see [PowerMTA DKIM Implementation](#).

##### Syntax:

```
domain-key SELECTOR,{DOMAIN|*},FILENAME
```

##### Example:

```
domain-key m,mail.port25.com,c:\pmta\m.pem
domain-key base,*,c:\pmta\base.pem
```

---

#### 3.3.1.36.2 invalid-virtual-mta-fallback

Specifies a fallback VirtualMTA to select for messages for which an invalid (non-existent) VirtualMTA was selected. By using this directive, you prevent PowerMTA from bouncing messages for which an invalid VirtualMTA was selected. Instead, the messages are delivered from the fallback VirtualMTA. This directive only works when feeding PowerMTA. It does not work with `reroute-to-virtual-mta` or `backoff-reroute-to-virtual-mta`.

##### Syntax:

```
invalid-virtual-mta-fallback VMTA-NAME
```

**Note:** To prevent PowerMTA from bouncing the messages to an invalid (non-existent) VirtualMTA, explicitly set `reject-invalid-virtual-mta` to false in the corresponding source scope (the default is true).

---

#### 3.3.1.36.3 include-headers-from

Specifies the name of a file containing RFC2822-formatted headers to insert in all messages delivered from a VirtualMTA. The headers are inserted immediately after the Received header added by PowerMTA. When used globally, all VirtualMTAs will inherit the settings. If a VirtualMTA already has the directive defined, both settings will be used.

**Syntax:**

```
include-headers-from FILENAME
```

---

### 3.3.1.36.4 domain-macro

Specifies the variable and bindings for domain macro expansion. Only letters, numbers, and the underscore are supported when naming a domain-macro. See [<domain> Directives](#) for more information on usage.

**Syntax:**

```
domain-macro MACRO-NAME DOMAIN-LIST
```

**Example:**

```
domain-macro rollupDomains yahoo.com,gmail.com,hotmail.com
domain-macro tld .com,.co.uk,.co.in
```

---

### 3.3.1.37 Pickup Directives

#### 3.3.1.37.1 pickup

This directive specifies the path to a "pickup directory" from which PowerMTA processes email submissions, as well as the path to a corresponding "bad mail" directory to which improperly formatted messages are moved. See [Pickup Directory](#) for more information. Only one pickup directory is supported.

**Syntax:**

```
pickup PICKUP-DIRECTORY BAD-FILE-DIRECTORY
```

**Example: (Windows)**

```
pickup c:\Inetpub\Mailroot\Pickup c:\Inetpub\Mailroot\BadMail
```

**Example: (Unix)**

```
pickup /opt/pickup/new /opt/pickup/badfiles
```

---

#### 3.3.1.37.2 pickup-retry-interval

**Deprecated**

---

#### 3.3.1.37.3 pickup-remove-dot

When using SMTP feeding and a line starts with a period, that period needs to be escaped by adding a second period. When using a pickup file, the period does not need to be escaped. If the feeding application is adding two periods to a pickup file this directive may be set to true to allow the second period to be removed for proper formatting of emails.

**Default:** false

**Example:**

```
pickup-remove-dot yes
```

---

### 3.3.1.38 Dummy SMTP ("blackholing") Directives

#### 3.3.1.38.1 dummy-smtp-ip

Specifies the IP address on which PowerMTA is to listen for incoming connections that are to be handled as "dummy" SMTP connections.

**Example:**

```
dummy-smtp-ip 127.0.0.1
```

---

#### 3.3.1.38.2 dummy-smtp-port

Specifies the TCP port on which PowerMTA is to listen for incoming connections that are to be handled as "dummy" SMTP connections.

**Default:** 0, meaning disabled.

**Example:**

```
dummy-smtp-port 2525
```

---

#### 3.3.1.38.3 dummy-smtp-await-slot

Specifies whether PowerMTA should wait for a new connection slot when a new dummy SMTP connection request is received but no more slots are available. If set to false, PowerMTA responds with a 421 greeting, indicating that the service is not available, and closes the connection.

**Default:** false

**Example:**

```
dummy-smtp-await-slot true
```

---

#### 3.3.1.38.4 dummy-smtp-has-chunking

Specifies whether dummy SMTP connections are to support the CHUNKING SMTP extension.

**Default:** false

**Example:**

```
dummy-smtp-has-chunking true
```

---

#### 3.3.1.38.5 dummy-smtp-has-pipelining

Specifies whether dummy SMTP connections are to support the PIPELINING SMTP extension.

**Default:** false

**Example:**

```
dummy-smtp-has-pipelining true
```

---

#### 3.3.1.38.6 dummy-smtp-has-verb

Specifies whether dummy SMTP connections are to support the VERP SMTP extension.

**Default:** false

**Example:**

```
dummy-smtp-has-verb true
```

---

#### 3.3.1.38.7 dummy-smtp-latency

Specifies the network latency, in milliseconds, to be simulated in dummy SMTP connections.

**Default:** 0

**Example:**

```
dummy-smtp-latency 100
```

---

#### 3.3.1.38.8 dummy-smtp-update-stats

Specifies whether dummy SMTP connections should update the traffic statistics displayed by e.g. the web-based Status Monitor.

**Default:** true

**Example:**

```
dummy-smtp-update-stats false
```

---

### 3.3.1.39 Web-Based Monitor Directives

#### 3.3.1.39.1 http-mgmt-port

Specifies the TCP port number for the web-based monitor. Since there is no standard port for this application, other software running on the same host may already be using that port. In this case, change the port to a different port number. If `http-mgmt-port` is set to 0, no web-based monitor is started.

Changing this directive requires a restart of the `pmtahttp` service.

**Default:** 8080

**Example:**

```
http-mgmt-port 8888
```

---

#### 3.3.1.39.2 http-redirect-to-https

Specifies whether plain HTTP connections to the web monitor are disabled. If this directive is set to `true`, the web monitor will only accept HTTPS connections, and any incoming HTTP connections are redirected to HTTPS.

**Default:** true

**Example:**

```
http-redirect-to-https true
```

**NOTE:** The default value is `false` in PowerMTA 4.5 whereas it is `true` in 5.0. So, `http-redirect-to-https` has to be explicitly set to `false` in 5.0 to disable automatic redirecting to https.

---

#### 3.3.1.39.3 http-access

Specifies the level of access given to the web monitor, depending on the IP address from where the request originated.

The keyword after the CIDR gives the level of access: `none`, `monitor` (read-only), `operator` (monitor level plus view configuration and run commands) or `admin` (operator level plus edit config).

Admin access is intended for things that actually modify PowerMTA's state (configuration, LAK, queues, etc.).

To restrict access to logs, `monitor` access level can be qualified with `deny-logs`.

The directive can be specified several times and builds an access list in which IP addresses are matched in the order in which they are

specified in the configuration file (i.e, from the top of the config file to the bottom).

Please see the pmtahttp.log file for errors relating to use of this directive. The pmtahttp.log file should indicate the IP address of any attempted accesses.

**Syntax:**

```
http-access CIDR { monitor[,deny-logs] | operator | admin | none }
```

**Default:** none

**Example:**

```
# allow internal hosts access to the web-based monitor
http-access 10.0.0.0/8 monitor

# allow all IPs to the web-based monitor (use for testing only! This allows
# anyone in the world access to the web monitor!)
http-access 0/0 monitor
```

---

### 3.3.1.39.4 http-csrf-protection

This directive enables protection against Cross-Site Request Forgery attacks in the web monitor. When set to 'true', the web monitor disallows HTTP GET requests that can change PowerMTA's or the Web monitor's state, and requires that HTTP POST requests either include an API key (see [http-api-key](#)) or a CSRF token received from a previous GET call to the web monitor.

**Default:** true

**Example:**

```
# legacy application X requires mutable GET support
http-csrf-protection false
```

---

### 3.3.1.39.5 http-api-key

This directive configures authentication keys for applications accessing the web monitor; it can be repeated to create multiple keys. Applications passing a valid key in the X-API-Key header are granted full (admin) access to the web monitor.

**Syntax:**

```
http-api-key STRING
```

where STRING is a set of alphanumeric characters or the underscore character, and between 32 and 128 characters in length.

**Example:**

```
# application keys
http-api-key abcd_0123_abcd_0123_abcd_0123_ab # app X
http-api-key 5f6e7e426949e3a5e4ec38e747e71963 # app Y
```

---

### 3.3.1.39.6 http-strict-transport-security-max-age

This directive can be used to configure the max-age value included in the Strict-Transport-Security returned in PowerMTA web monitor responses.

**Default:** 0

**Example:**

```
http-strict-transport-security-max-age 0
```

---

### 3.3.1.39.7 http-mgmt-source & no-http-mgmt-source

**Deprecated** Use http-access instead.

---

### 3.3.1.39.8 http-tls-allow-ssl3

This directive is no longer valid as PowerMTA no longer supports SSLv3.

---

### 3.3.1.39.9 http-tls-allow-tls1

Allows PowerMTA web monitor to use TLSv1 when accepting a secure connection from a client.

**Default:** false

**Example:**

```
http-tls-allow-tls1 false
```

---

### 3.3.1.39.10 http-tls-allow-tls1.1

Allows PowerMTA web monitor to use TLSv1.1 when accepting a secure connection from a client.

**Default:** true

**Example:**

```
http-tls-allow-tls1.1 false
```

---

### 3.3.1.39.11 http-tls-allow-tls1.2

Allows PowerMTA web monitor to use TLSv1.2 when accepting a secure connection from a client.

**Default:** true

**Example:**

```
http-tls-allow-tlsv1.2 false
```

### 3.3.1.39.12 http-tls-allow-tlsv1.3

Allows PowerMTA web monitor to use TLSv1.3 when accepting a secure connection from a client.

**Default:** true

**Example:**

```
http-tls-allow-tlsv1.3 false
```

### 3.3.1.39.13 http-tls-ciphers

Specifies the ciphers that PowerMTA web monitor is allowed to use when negotiating a session using a TLSv1.2 or earlier protocol with remote client. Please see documentation for OpenSSL command `ciphers` for valid cipher names that can be used with this directive.

**Default:** "HIGH:!ADH:@STRENGTH"

**Example:**

```
http-tls-ciphers HIGH:!ADH:@STRENGTH
```

### 3.3.1.39.14 http-tls-tlsv1.3-ciphersuites

Specifies the ciphersuites that PowerMTA web monitor is allowed to use when negotiating a session using the TLSv1.3 protocol with remote client. Please see documentation for OpenSSL command `ciphers` for valid ciphersuite names that can be used with this directive.

**Default:**

TLS\_AES\_256\_GCM\_SHA384:TLS\_CHACHA20\_POLY1305\_SHA256:TLS\_AES\_128\_GCM\_SHA256:TLS\_AES\_128\_CCM\_SHA256:TLS\_AES\_128\_CCM\_8\_SHA256

**Example:**

```
http-tls-tlsv1.3-ciphersuites TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
```

### 3.3.1.39.15 http-tls-ca-file

Specifies a "CA file" containing Certification Authority certificates to allow verifying any certificates presented by clients on inbound HTTP connections to the web monitor. The certificates must be in PEM format.

**Example:**

```
http-tls-ca-file /etc/pmta/certchain.pem
```

### 3.3.1.39.16 http-log-tls

Specifies whether PowerMTA web monitor should log information during negotiation of a TLS session with the remote client.

**Default:** false

**Example:**

```
http-log-tls false
```

### 3.3.1.39.17 http-log-data

Specifies whether web monitor data exchanges are logged or not. More verbose than `http-log-requests`.

**Default:** false

### 3.3.1.39.18 http-log-requests

Specifies whether web monitor data exchanges are logged or not.

**Default:** false

### 3.3.1.39.19 http-default-max-items

Specifies the default maximum number of items to show in the "Top Queues", "Top Domains", "Top VirtualMTAs", "Top Jobs" pages and the "Top Domains" table in the home page.

**Default:** 10

### 3.3.1.39.20 http-default-sort-order

Specifies the default sort order for items in the "Top Queues", "Top Domains", "Top VirtualMTAs", "Top Jobs" pages and the "Top Domains" table in the home page.

Set it to `name` to order the list alphabetically by name, `rcpt` to order the list by number of recipients in decreasing order, or `size` to order the list by volume/kbytes in decreasing order.

**Syntax:**

```
http-default-sort-order { rcpt | name | size }
```

**Default:** rcpt

---

### 3.3.1.40 Logging Directives

#### 3.3.1.40.1 log-auto-rotation

Specifies whether the logging files are to be rotated automatically at midnight. See the `log-rotate` directive for more information on file rotation.

**Default:** true

---

#### 3.3.1.40.2 log-file

Specifies the name of PowerMTA's logging file. On Windows, it can be either specified with a path name relative to the installation directory or with an absolute path. On Unix it must always be specified as an absolute path. The `log-file` directive controls the location of the PowerMTA log file, the HTTP log file, and the SNMP log file.

This directive is required and must always be configured in PowerMTA's configuration file.

**Syntax:**

```
log-file FILENAME
```

**Example:** (Windows)

```
log-file log\pmta.log
```

**Example:** (Linux)

```
log-file /var/log/pmta/pmta.log
```

---

#### 3.3.1.40.3 log-rotate

Specifies the number of files to keep when rotating the logging files. File rotation is a practical way of preventing files from growing too large and, at the same time, ensure that the most recent data is available in case it is needed. When the log files are rotated, PowerMTA renames the existing logging files, attaching a number to them, starting with 1 for the most recent file. On Unix, assuming a file named `log`, that's `log.N -> log.N+1`, `log -> log.1` and on Windows (assuming `log.txt`), `log-N.txt -> log-N+1.txt`, `log.txt -> log-1.txt`. It then creates a new file (either `log` or `log.txt`) to which it continues writing logging information. The number passed includes the file currently being written to, so a value of 1 means only the current file is kept (but no older files), 2 means the current file plus one old file, and so on. A value of 0 disables rotation entirely.

To trigger manual rotation run the following command:

```
pmta rotate log
```

**Default:** 8

**Example:**

```
log-rotate 8
```

---

#### 3.3.1.40.4 log-file-world-readable

Setting this directive to true will allow anyone with access to the machine to read log files generated by PowerMTA.

**Default:** false

---

### 3.3.1.41 MX and IP Connection Limit Directives

#### 3.3.1.41.1 mx-connection-limit

**Note that the [MX-rollover](#) feature is a better way to enforce MX-level controls than the use of `mx-connection-limit` directive.**

Per-MX connection limits can be configured with this directive.

One can define a specific MX record to match on, or use wildcards to define multiple MX records, for example, `*.domain.com`, using the same conventions available within `<domain>` tags. The connection limit applies on a per source IP address basis (per IP address that PowerMTA will make connections from). If the wildcard is used, and multiple records are matched, the connection limit will apply per record matched, and not globally across all matches.

The global directive can be entered several times in the configuration file, once per line, forming a list of rules specifying the limits for the various MX records. It is important to note that `MX-NAME` matches the MX record for a domain, and not the recipient domain itself.

This directive is most useful when an ISP or hosting provider hosts hundreds of domains, but has one or few MX records for all of those domains. This directive allows you to limit the number of connections per MX, regardless of the number of recipient domains you are mailing to. Note again, that the connection limit applies on a per source IP address basis and not per VirtualMTA.

**Syntax:**

```
mx-connection-limit MX-NAME N
```

where `MX-NAME` is the domain pattern that matches the MX(es) for whom the limit is being set and `N` is the number of simultaneous connections allowed to each MX in that group of MX(es).

**Example:**

```
mx-connection-limit *.secureserver.net 20
```

```
mx-connection-limit *.aspmx.l.google.com 5
```

---

#### 3.3.1.41.2 ip-connection-limit

Per-IP connection limits can be configured with this directive. The IP in this case refers to the IP address of the MX that PowerMTA is

connecting to.

The directive is only permitted in global scope. The syntax for the directive is similar to `mx-connection-limit` directive, except that it is uses IP addresses (CIDR notation).

**Syntax:**

```
ip-connection-limit CIDR N
```

where CIDR is the IP address specification of the destination, and N is the number of simultaneous connections allowed to each MX in the group of IP addresses that match the CIDR.

**Example:**

```
ip-connection-limit 173.194/16 50
ip-connection-limit 104.47.2.33 1
```

### 3.3.1.42 use-proxy

Specifies a comma separate list of proxy servers (see [proxy](#) for more details) that virtual MTAs can use when connecting to a remote destination.

When this directive is defined at global scope, its value is inherited by all the virtual MTAs in the system unless they override it specifically.

The proxy servers in the list are used in the order in which they are defined. The names in the list are strictly used for failover only - the second proxy on the list is used only if the connection to the first proxy fails, the third proxy is used only when connection to both first and second proxy have failed, and so on.

Every proxy server mentioned in this directive must be defined in the configuration with a [proxy](#) tag.

**Syntax:**

```
use-proxy PROXY-SERVER-LIST
```

**Default:** None

**Example:**

```
# global level
use-proxy mta-proxy-a

<proxy mta-proxy-a>
  server 10.1.2.3:5000
  client 16.70.55.233 client.hostname.com
</proxy>

<proxy mta-proxy-b>
  server mta-proxy-b.internal.hostname.com:5000
  client 16.80.77.192 client2.hostname.com
</proxy>

<virtual-mta v1>
  # inherits global "use-proxy"
</virtual-mta>

<virtual-mta v2>
  # overrides global "use-proxy"
  use-proxy mta-proxy-b
</virtual-mta>
```

### 3.3.2 <spool> Directives

Each `<spool>` specifies a directory for the message spool, where PowerMTA stores the message files while queued for delivery. While at least one spool directory is required, additional entries can be added for greater performance, for example by distributing the I/O load over several disks.

**Syntax:**

```
<spool DIRECTORY>
</spool>
```

**Example: (Linux)**

```
<spool /var/spool/pmta>
  priority 50
  delete-file-holders false
</spool>
```

**Example: (Windows)**

```
<spool C:\pmta\spool>
  priority 50
  delete-file-holders false
</spool>
```

#### 3.3.2.1 delete-file-holders

Determines whether PowerMTA should delete any "file holders" (which are files temporarily left behind by PowerMTA in order to save on disk I/O) in the spool directory after a message has been delivered.

**Default:** false

**Example:**

```
<spool /var/spool/pmta>
  delete-file-holders true
</spool>
```

**3.3.2.2 deliver-only**

Indicates whether PowerMTA should only deliver messages from a spool directory, without writing new spool files to this directory.

This can be useful during failure recovery by allowing you to attach a spool directory from a failed server and having the messages in that directory be delivered without writing any new spool files to that directory.

**Default:** false

**Example:**

```
<spool /var/spool/pmta>
  deliver-only true
</spool>
```

**3.3.2.3 priority**

Sets the priority for a spool location, from 100 (highest) to 1 (lowest). Spools with higher priority are always used, unless they are full, at which point lower priority spool locations are considered. If two or more spools have same priority, the spools are selected in round-robin fashion.

**Syntax:**

```
<spool DIRECTORY>
  priority N
</spool>
```

where N is a positive integer between 1 and 100.

**Default:** 1

**Example:**

```
<spool /var/spool/pmta/primary>
  priority 10
</spool>

<spool /var/spool/pmta/secondary>
  priority 5
</spool>
```

**3.3.2.4 min-free-space**

Instructs PowerMTA to stop accepting email into the spool if the space available on the file system falls below the amount specified. When available disk space increases above that level, messages are accepted again.

On Unix, this is based on the space available to non-root users.

**Syntax:**

```
min-free-space { N{b|k|m|g|t} | 0 }
```

where N is a positive integer followed by a size multiplier (B for byte, K for Kilobyte, M for Megabyte, G for Gigabyte or T for Terabyte).

**Default:** 0

**Example:**

```
<spool /var/spool/pmta>
  min-free-space 1G
</spool>
```

**3.3.3 <bounce-processor> Directives**

The <bounce-processor> directive creates a bounce processor in PowerMTA. PowerMTA will then look for bounce emails when processing incoming messages. See [here](#) for more details on usage.

**Example:**

```
<bounce-processor>
  deliver-unmatched-email yes          # default: yes
  deliver-matched-email yes            # default: no
  forward-unmatched-to auto-feedback@port25.com
  forward-errors-to auto-feedback@port25.com
  <address-list>
    domain domain.to.filter            # whole domain
    address /regex@domain.to.filter/   # regex
  </address-list>
</bounce-processor>
```

**3.3.3.1 deliver-unmatched-email**

Controls whether any emails that were not matched by the bounce processor should be delivered. If set to false, unmatched emails are discarded.

**Default:** true

**Example:**

```
<bounce-processor>
```

```
    deliver-unmatched-email false
</bounce-processor>
```

---

#### 3.3.3.2 deliver-matched-email

Determines whether any emails that were matched by the bounce processor should be delivered. If set to `false`, matched emails are discarded.

**Default:** `false`

**Example:**

```
<bounce-processor>
    deliver-matched-email true
</bounce-processor>
```

---

#### 3.3.3.3 deliver-email

**Deprecated** Use `deliver-matched-email`/`deliver-unmatched-email` instead.

---

#### 3.3.3.4 forward-unmatched-to

This directive allows for forwarding messages unmatched by the bounce processor to an external address.

**Syntax:**

```
forward-unmatched-to EMAIL-ADDRESS
```

**Example:**

```
<bounce-processor>
    forward-unmatched-to  foo@port25.com
</bounce-processor>
```

---

#### 3.3.3.5 forward-errors-to

This directive configures forwarding for messages that the bounce processor failed to parse.

**Syntax:**

```
forward-errors-to EMAIL-ADDRESS
```

**Example:**

```
<bounce-processor>
    forward-errors-to  foo@port25.com
</bounce-processor>
```

---

### 3.3.4 <feedback-loop-processor> Directives

The `<feedback-loop-processor>` directive creates a feedback loop processor in PowerMTA. PowerMTA will then look for FBL emails when processing incoming messages. See [here](#) for more details on usage.

**Example:**

```
<feedback-loop-processor>
    deliver-unmatched-email no      # default: yes
    deliver-matched-email yes      # default: no
    forward-unmatched-to auto-feedback@port25.com
    forward-errors-to auto-feedback@port25.com
    <address-list>
        address /fbl@fbl.yourdomain.com/
    </address-list>
</feedback-loop-processor>
```

---

#### 3.3.4.1 deliver-unmatched-email

Controls whether any emails that were not matched by the feedback-loop processor should be delivered. If set to `false`, unmatched emails are discarded.

**Default:** `true`

**Example:**

```
<feedback-loop-processor>
    deliver-unmatched-email false
</feedback-loop-processor>
```

---

#### 3.3.4.2 deliver-matched-email

Determines whether any emails that were matched by the feedback-loop processor should be delivered. If set to `false`, matched emails are discarded.

**Default:** `false`

**Example:**

```
<feedback-loop-processor>
    deliver-matched-email true
</feedback-loop-processor>
```

---

### 3.3.4.3 deliver-email

**Deprecated** Use `deliver-matched-email/deliver-unmatched-email` instead.

---

### 3.3.4.4 forward-unmatched-to

This directive allows for forwarding messages unmatched by the feedback loop processor to an external address.

**Syntax:**

```
forward-unmatched-to EMAIL-ADDRESS
```

**Example:**

```
<feedback-loop-processor>
  forward-unmatched-to foo@port25.com
</feedback-loop-processor>
```

---

### 3.3.4.5 forward-errors-to

This directive configures forwarding for messages that the feedback loop processor failed to parse.

**Syntax:**

```
forward-errors-to EMAIL-ADDRESS
```

**Example:**

```
<feedback-loop-processor>
  forward-errors-to foo@port25.com
</feedback-loop-processor>
```

---

## 3.3.5 <address-list> Directives

An <address-list> can be used to organize domains, or email addresses, or regular expressions into a single list. If the list is named, it can be used with directives such as `suppression-list`. Unnamed lists can be used in <bounce-processor> and <feedback-loop-processor> tags.

**Example:**

```
# Named <address-list>
<address-list foo>
  domain foo.com
  address /^bounce@/
</address-list>

# Unnamed <address-list> inside a <bounce-processor>
<bounce-processor>
  <address-list>
    domain foo.com
    address /^bounce@/
  </address-list>
</bounce-processor>
```

### 3.3.5.1 address

An email address, or a regular expression that matches an email address.

**Syntax:**

```
address { EMAIL-ADDRESS | REGEX }
```

**Example:**

```
<address-list foo>
  address abc@xyz.com
  address /^bounce@/
</address-list>
```

---

### 3.3.5.2 address-file

Includes into the <address-list> all addresses in the file, where each line in the file is either the complete email address, or just the local part of an address.

**Syntax:**

```
address-file FILENAME
```

**Example:**

```
<address-list foo>
  address-file /etc/pmta/suppression-address-list.txt
</address-list>
```

where `/etc/pmta/suppression-address-list.txt` contains:

```
foo
bar
baz@quux.com
```

---

### 3.3.5.3 domain

A domain name.

**Syntax:**

```
domain DOMAIN
```

where DOMAIN is a complete domain name.

**Example:**

```
<address-list foo>
  domain bounce.foo.com
  domain fbl.bar.com
</address-list>
```

**3.3.6 <email-patterns> Directives**

<email-pattern> directives can be used to configure regular expressions that will be matched against incoming remote bounces or feedback loop messages in order to extract useful information from them. This feature is useful when the remote MTA does not use standard DSN/ARF formats for remote bounces/feedback loop messages they return. See [Handling non-standard DSN/FBL reports](#) for more details.

<email-patterns> can only be used with <bounce-processor> and <feedback-loop-processor> tags.

**Example:**

```
<email-patterns>
  body-pattern-file /etc/pmta/report-pattern.txt
  log-match-info true
</email-patterns>
```

**3.3.6.1 body-pattern-file**

This directive allows you to configure a file that contains the pattern matching the body of a non-standard FBL/DSN bounce report. See [Handling non-standard DSN/FBL reports](#) for more details.

**Example:**

```
<email-patterns>
  body-pattern-file /etc/pmta/report-pattern.txt
</email-patterns>
```

Sample content of the file /etc/pmta/report-pattern.txt

```
Email from (?<dlvSourceIp>[0-9\\.]+)
```

**3.3.6.2 log-match-info**

If set to true, a message will be printed to log file with a summary about the info that was extracted from a remote bounce/feedback loop message that was matched, along with the name of the pattern that matched it. Useful for debugging <email-patterns> when you have multiple email patterns configured.

**Example:**

```
<email-patterns>
  body-pattern-file /etc/pmta/report-pattern-1.txt
  body-pattern-file /etc/pmta/report-pattern-2.txt
  body-pattern-file /etc/pmta/report-pattern-3.txt
  body-pattern-file /etc/pmta/report-pattern-4.txt
  log-match-info true
</email-patterns>
```

Sample log message when log-match-info true is set

```
matched body-pattern defined in file:/etc/pmta/report-pattern-3.txt
```

**3.3.7 <source> Directives**

Directives entered within a <source> entry apply to SMTP connections from the given source. Per-source directives can be specified in two ways: by IP address and by name.

Each <source> entry to be applied by IP address specifies either an IP address in dotted decimal notation (like 10.0.0.1), or a range of IP addresses in CIDR notation (like 10/8).

A <source> entry to be applied by name specifies its name in the entry, like in <source S1>.

The example below shows entries for a specific IP, for a range of IPs in CIDR notation, and for the global 0/0 wildcard. More specific settings are used in favor of the more general if there is a conflict.

```
<source 10.0.0.5> # matches 10.0.0.5 only
  allow-mailmerge yes
</source>

<source 10.0.0.0/8> # matches 10.*
  always-allow-relaying yes
</source>

<source 0/0> # matches all
  always-allow-relaying no
  allow-mailmerge no
</source>
```

Note that PowerMTA merges ("inherits") settings in source entries like it does for the domain entries, allowing you to provide defaults in a more generic entry (such as 0/0). After the inheritance, this is how PowerMTA sees the <source> settings:

```
<source 10.0.0.5> # matches 10.0.0.5 only
  always-allow-relaying yes
  allow-mailmerge yes
```

```

</source>

<source 10.0.0.0/8> # matches 10.*
    always-allow-relaying yes
    allow-mailmerge no
</source>

<source 0/0> # matches all
    always-allow-relaying no
    allow-mailmerge no
</source>

```

Named entries beginning with { are reserved and have special meaning: <source {pickup}> can be used to modify pickup directory handling and <source {auth}> can be used to specify defaults for any authenticated SMTP user.

Non-reserved named entries are used to override the settings obtained based on the source IP address. They apply only when specifically requested, like in the source=... parameter of the smtp-listener directive or <smtp-user>'s source directive.

#### 3.3.7.1 add-date-header

Specifies whether PowerMTA should add a Date header if missing. If a Date header is present, it is not overridden unless using the override parameter.

**Note:** Automatic addition of the Date header is not supported for mailmerge messages. When submitting mailmerge messages, you can use the \*date variable to easily accomplish the same.

##### Syntax:

```
add-date-header { no | yes | override }
```

**Default:** no

##### Example:

```

<source 0/0>
    add-date-header yes
</source>

```

#### 3.3.7.2 date-header-time

Specifies whether PowerMTA should set the Date header to the time the message was received and queued for delivery, or the time that the message was actually delivered. The directive add-date-header needs to be set to yes or override when using this directive.

##### Syntax:

```
date-header-time { reception | delivery }
```

**Default:** reception

##### Example:

```

<source 0/0>
    add-date-header yes
    date-header-time delivery
</source>

```

#### 3.3.7.3 bcc

This directive takes an email address as a value, and if set, causes each message received from that source to be BCC'ed to the given email address. This can be used for creating a copy of every message for archival purposes.

##### Syntax:

```
bcc EMAIL-ADDRESS
```

##### Example:

```

<source 0/0>
    bcc foo@bar.com
</source>

```

#### 3.3.7.4 add-message-id-header

Specifies whether PowerMTA should add a Message-Id header if missing. If a Message-Id header is present, it is not overridden.

**Default:** false

##### Example:

```

<source 0/0>
    add-message-id-header true
</source>

```

#### 3.3.7.5 add-received-header

Specifies whether PowerMTA should add a Received header upon reception of an email.

**Default:** true

##### Example:

```

<source 0/0>
    add-received-header false
</source>

```

### 3.3.7.6 allow-auth

Specifies whether PowerMTA should allow usage of the AUTH SMTP command.

**Default:** true

**Example:**

```
<source 0/0>
  allow-auth false
</source>
```

### 3.3.7.7 smtp-max-auth-failure-rate

Specifies after how many unsuccessful authentication attempts in the given time PowerMTA should refuse further authentication attempts from this source. This is used to defend against password-guessing attacks.

For example, setting the rate limit to 5/min allows only five failed authentication attempts per minute.

Once a source is locked, users can only re-attempt authentication after the configured time period, counted from the time of the first failed attempt. For example, if the max auth failure rate is set to 3/5m and someone tries to authenticate unsuccessfully at 10:00, 10:01, and 10:02, authentication will be refused till 10:05.

If this directive is set to unlimited (the default) or 0, PowerMTA will not block the account on failed authentication attempts.

For this directive to be effective, you must enforce authentication with the `require-auth` directive.

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Syntax:**

```
smtp-max-auth-failure-rate { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<source 0/0>
  smtp-max-auth-failure-rate 5/m
</source>
```

### 3.3.7.8 remove-header

Specifies a header that PowerMTA should remove from the incoming or outgoing message. The directive takes a comma separated list of headers if there is more than one header to be removed. The header is removed at delivery, not when the message is queued to disk. The directive may be used in a `<source>` or `<domain>`, but is not required to be defined in both.

Use empty string ("") to clear any inherited value.

**Syntax:**

```
remove-header HEADER-LIST
```

**Example:**

```
<source 0/0>
  remove-header nameOfYourCustomHeader,nameOfOtherHeader
</source>
```

### 3.3.7.9 require-auth

Allows for requiring authentication before any emails are received via SMTP (local or not).

**Default:** false

**Example:**

```
<source 0/0>
  require-auth true
</source>
```

### 3.3.7.10 allow-mailmerge

Specifies whether PowerMTA should allow usage of the mailmerge SMTP extensions from the given source.

**Default:** false

**Example:**

```
<source 0/0>
  allow-mailmerge true
</source>
```

### 3.3.7.11 allow-unencrypted-plain-auth

Specifies whether PowerMTA should allow usage of the PLAIN SASL authentication mechanism in unencrypted connections.

**Default:** false

**Example:**

```
<source 0/0>
  true
```

```
</source>
```

### 3.3.7.12 allow-starttls

This directive controls whether PowerMTA supports the use of STARTTLS extension with clients that connect to it for submitting email.

**Default:** false

**Example:**

```
<source 0/0>
  allow-starttls true
</source>
```

### 3.3.7.13 require-starttls-before-auth

Setting this directive to true will enforce the requirement that connecting clients must perform STARTTLS before issuing any AUTH commands during the session.

**Default:** false

**Example:**

```
<source 0/0>
  require-starttls-before-auth true
</source>
```

### 3.3.7.14 broken-auth-clients

**\*\* Deprecated \*\***

### 3.3.7.15 verp-default

If you enable VERP by setting verp-default to yes or true, PowerMTA will create a customized SMTP MAIL FROM address for the message based on the original SMTP MAIL FROM address and the RCPT TO addresses in the message. Since one can then identify the original RCPT TO address in this new encoded SMTP MAIL FROM address, this may make tracking bounces easier for you. The format is fixed, and can be seen in the following example:

```
Original SMTP MAIL FROM: jsmith@example.com
SMTP RCPT TO: jdoe@yahoo.com
VERPed SMTP MAIL FROM: jsmith-jdoe@yahoo.com@example.com
```

**Default:** no

**Example:**

```
<source 0/0>
  verp-default true
</source>
```

### 3.3.7.16 check-dkim-inbound

Specifies whether or not PowerMTA should enable checking DKIM signatures on inbound messages. If the message has DKIM signature header in it, PowerMTA will add an Authentication-Results header stating the results.

**Default:** false

**Example:**

```
<source 0/0>
  check-dkim-inbound true
</source>
```

### 3.3.7.17 trace-dkim-check

Allows selecting whether the trace data (currently including all the DNS data looked up) is included in the Authentication-Results headers for the dkim check.

**Default:** false

**Example:**

```
<source 0/0>
  trace-dkim-check true
</source>
```

### 3.3.7.18 trace-spf-check

Requests inclusion of trace data (currently including all the DNS data looked up) for the SPF check in the Authentication-Results header.

**Default:** false

**Example:**

```
<source 0/0>
  trace-spf-check true
</source>
```

### 3.3.7.19 check-spf-inbound

Specifies whether to perform an SPF check on inbound messages. PowerMTA adds an Authentication-Results header with the results of the test.

**Default:** false

**Example:**

```
<source 0/0>
  check-spf-inbound true
</source>
```

---

#### 3.3.7.20 check-spf-inbound-best-guess

**Deprecated** This directive is deprecated, do not use.

---

#### 3.3.7.21 check-spf-inbound-max-void-lookups

Specifies the maximum number of “void” (empty or NXDOMAIN) responses to allow before failing the SPF check with “permerror”. A low number, such as 2, helps minimize certain denial-of-service attacks, but may also cause errors evaluating complicated SPF policies.

**Default:** 2

**Example:**

```
<source 0/0>
  check-spf-inbound-max-void-lookups 5
</source>
```

---

#### 3.3.7.22 check-iprev-inbound

Determines if PowerMTA performs an inbound DNS PTR check on the IP and EHLO hostname used for the connection.

**Default:** false

**Example:**

```
<source 0/0>
  check-iprev-inbound true
</source>
```

---

#### 3.3.7.23 trace-iprev-check

Requests inclusion of trace data (currently including all the DNS data looked up) for the DNS PTR check in the Authentication-Results header.

**Default:** false

**Example:**

```
<source 0/0>
  trace-iprev-check true
</source>
```

---

#### 3.3.7.24 reject-iprev-check-temperror

If set to true, and DNS PTR check for a message returns a temp error (such as DNS lookup failure), PowerMTA will reject the message.

**Default:** false

**Example:**

```
<source 0/0>
  reject-iprev-check-temperror true
</source>
```

---

#### 3.3.7.25 reject-iprev-check-permerror

If set to true, and DNS PTR check for a message returns a perm error, PowerMTA will reject the message.

**Default:** false

**Example:**

```
<source 0/0>
  reject-iprev-check-permerror true
</source>
```

---

#### 3.3.7.26 reject-iprev-check-fail

If set to true, and DNS PTR check for a message results in a failure, PowerMTA will reject the message.

**Default:** false

**Example:**

```
<source 0/0>
  reject-iprev-check-fail true
</source>
```

---

#### 3.3.7.27 suppress-local-dsn

Disables the generation of DSN (bounce, delay or delivery) reports for messages received from this <source>.

**Default:** false

**Example:**

```
<source 0/0>
  suppress-local-dsn true
</source>
```

---

**3.3.7.28 dsn-return-default**

Specifies the default for the DSN RET parameter, i.e., whether the full message body or only its headers should be returned in a DSN delivery report. system specifies that the system default should be used. This directive is overridden by the use of "dsn-format plain-text".

**Syntax:**

```
dsn-return-default { headers | full | system }
```

**Default:** system

**Example:**

```
<source 0/0>
  dsn-return-default headers
</source>
```

---

**3.3.7.29 default-virtual-mta**

Specifies the default VirtualMTA (or VirtualMTA pool) to select for all messages received from the source. This directive is used only when other higher priority methods of selecting a VirtualMTA (such as the use of X-Virtual-Mta header) are not applicable.

The special value `by-smtp-source-ip` can be used in place of a VirtualMTA name. When configured in this way, on an inbound SMTP connection PowerMTA will use as the default VirtualMTA whose "smtp-source-host" setting equals the destination IP address on the inbound connection.

This directive may also designate the hostname used for SMTP greeting messages. See [host-name](#) definition in [Global Directives](#) for more information.

**Syntax:**

```
default-virtual-mta { VMTA-NAME | by-smtp-source-ip }
```

**Example:**

```
<source 127.0.0.1>
  default-virtual-mta mta1 # mta1 is a virtual MTA
</source>

<source 0/0>
  default-virtual-mta by-smtp-source-ip
</source>
```

---

**3.3.7.30 log-tls**

Specifies whether PowerMTA should log information about negotiation of a TLS session with the remote client during STARTTLS.

**Default:** false

**Example:**

```
<source 0/0>
  log-tls true
</source>
```

---

**3.3.7.31 reject-invalid-virtual-mta**

Specifies that PowerMTA should reject messages when an invalid or null VirtualMTA is selected.

**Default:** true

**Example:**

```
<source 0/0>
  reject-invalid-virtual-mta true
</source>
```

---

**3.3.7.32 disconnect-on-transient-error**

Specifies that PowerMTA should close connections instead of sending transient SMTP errors. This is useful in situations where the feeding software cannot process 4xx errors.

**Default:** false

**Example:**

```
<source 0/0>
  disconnect-on-transient-error true
</source>
```

---

**3.3.7.33 hide-message-source**

Specifies that PowerMTA should attempt to hide the source of the message while delivering email from this source. Currently, this just means that the name and IP address of the MTA from which PowerMTA received this message will not be included in the Received: header added. Hiding the message's source may be desirable, for example, for security purposes, to avoid revealing details from the internal network from which the message was submitted.

**Default:** false

**Example:**

```
<source 0/0>
  hide-message-source true
</source>
```

#### 3.3.7.34 max-message-size

Specifies the maximum size allowed in messages received through SMTP. If set to unlimited, no explicit limit is enforced.

**Syntax:**

```
max-message-size { N{b|k|m|g|t} | 0 | unlimited }
```

where N is a positive integer followed by a size multiplier (B for byte, K for Kilobyte, M for Megabyte, G for Gigabyte or T for Terabyte).

An unsigned non-zero number is also accepted as the maximum size in bytes, but that syntax is deprecated.

**Default:** unlimited

**Example:**

```
<source 0/0>
  max-message-size 40K
</source>
```

#### 3.3.7.35 max-message-hops

Specifies the maximum number of "Received" headers that can be present in a message that PowerMTA accepts.

**Syntax:**

```
max-message-hops N
```

**Default:** 100

**Example:**

```
<source 0/0>
  max-message-hops 50
</source>
```

#### 3.3.7.36 max-rcpt-per-message

Specifies the maximum number of recipients accepted in each message. If set to zero or unlimited, no explicit limit is enforced.

**Syntax:**

```
max-rcpt-per-message { N | unlimited }
```

**Default:** 0

**Example:**

```
<source 0/0>
  max-rcpt-per-message 10
</source>
```

#### 3.3.7.37 too-many-rcpts-fails-message

Specifies how PowerMTA should enforce the max-rcpt-per-message limit. If enabled and too many recipients are passed for a message, rather than reject the recipient(s) over the limit (RCPT SMTP command), PowerMTA rejects the entire message (DATA command).

**Default:** false

**Example:**

```
<source 0/0>
  too-many-rcpts-fails-message true
</source>
```

#### 3.3.7.38 suppression-lists

Specifies the suppression-list(s) to be used for suppressing addresses. Multiple lists may be specified, separating each list name with a comma. See [Address Suppression Lists](#) for more information.

**Syntax:**

```
suppression-lists SUPPRESSION-LIST-NAME
```

**Example:**

```
<address-list foo>
  address jdoe@yahoo.com
</address-list>

<source 0/0>
  suppression-lists foo
</source>
```

#### 3.3.7.39 pattern-list

Specifies that messages received from the source are to be matched against the given pattern list. The pattern list referenced must precede the source definition in the configuration file. See [pattern-list Directives](#) for more details.

**Syntax:**

```
pattern-list PATTERN-LIST-NAME
```

**Example:**

```
<pattern-list foo>
</pattern-list>

<source 0/0>
  pattern-list foo
</source>
```

### 3.3.7.40 process-x-envid

Specifies whether PowerMTA should process x-envid headers. If set to true and one such header is included, PowerMTA will set the message's (DSN) envelope ID to the header's body, as well as remove the header from the message. If set to false, PowerMTA will ignore this header, leaving it in the message if present. While the envelope ID can be set by using the ENVID parameter in the SMTP MAIL command, this directive is useful when the software you use to submit the messages does not allow you to control it.

**Default:** false

**Example:**

```
<source 0/0>
  process-x-envid true
</source>
```

### 3.3.7.41 process-x-dkim-options

Specifies whether PowerMTA should process x-dkim-options header. If set to true and one such header is included, PowerMTA will use any values in the header for signing messages with DKIM if DKIM signing is configured. If set to false, PowerMTA will ignore this header, leaving it in the message if present.

**Default:** false

**Example:**

```
<source 0/0>
  process-x-dkim-options true
</source>
```

### 3.3.7.42 process-x-dkim-key

Specifies whether PowerMTA should process X-DKIM-Key headers. If set to true, PMTA will extract the private key specified in PEM format in the header X-DKIM-Key, use it for DKIM signing the message, and remove that header from the message prior to transmission. Note that if process-x-dkim-key isn't set, PowerMTA does not remove that header; you should test your setup to avoid leaking private keys. Also, this feature requires specifying the DKIM key selector (s=) in a X-DKIM-Options header and thus also enabling the per-source process-x-dkim-options directive.

**Default:** false

**Example:**

```
<source 0/0>
  process-x-dkim-key true
</source>
```

### 3.3.7.43 retain-x-dkim-key

Specifies whether PowerMTA should retain a processed x-dkim-key header in the message. If set to true, PowerMTA will keep the x-dkim-key header in the message when sending it on, otherwise that header will be removed. While this option is generally used in conjunction with the process-x-dkim-key option, they are completely independent of each other. Both directives can be set to false for example, in which case PowerMTA will ignore the header with regards to processing however still removing the header before delivery.

**Note:** If this directive is set to true, the private key will be included in the email when PowerMTA delivers it. This is a security risk so it is strongly recommended to keep this directive set to false.

**Default:** false

**Example:**

```
<source 0/0>
  retain-x-dkim-key false
</source>
```

### 3.3.7.44 jobid-prefix

Allows one to have PowerMTA add a custom prefix to the data supplied in the x-job: header in the message. A special reserved {auth} prefix has PowerMTA adding the actual authenticated user as the prefix, allowing for example, a relay service to more easily track messages sent through by PowerMTA by each authenticated user.

**Syntax:**

```
jobid-prefix STRING
```

**Example:**

```
<source 0/0>
  jobid-prefix customer1
</source>
```

#### 3.3.7.45 process-x-job

Specifies whether PowerMTA should process x-job headers. If set to true and one such header is included, PowerMTA will set the job ID for the message to the job ID given in the x-job header. The job ID must not include any non-printable or white space characters. A message is supposed to include at most one x-job header. If using Mailmerge, you must use the \*jobid variable, and not the x-job header.

Sample header:

```
x-job: abc123
```

**Default:** false

**Example:**

```
<source 0/0>
  process-x-job true
</source>
```

#### 3.3.7.46 jobid-header

Allows setting an alternate header for which the message's jobid is set in place of using x-job. If other ways to set a jobid (such as x-job processing, if enabled, or \*jobid mail merge variable) apply, they take precedence.

**Example:**

```
<source 0/0>
  jobid-header Campaign-Id
</source>
```

#### 3.3.7.47 retain-x-job

Specifies whether PowerMTA should retain a processed x-job header in the message. If set to true, PowerMTA will keep the x-job header in the message when sending it on, otherwise that header will be removed. This option is only used if process-x-job is set to true and the message contains an x-job header.

**Default:** false

**Example:**

```
<source 0/0>
  retain-x-job true
</source>
```

#### 3.3.7.48 process-x-virtual-mta

Specifies whether PowerMTA should process x-virtual-mta headers. If set to true and one such header is included, PowerMTA will select the VirtualMTA specified in the header's body for the message, as well as remove the header from the message. If set to false, PowerMTA will ignore this header, leaving it in the message if present. You should enable this directive for the source IP addresses from which you feed messages if you would like to use the x-virtual-mta header to select which VirtualMTA to use.

**Default:** false

**Example:**

```
<source 0/0>
  process-x-virtual-mta true
</source>
```

#### 3.3.7.49 retain-x-virtual-mta

Specifies whether PowerMTA should retain a processed x-virtual-mta header in the message. If set to true, PowerMTA will keep the x-virtual-mta header in the message when sending it on, otherwise that header will be removed. This option is only used if process-x-virtual-mta is set to true and the message contains an x-virtual-mta header.

**Default:** false

**Example:**

```
<source 0/0>
  retain-x-virtual-mta true
</source>
```

#### 3.3.7.50 add-x-virtual-mta-header

Requests that an "X-Virtual-MTA" header be added to the message with the name of the Virtual MTA from which it was delivered, if no such header is already present.

**Default:** false

**Example:**

```
<source 0/0>
  add-x-virtual-mta-header true
</source>
```

#### 3.3.7.51 source-group

Specifies the name of a source group to associate the given source to, allowing linking sources to groups (and thus grouping sources). The directive allows for specifying certain source configuration items whose functionality require them to be grouped by name. The shared resources will be defined in the <source-group> tag.

**Syntax:**

```
source-group SOURCE-GROUP-NAME
```

**Example:**

```
<source-group localSources>
  max-smtp-in 10
  reserved-smtp-in 5
</source-group>

<source 192.168.0.30>
  smtp-service yes
  always-allow-relaying yes
  source-group localSources
</source>

<source 192.168.0.40>
  smtp-service yes
  always-allow-relaying yes
  source-group localSources
</source>
```

### 3.3.7.52 smtp-data-timeout

Specifies the amount of time that PowerMTA will wait for data from the remote end during data reception phase (after DATA/BDAT command) of the message. Applies to inbound connections only.

**Syntax:**

```
smtp-data-timeout TIME-INTERVAL
```

**Default:** 10m

**Example:**

```
<source 0/0>
  smtp-data-timeout 5m
</source>
```

### 3.3.7.53 smtp-command-timeout

Specifies the amount of time that PowerMTA will wait for data to be sent from a remote end during inbound SMTP command exchanges.

**Syntax:**

```
smtp-command-timeout TIME-INTERVAL
```

**Default:** 10m

**Example:**

```
<source 0/0>
  smtp-command-timeout 5m
</source>
```

### 3.3.7.54 process-x-schedule

Specifies whether PowerMTA should process the x-schedule header if included in the message. If set to true and one such header is included, PowerMTA will respect the schedule defined in the header, and only attempt to deliver messages per the schedule. If set to false, PowerMTA will ignore this header and attempt to deliver normally, without any time based schedule.

**Default:** false

**Example:**

```
<source 0/0>
  process-x-schedule true
</source>
```

### 3.3.7.55 retain-x-schedule

Specifies whether PowerMTA should leave the x-schedule header in the message when messages are delivered (when using PowerMTA's Schedule Delivery Control feature). While this option is generally used in conjunction with the process x-schedule header option, they are completely independent of each other. Both can be set to false for example, which has PowerMTA ignoring the header with regards to processing however still removing the header before delivery.

**Default:** false

**Example:**

```
<source 0/0>
  retain-x-schedule true
</source>
```

### 3.3.7.56 retain-x-dkim-options

Specifies whether PowerMTA should retain the x-dkim-options header when sending a message to a remote mail server.

**Default:** false

**Example:**

```
<source 0/0>
  retain-x-dkim-options true
</source>
```

---

### 3.3.7.57 allow-chunking

Specifies whether PowerMTA should allow usage of the CHUNKING SMTP extension (i.e., the BDAT command) on inbound connections from the source.

**Default:** true

**Example:**

```
<source 0/0>
  allow-chunking true
</source>
```

---

### 3.3.7.58 allow-smtputf8

Specifies whether one can submit internationalized email addresses per RFC6531 to PowerMTA from the connecting IP address. When set to yes (or true), PowerMTA will list "SMTPUTF8" in the list of extended SMTP commands supported for the connecting IP, allowing the submitter to use internationalized email addresses in the SMTP envelope. Note that if the remote gateway for the domain does not advertise support for SMTPUTF8, PowerMTA will subsequently bounce the message(s) for the recipient out of the queue.

**Default:** true

**Example:**

```
<source 0/0>
  allow-smtputf8 true
</source>
```

---

### 3.3.7.59 smtp-service

Specifies whether to allow access to the SMTP service to connections from the given source. If set to false, PowerMTA issues a 5XX SMTP greeting like

```
521 mta1.foo.com does not accept mail from you
```

instead of the regular one:

```
220 mta1.foo.com (PowerMTA v5.0r1) ESMTP service ready
```

Note that since the default is to grant SMTP service, if you only wish to restrict service to specific sources you must specify that in the 0/0 entry, as in the example below.

**Default:** true

**Example:**

```
<source 10.0.0.1> # we feed from 10.0.0.1
  smtp-service yes
  always-allow-relaying yes # normally desirable for feeder sources
</source>

<source 0/0>
  smtp-service no
</source>
```

---

### 3.3.7.60 log-connections

Instructs PowerMTA to write an entry in the log at the beginning and end of each connection.

**Default:** false

**Example:**

```
# log *all* incoming connections
<source 0/0>
  log-connections yes
</source>
```

---

### 3.3.7.61 log-commands

Instructs PowerMTA to log the full SMTP protocol exchanges between itself and the receiving or sending mailer. Since it displays at which point of the delivery process an error occurs, enabling this directive is generally the first and most useful step in debugging connectivity problems.

Sample Log output:

```
2003-06-29 15:32:00 ( 20)starting aol.com
2003-06-29 15:32:00 ( 20)connecting to yh.mx.aol.com (205.188.157.1)
2003-06-29 15:32:00 ( 20)>>> 220-rly-yh01.mx.aol.com ESMTP relay_in.5; Thu, 29 Jun 2003 09:27:58
-0400
2003-06-29 15:32:00 ( 20)>>> 220-America Online (AOL) and its affiliated companies do not
2003-06-29 15:32:00 ( 20)>>> 220- authorize the use of its proprietary computers and computer
2003-06-29 15:32:00 ( 20)>>> 220- networks to accept, transmit, or distribute unsolicited bulk
2003-06-29 15:32:00 ( 20)>>> 220 email sent from the internet.
2003-06-29 15:32:00 ( 20)<<< EHLO hazmat.port25.com
2003-06-29 15:32:00 ( 20)>>> 250-rly-yh01.mx.aol.com hazmat.port25.com
2003-06-29 15:32:00 ( 20)>>> 250 HELP
...
```

**Default:** false

**Example:**

```
<source 0/0>
  log-commands true
```

---

```
</source>
```

### 3.3.7.62 log-data

This directive is useful for debugging protocol or interoperability problems between the mailers, for it logs every byte sent, both in ASCII and hexadecimal. This would help show things like whether your messages may be finishing its lines with just LF (line feeds) instead of the CRLF (carriage return, line feed) pair as prescribed by the standards, and the hex data allows you to identify non-printable characters (such as CR and LF) based on their hex codes.

Sample Log output:

```
2003-06-29 09:53:59 ( 19)starting yahoo.com
2003-06-29 09:53:59 ( 19)connecting to mx1.mail.yahoo.com (128.11.68.155)
2003-06-29 09:54:04 ( 19)>>> rd 53
2003-06-29 09:54:04 ( 19)3232302059536D7470206D7461313334 220 YSmtp mta134
2003-06-29 09:54:04 ( 19)2E6D61696C2E7961686F6F2E636F6D20 .mail.yahoo.com
2003-06-29 09:54:04 ( 19)45534D54502073657276696365207265 ESMTP service re
2003-06-29 09:54:04 ( 19)6164790D0A ady..
2003-06-29 09:54:04 ( 19)<<< wr 24
2003-06-29 09:54:04 ( 19)45484C4F2068617A6D61742E706F7274 EHLO hazmat.port
2003-06-29 09:54:04 ( 19)32352E636F6D0D0A 25.com..
2003-06-29 09:54:04 ( 19)>>> rd 75
2003-06-29 09:54:04 ( 19)3235302D6D74613133342E6D61696C2E 250-mta134.mail.
2003-06-29 09:54:04 ( 19)7961686F6F2E636F6D0D0A3235302D38 yahoo.com..250-8
2003-06-29 09:54:04 ( 19)4249544D494D450D0A3235302D53495A BITIME..250-SIZ
2003-06-29 09:54:04 ( 19)4520333134353732380D0A3235302050 E 3145728..250 P
2003-06-29 09:54:04 ( 19)4950454C494E494E470D0A IPELINING..
2003-06-29 09:54:04 ( 19)<<< wr 43
2003-06-29 09:54:04 ( 19)4D41494C2046524F4D3A3C696E666F40 MAIL FROM:<info@
2003-06-29 09:54:04 ( 19)706F727432352E636F6D3E20424F4459 port25.com> BODY
2003-06-29 09:54:04 ( 19)3D384249544D494D450D0A =8BITIME..
2003-06-29 09:54:04 ( 19)>>> rd 33
2003-06-29 09:54:04 ( 19)3235302073656E646572203C696E666F 250 sender <info
2003-06-29 09:54:04 ( 19)40706F727432352E636F6D3E206F6B0D @port25.com> ok.
2003-06-29 09:54:04 ( 19)0A .
2003-06-29 09:54:04 ( 19)<<< wr 29
...
```

Default: false

Example:

```
<source 0/0>
log-data true
</source>
```

### 3.3.7.63 allow-empty-x-virtual-mta

Specifies PowerMTA should allow an empty string value for the x-virtual-mta header. If set to yes and such a header is passed (for example, x-virtual-mta: ), the message will be accepted and then queued to the {default} VirtualMTA.

Default: false

Example:

```
<source 0/0>
allow-empty-x-virtual-mta true
</source>
```

### 3.3.7.64 accept-invalid-recipients

Allows feeding invalid recipients into PowerMTA (which are then immediately bounced). Invalid in this sense means syntactically illegal email addresses (i.e., me@there or you@yahoo.com), but not e.g., bad mailboxes (syntactically correct but non-existent email addresses). Pickup directory is not supported. For this functionality in a pickup file use XACK OFF/ON. See [BSMTF Files](#) for more information.

\*\* Note that any such recipients (with syntax errors in their addresses) are immediately bounced and never delivered. This directive only prevents the rejections from happening within the session that the recipients were submitted in. \*\*

Default: false

Example:

```
<source 0/0>
accept-invalid-recipients true
</source>
```

### 3.3.7.65 reject-spf-check-temperror

Specifies whether to reject the MAIL command (with a 4xx error) in the SMTP protocol if the SPF check results in a temporary error, such as when unable to look up certain DNS records.

Default: true

Example:

```
<source 0/0>
reject-spf-check-temperror true
</source>
```

### 3.3.7.66 reject-spf-check-permerror

Specifies whether to reject the MAIL command (with a 5xx error) in the SMTP protocol if the SPF check results in a permanent error, such as when there is a syntax error in a SPF record.

Default: true

**Example:**

```
<source 0/0>
  reject-spf-check-permerror true
</source>
```

**3.3.7.67 reject-spf-check-fail**

Specifies whether to reject the MAIL command (with a 5xx error) in the SMTP protocol if the SPF check results in a failure, such as when email is being sent from an unauthorized IP address.

**Default:** true

**Example:**

```
<source 0/0>
  reject-spf-check-fail true
</source>
```

**3.3.7.68 always-allow-api-submission**

This directive allows easily enabling all SMTP permissions and features required by our submission APIs. If set to "yes", this is currently equivalent to setting "allow-mailmerge yes". If set to "no", the other settings determine whether submission is allowed.

**Default:** false

**Example:**

```
<source 0/0>
  always-allow-api-submission true
</source>
```

**3.3.7.69 auto-qualify-domain**

Specifies whether PowerMTA should automatically qualify the domain name in any email addresses received whose domain is not yet fully qualified. For example, if MAIL FROM:[user@host](mailto:user@host) is received and the host running PowerMTA is named mail.foo.com, then that address is changed into [user@host.foo.com](mailto:user@host.foo.com). Automatic qualification is done by appending the *domain suffix*, which is also configurable.

**Default:** true

**Example:**

```
<source 0/0>
  auto-qualify-domain true
</source>
```

**3.3.7.70 include**

This directive specifies an additional configuration file to process. This can be used, for example, to facilitate maintenance of the configuration files across multiple hosts, by storing those settings which differ from host to host in a separate file and including it from the main (common) file. Wildcards may be used. An include may be used anywhere including in a <source>, <domain>, <virtual-mta> or <smtp-pattern-list> tag.

**Example:**

```
<source 0/0>
  include /etc/pmta/smtp-patterns.txt
</source>
```

**3.3.8 <smtp-user> Directives**

<smtp-user> directives can be used to configure credentials for users who are allowed to relay email through the system, regardless of the IP from which they are submitting the mail to PowerMTA. Authenticating users can be done either through the use of passwords, or through system logins (Linux only) or using certificates.

**Example:**

```
<smtp-user foo>
  password barbaz
  authentication-method password
  source source-1
</smtp-user>

<smtp-user bar>
  authentication-method certificate
  tls-certificate /etc/pmta/bar-cert.pem
  source source-2
</smtp-user>
```

**3.3.8.1 password**

Specifies the password for the defined user in <smtp-user>. The authentication-method must be set to password for this directive to be effective.

**Example:**

```
<smtp-user foo>
  password barbaz
  authentication-method password
</smtp-user>
```

**3.3.8.2 authentication-method**

Specifies how to authenticate a user logging in to PowerMTA via SMTP. Possible values are:

- **password** - The user needs to enter the password as configured by the password directive.
- **certificate** - The user needs to send the TLS certificate configured by the tls-certificate directive. This requires a TLS connection, either by using STARTTLS in the SMTP dialog or by opening a TLS encrypted connection to PowerMTA. Make sure the submitting applications are configured to support this.
- **system** (Linux only) - PowerMTA will check the user's submitted password against the Linux system's user accounts.

**Syntax:**

```
authentication-method { password | system | certificate }
```

**Default:** password

**Example:**

```
<smtp-user foo>
  password barbaz
  authentication-method password
</smtp-user>
```

### 3.3.8.3 tls-certificate

Specifies the file name containing the certificate the user needs to provide when authenticating to PowerMTA in the SMTP dialog. You can use PowerMTA's pmtakeytool that ships with PowerMTA to create a certificate. Another option is to use OpenSSL's command line tools. (Please see the OpenSSL documentation for details.)

**Example:**

```
<smtp-user foo>
  authentication-method certificate
  tls-certificate /etc/pmta/bar-cert.pem
</smtp-user>
```

### 3.3.8.4 source

Specifies the <source> to be used for the defined user in <smtp-user>.

**Example:**

```
<smtp-user foo>
  authentication-method certificate
  tls-certificate /etc/pmta/bar-cert.pem
  source source-1
</smtp-user>
```

## 3.3.9 <source-group> Directives

### 3.3.9.1 max-smtp-in

Specifies a limit for the number of incoming connections from that source group.

**Default:** unlimited

**Example:**

```
<source-group one>
  max-smtp-in 1000
</source-group>
```

### 3.3.9.2 reserved-smtp-in

Specifies that a certain number of inbound connections are to be reserved to connections from the given source group.

**Default:** unlimited

**Example:**

```
<source-group one>
  reserved-smtp-in 10
</source-group>
```

## 3.3.10 <domain> Directives

These directives are specified with a domain pattern and apply to all matching domains, such as in

```
<domain *.port25.com>
  bounce-after 6d
</domain>
```

Domain directives may be added in any order to the configuration file. When PowerMTA needs to determine what parameters to use for a specific domain, it uses the *best* matching (most specific) entry, regardless of the order domain directives are placed into the configuration file:

```
<domain discard.port25.com> # matches 'discard' only
...
</domain>

<domain *.port25.com> # matches subdomains of port25.com
...
</domain>

<domain [*.]port25.com> # matches port25.com and all subdomains
...
</domain>
```

```
<domain *> # matches all domains
...
</domain>
```

When PowerMTA reads the configuration file, it merges settings for related domains from more general to more specific domains, so you can provide defaults for all domains by specifying them for the \* pseudo-domain or for a specific site by specifying them for the toplevel domain for the site.

In the following example, commands will be logged for all proper subdomains of port25.com (but not for port25.com itself) and undelivered messages to any address at port25.com will bounce after four days, except for messages to discard.port25.com, which will bounce after one hour.

```
<domain discard.port25.com>
  bounce-after 1h
</domain>

<domain *.port25.com>
  log-commands yes
</domain>

<domain [*.]port25.com>
  bounce-after 4d
</domain>

<domain *>
  bounce-after 4d
</domain>
```

You can use domain macros to specify settings for multiple domains at once. For example, if you could use

```
domain-macro topLevel com, de, co.uk

<domain yahoo.$topLevel>
  ...settings...
</domain>
```

to configure settings that apply to yahoo.com, yahoo.de and yahoo.co.uk.

You can also use macros within macros:

```
domain-macro com com, co.uk, com.br
domain-macro tld $com, org, net
```

In which case tld expands to com, co.uk, com.br, org and net. Any loops are detected and result in a configuration load or startup error.

Also, specifically listed domains always take precedence over expanded ones. For example, in

```
<domain aol.com>
  max-smtp-out 100
</domain>

domain-macro com com, co.uk

<domain aol.$com>
  max-smtp-out 50
</domain>
```

aol.com would have a limit of 100.

### 3.3.10.1 max-errors-per-connection

This directive tells PowerMTA to break a connection after a certain amount of recipient level errors. The default is unlimited, meaning that PowerMTA will not break a connection due to a number of recipient level errors.

**Default:** unlimited

**Example:**

```
<domain hotmail.com>
  max-errors-per-connection 10
</domain>
```

### 3.3.10.2 check-spf-outbound

Specifies whether PowerMTA should perform an SPF check on outbound messages to ensure compliance. If enabled and the check returns a "Pass", message delivery continues; if instead it returns a "TempError", the message is retried. Otherwise, the message bounces. To use this feature, you must currently configure a specific smtp-source-host (either at the global or VirtualMTA level).

**Default:** false

**Example:**

```
<domain hotmail.com>
  check-spf-outbound true
</domain>
```

### 3.3.10.3 dsn-format

The default is "standard", which has PowerMTA send a DSN in the standard format. If set to "plain-text", the "message/delivery-status" portion of the DSN report is delivered instead of the full report. The portion is delivered within a mime-type of "text/plain", thus allowing this data to pass through (overzealous) email firewalls that strip all "attachments"

**Syntax:**

```
dsn-format { standard | plain-text }
```

**Default:** standard

**Example:**

```
<domain hotmail.com>
  dsn-format plain-text
</domain>
```

**3.3.10.4 log-resolution**

Specifies that PowerMTA should log information on DNS (name and routing) lookups performed for the domain. This information is generally useful when debugging DNS-based connectivity problems (such as misconfigured DNS entries).

**Default:** false

**Example:**

```
<domain hotmail.com>
  log-resolution true
</domain>
```

**3.3.10.5 log-transfer-failures**

Specifies whether PowerMTA should write a warning to the logging file when connections fail while transferring message's contents (headers and body). Setting this directive to true may help when diagnosing delivery problems, especially duplicate or partial deliveries.

**Default:** false

**Example:**

```
<domain hotmail.com>
  log-transfer-failures true
</domain>
```

**3.3.10.6 log-resolution**

Specifies that PowerMTA should log information on DNS (name and routing) lookups performed for the domain. This information is generally useful when debugging DNS-based connectivity problems (such as misconfigured DNS entries).

**Default:** false

**Example:**

```
<domain hotmail.com>
  log-resolution true
</domain>
```

**3.3.10.7 log-disabled-ips**

Turns on logging to show when and which IPs were disabled.

**Default:** false

**Example:**

```
<domain hotmail.com>
  log-disabled-ips true
</domain>
```

**3.3.10.8 max-events-recorded**

Specifies the number of *events* PowerMTA is to record about the given domain. These events are not written to the logging file, but just recorded in memory. They can be viewed using the `pmta show domains` and the `pmta show topdomains` commands, or via the web interface. While recording such events can be very useful when investigating delivery problems, may have an impact on memory consumption. Currently only error events are recorded.

**Default:** 10

**Example:**

```
<domain port25.com>
  max-events-recorded 3
</domain>
```

At the command prompt:

```
$ pmta show dom port25.com --errors
-----domain --#rcpt ---kbytes conn last error-----
port25.com 4 0.5 0 ETIMEDOUT connect...

2003-06-19 16:40:10 ETIMEDOUT connecting to mail.port25.com (193.96.192.241)
2003-06-19 16:40:10 ETIMEDOUT connecting to mail.port25.com (193.96.192.241)
2003-06-19 16:40:10 ETIMEDOUT connecting to mail.port25.com (193.96.192.241)

1 of 10 domains shown.
```

**3.3.10.9 disable-acct-records**

Allows one to disable the creation of one or more than one type of accounting file record on a per queue (domain / virtualMTA) basis. For example, there are times when a user wants PowerMTA to discard certain messages without creating an accounting file record for these messages when discarded. By default, PowerMTA would create a "d" (delivered ) record for this, since it was delivered to discard, however the creation of a delivered record can be disabled with this option.

**Syntax:**

```
disable-acct-records TYPES
```

where TYPES can be one or more of d,b,t,tq

**Example:**

```
<domain hotmail.com>
  disable-acct-records d,b,t,tq
</domain>
```

---

### 3.3.10.10 deliver-local-dsn

Specifies whether PowerMTA should generate DSN (bounce, delay or delivery) reports for messages received **from** the domain specified in this <domain>. Accounting records are still written normally, so this only affects the actual report emails.

**Deprecated** Use <source> suppress-local-dsn instead.

**Default:** true

**Example:**

```
<domain bounce.port25.com>
  deliver-local-dsn no
</domain>
```

---

### 3.3.10.11 bounce-after

Determines the maximum amount of time PowerMTA will continue to try to deliver messages to the specified destination domain. Messages older than this time interval are bounced, i.e., a (non-)delivery report in DSN format will be sent back to the originator.

**Syntax:**

```
bounce-after TIME-INTERVAL
```

**Default:** 4d12h

**Example:**

```
<domain hotmail.com>
  bounce-after 1d
</domain>
```

---

### 3.3.10.12 bcc-upon-delivery

Automatically BCC delivered emails to the given address.

Use empty string ("") to clear any inherited value.

**Syntax:**

```
bcc-upon-delivery EMAIL-ADDRESS
```

**Example:**

```
<domain hotmail.com>
  bcc-upon-delivery foo@bar.com
</domain>
```

---

### 3.3.10.13 allow-priority-interruption

When set to "no", this directive prevents interruptions by higher priority queues.

**Default:** yes

**Example:**

```
<domain hotmail.com>
  allow-priority-interruption true
</domain>
```

---

### 3.3.10.14 allow-priority-interruption-during-transfer

When set to 'no', the interruption is delayed until a connection is available. When set to 'yes', the interruption to an existing connection happens immediately. This directive applies to SMTP queues only.

**Default:** no

**Example:**

```
<domain hotmail.com>
  allow-priority-interruption-during-transfer true
</domain>
```

---

### 3.3.10.15 queue-priority

When PowerMTA needs to deliver email for a queue but no connection slots are available, it looks for a queue with lower priority currently connected and interrupts as many connections as needed, starting with the lowest priority connections and working its way up. A queue's priority is set with a number between 0 (lowest) and 100 (highest); 50 by default.

**Default:** 50

**Example:**

```
<domain hotmail.com>
  queue-priority 75
</domain>
```

---

### 3.3.10.16 track-recipient-events

Determines whether PowerMTA will keep track of the last 50 recipient delivery, bounce, and deferred events in memory.

**Default:** false

**Example:**

```
<domain hotmail.com>
  track-recipient-events true
</domain>
```

---

### 3.3.10.17 delivery-priority

**Deprecated** use queue-priority

---

### 3.3.10.18 notify-of-delay-every

Sets how often PowerMTA checks to see if delay notification messages should be sent when using NOTIFY=DELAY option with RCPT TO command.

**Syntax:**

```
notify-of-delay-every { TIME-INTERVAL | never }
```

**Default:** 4h

**Example:**

```
<domain hotmail.com>
  notify-of-delay-every 1h
</domain>
```

---

### 3.3.10.19 dkim-sign

Specifies whether or not PowerMTA should perform DKIM signing on messages destined for this domain. Domain keys must be configured for this directive to take effect. Please see the `domain-key` directive below for more information. Also accepts a special value: `if-x-dkim-options-present` as a way to enable DKIM signing only when an X-DKIM-Options header is present in the email. The directive continues to accept 'yes' and 'no' settings with their existing meaning.

**Default:** false

**Example:**

```
<domain hotmail.com>
  dkim-sign true
</domain>
```

---

### 3.3.10.20 dkim-add-body-limit

Specify whether to add an "l=" tag with the body length to the DKIM signature, limiting the signed portion of the body. This allows further content to be added at the bottom of the message without invalidating the signature.

**Default:** false

**Example:**

```
<domain hotmail.com>
  dkim-add-body-limit true
</domain>
```

---

### 3.3.10.21 dkim-add-timestamp

Specify whether to add a timestamp to the DKIM signature.

**Default:** false

**Example:**

```
<domain hotmail.com>
  dkim-add-timestamp true
</domain>
```

---

### 3.3.10.22 dkim-algorithm

Specify which DKIM signing algorithm to use.

**Syntax:**

```
dkim-algorithm { rsa-sha1 | rsa-sha256 }
```

**Default:** rsa-sha256

**Example:**

```
<domain hotmail.com>
  dkim-algorithm rsa-sha1
</domain>
```

---

### 3.3.10.23 dkim-body-canon

Specify which DKIM canonicalization to apply to the message body.

**Syntax:**

```
dkim-body-canon { simple | relaxed }
```

**Default:** relaxed

**Example:**

```
<domain hotmail.com>
  dkim-body-canon simple
</domain>
```

---

### 3.3.10.24 dkim-headers-canon

Specify which DKIM canonicalization to apply to the message headers.

**Syntax:**

```
dkim-headers-canon { simple | relaxed }
```

**Default:** relaxed

**Example:**

```
<domain hotmail.com>
  dkim-headers-canon simple
</domain>
```

---

### 3.3.10.25 dkim-expire-after

Requests that an "x=" tag be included in the DKIM signature to indicate when it expires.

**Syntax:**

```
dkim-expire-after { TIME-INTERVAL | never }
```

**Default:** never

**Example:**

```
<domain hotmail.com>
  dkim-expire-after 1d
</domain>

<domain yahoo.com>
  dkim-expire-after never
</domain>
```

---

### 3.3.10.26 dkim-query-method

Specifies the value of the q= tag in DKIM signatures.

**Syntax:**

```
dkim-query-method STRING
```

**Example:**

```
<domain hotmail.com>
  dkim-query-method dns/txt
</domain>
```

---

### 3.3.10.27 dkim-headers

Specifies the headers to be used for DKIM signing. The following headers will always be used if included in the message and cannot be removed:

- CC
- List-Id
- Resent-Date
- Content-Description
- List-Owner
- Resent-From
- Content-Id
- List-Post
- Resent-Message-ID
- Content-Transfer-Encoding
- List-Subscribe
- Resent-Sender
- Content-Type
- List-Unsubscribe
- Resent-To
- Date
- MIME-Version
- Sender
- From
- Message-ID
- Subject
- In-Reply-To
- References
- To
- List-Archive
- Reply-To
- List-Help

- Resent-Cc

RFC 4871 forbids signing the following headers:

- Bcc
- Keywords
- Return-Path
- Comments
- Received
- DKIM-Signature
- Resent-Bcc

**Example:**

```
<domain yahoo.com>
  dkim-headers foo                # exact match one header
</domain>

<domain hotmail.com>
  dkim-headers foo, bar, baz      # exact match many headers
</domain>

<domain gmail.com>
  dkim-headers /foo/             # regular expression
</domain>

<domain aol.com>
  dkim-headers /foo/, /bar/, /baz/ # many regular expressions
</domain>

<domain comcast.net>
  dkim-headers foo, /bar/        # exact match and a regular expression
</domain>

<domain port25.com>
  dkim-headers ""               # no changes to default headers
</domain>
```

### 3.3.10.28 dkim-identity

Specifies the domain or email address of the signing identity to be used when signing messages with DKIM. (Supported in the DKIM specification only). If defined, and the domain matches, or is a subdomain of the domain defined in the "domain-key" directive, the domain will be used in place of the Sender or From: header domain for signing and for subsequent key verification by the receiving gateways. This may be useful for those who are sending on behalf of another party, and do not have control of the other party's DNS record. A service provider for example, can sign mail using their own private key, use their own email address/domain as the signing identity, and have their own public key in their own DNS record for validation, all while having the From: header domain in the messages sent being the domain of their customer. The "dkim-identity" address will be seen in the i= parameter in the signature, while the d= parameter is set via the domain used in the "domain-key" directive and which has to match or be the parent domain in the dkim-identity directive. The directive also takes the keyword `strict-from` which requires an exact match for the From: header's domain to find a matching domain key for signing. Any sub-domain or the sender header will not be used.

In the above example, the "From:" or "Sender:" headers can be a different domain than "esp123.com", but it will be the domain "esp123.com" that will be used for the signing, and, the public key in the TXT record in DNS for the domain `key1._domainkey.esp123.com` will be the one checked for verification.

**Syntax:**

```
dkim-identity { EMAIL-ADDRESS | DOMAIN | sender-or-from }
```

**Default:** sender-or-from

**Example:**

```
domain-key key1,esp123.com,c:\pmta\m.pem

<domain yahoo.com>
  dkim-sign yes
  dkim-identity @esp123.com
</domain>

<domain hotmail.com>
  dkim-sign yes
  dkim-identity sender-or-from
</domain>

<domain gmail.com>
  dkim-sign yes
  dkim-identity email@esp123.com
</domain>
```

### 3.3.10.29 dkim-identity-fallback

Specifies an alternative DKIM identity to use if the primary identity isn't usable (usually because no keys are matched). This allows PowerMTA to use the dkim key for the From or Sender header if available ( eg. domain-key directive), but if the key is not available, PowerMTA switches to using the dkim-identity-fallback for signing (i= specified email address in the signature). Requires the use of dkim-identity.

**Example:**

```
domain-key key1,esp456.com,c:\pmta\m.pem
<domain yahoo.com>
  dkim-sign yes
  dkim-identity sender-or-from
  dkim-identity-fallback @esp456.com
</domain>
```

### 3.3.10.30 dkim-disallow-adding-headers

Primarily used to minimize DKIM replay attacks, it allows one to specify headers that invalidate the DKIM signature if the headers are

added after the DKIM signing. The directive takes a comma separated list of headers to include an extra time in the signature's "h" tag. Any headers listed in this directive that are also present in the original message will be signed as well.

**Syntax:**

```
dkim-disallow-adding-headers HEADER-LIST
```

**Example:**

```
<domain yahoo.com>
    dkim-disallow-adding-headers from,to,subject
</domain>
```

### 3.3.10.31 queue-to

Allows for grouping domains with the same MX record together. See [here](#) for more information.

Use the empty string ("") to clear any inherited value.

### 3.3.10.32 source-ip-max-msg-rate

IP rate limiting allows for controlling the number of attempted recipients on a per-hour, per-minute and per-second basis for each IP address for each domain/VirtualMTA. Primarily used by sites that define multiple IPs in a single VirtualMTA, and that want to limit the attempted delivery rate for each IP address in the VirtualMTA to the respective domains.

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Syntax:**

```
source-ip-max-msg-rate CIDR { unlimited | 0 | N/[M]{h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<domain comcast.net>
    source-ip-max-msg-rate 1.2.3.4 600/h
</domain>
```

### 3.3.10.33 max-msg-rate

Specifies *maximum* attempted recipients to this domain per time interval.

Both deliveries and attempted deliveries are counted towards the limit. The following error will be displayed in the web monitor when the rate limit is reached:

```
message rate limit reached (based on max-msg-rate in configuration)
```

When using rate limiting, it is best to establish a baseline without rate limiting to ensure that the limits set are having a positive impact on deliverability.

**New in PowerMTA v5** When specifying the rate limit, you can specify a multiple in the denominator as well. This makes it easier to configure rate limits for time intervals that may not be exact seconds, minutes, hours, or day boundaries. For example, setting the rate limit to 100/15m states that the rate limit allows 100 events every 15 minutes. Setting the rate limit to 83/5s allows 83 events every 5 seconds.

**Syntax:**

```
max-msg-rate { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<domain comcast.net>
    max-msg-rate 83/10m
</domain>

<domain gmail.com>
    max-msg-rate 500/h
</domain>
```

### 3.3.10.34 backoff-max-msg-rate

This directive has the same function as max-msg-rate, and applies instead of max-msg-rate while a queue is in backoff mode. Both deliveries and attempted deliveries are counted towards the limit. The following error will be displayed in the web monitor when the rate limit is reached: "message rate limit reached (based on backoff-max-msg-rate in configuration)" When using rate limiting, it is best to establish a baseline without rate limiting to ensure that the limits set are having a positive impact on deliverability.

**Syntax:**

```
backoff-max-msg-rate { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** If max-msg-rate is also configured for the same queue, the default is equal to the configured max-msg-rate. Else, it is unlimited.

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Example:**

```

<domain hotmail.com>
  # backoff-max-msg-rate is 600/h
  backoff-max-msg-rate 600/h
</domain>

<domain port25.com>
  # backoff-max-msg-rate is unlimited
</domain>

<domain foo.com>
  max-msg-rate 100/h

  # backoff-max-msg-rate defaults to 100/h
</domain>

```

### 3.3.10.35 max-msg-per-hour

**Deprecated** Use max-msg-rate

### 3.3.10.36 backoff-max-msg-per-hour

**Deprecated** Use backoff-max-msg-rate

### 3.3.10.37 max-data-volume-rate

Specifies *maximum* amount of data that will be transferred to this domain in the given time interval.

Only the volume of data for successful deliveries are counted towards the limit. The following error will be displayed in the web monitor when the rate limit is reached:

```
message rate limit reached (based on max-data-volume-rate in configuration)
```

Data size suffixes can be used to specify the volume of data to limit in the given time interval. For instance, setting the volume rate to 10g/2h specifies that at most 10 Gigabytes of data will be transferred in a 2 hour time interval.

**New in PowerMTA v5** When specifying the rate limit, you can specify a multiple in the denominator as well. This makes it easier to configure rate limits for time intervals that may not be exact seconds, minutes, hours, or day boundaries. For example, setting the rate limit to 100/15m states that the rate limit allows 100 events every 15 minutes. Setting the rate limit to 83/5s allows 83 events every 5 seconds.

**Syntax:**

```
max-data-volume-rate { unlimited | 0 | N [{b|B|k|K|m|M|g|G|t|T}] /[X]{d|day|h|hr|m|min|s|sec} }
```

For the numerator, "b" or "B" means bytes, "k" or "K" means kilobytes, "m" or "M" means megabyte, "g" or "G" means gigabyte, "t" or "T" means terabyte. A number N must be specified in the numerator. If a data size suffix is not specified, it is assumed to mean bytes.

For the denominator, "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. X is optional, but must be a number if specified. If X is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```

<domain hotmail.com>
  # limit data volume to 1048576 bytes (or, 1 megabyte) every 1 hour
  max-data-volume-rate 1048576/1h
</domain>

<domain aol.com>
  # limit data volume to 1024 kilobytes (or, 1 megabyte) every 15 seconds
  max-data-volume-rate 1024K/15s
</domain>

<domain comcast.net>
  # limit data volume to 12 megabytes every 1 minute
  max-data-volume-rate 12M/m
</domain>

<domain yahoo.com>
  # limit data volume to 2 gigabytes every 10 hours
  max-data-volume-rate 2g/10h
</domain>

<domain gmail.com>
  # limit data volume to 1 terabyte every 7 days
  max-data-volume-rate 1T/7d
</domain>

```

### 3.3.10.38 backoff-max-data-volume-rate

This directive has the same function as max-data-volume-rate, and applies instead of max-data-volume-rate while a queue is in backoff mode.

The following error will be displayed in the web monitor when the rate limit is reached: "message rate limit reached (based on backoff-max-data-volume-rate in configuration)"

**Syntax:**

```
backoff-max-data-volume-rate { unlimited | 0 | N [{b|B|k|K|m|M|g|G|t|T}] /[X]{d|day|h|hr|m|min|s|sec} }
```

For the numerator, "b" or "B" means bytes, "k" or "K" means kilobytes, "m" or "M" means megabyte, "g" or "G" means gigabyte, "t" or "T" means terabyte. A number N must be specified in the numerator. If a data size suffix is not specified, it is assumed to mean bytes.

For the denominator, "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means

per-second. X is optional, but must be a number if specified. If X is not specified, it defaults to 1.

**Default:** If `max-data-volume-rate` is also configured for the same queue, the default is equal to the configured `max-data-volume-rate`. Else, it is unlimited.

See [max-data-volume-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Example:**

```
<domain hotmail.com>
  # backoff-max-msg-rate is 600/h
  backoff-max-msg-rate 600/h
</domain>

<domain port25.com>
  # backoff-max-msg-rate is unlimited
</domain>

<domain foo.com>
  max-msg-rate 100/h

  # backoff-max-msg-rate defaults to 100/h
</domain>
```

---

### 3.3.10.39 retry-after

Specifies the retry interval or intervals for the domain/virtualMTA once the domain/virtualMTA is put in retry mode. The directive is a time interval(s), takes a single interval or comma separated list of intervals, with each having a number along with "d" for days, "h" for hours, "m" for minutes, or "s" for seconds, with no spaces between the parameters.

**Syntax:**

```
retry-after TIME-INTERVAL-LIST
```

**Default:** 10m

**Example:**

```
<domain example.com>
  retry-after 10m,10m,10m,10m,10m,1h,1h,1h,4h,4h,12h
</domain>
```

In the above example, PowerMTA would try every 10 minutes for the first 5 tries, every hour for the next three tries, every 4 hours for the next two tries, then every 12 hours until the message is bounced. If during any of the retry intervals an email is delivered then the retry-after period is reset and starts again at the first retry interval. The feature is meant to protect against using resources to continually try and email to a dead domain, like example.com.

---

### 3.3.10.40 backoff-retry-after

This directive has the same function as `retry-after`, and applies instead of `retry-after` while a queue is in backoff mode.

**Syntax:**

```
backoff-retry-after TIME-INTERVAL-LIST
```

**Default:** 1h

**Example:**

```
<domain hotmail.com>
  backoff-retry-after 2h
</domain>
```

---

### 3.3.10.41 retry-recipients-after

Specifies the minimum time interval to wait before retrying delivery for a recipient that previously received a 4XX (temp error) from the receiving gateway. Even if the domain/VirtualMTA is retried as per its `retry-after` setting, individual recipients do not get retried until the `retry-recipients-after` has elapsed.

**Syntax:**

```
retry-recipients-after TIME-INTERVAL
```

**Default:** 10m

**Example:**

```
<domain hotmail.com>
  retry-recipients-after 10m
</domain>
```

---

### 3.3.10.42 retry-after-dns-error

Specifies the retry interval for the domain/virtualMTA once the domain/virtualMTA is put in retry mode due to a DNS resolution error.

**Syntax:**

```
retry-after-dns-error TIME-INTERVAL
```

**Default:** 1m

**Example:**

```
<domain example.com>
  retry-after-dns-error 30s
</domain>
```

---

In the above example, PowerMTA would try every 30 seconds until the message is bounced. Typically this is set to be lesser than `retry-after` so that PowerMTA retries sooner in case of any temporary errors during DNS resolution (such as a timeout connecting to the DNS server, or a DNS misconfiguration).

This directive also applies while the queue is in backoff mode.

---

#### 3.3.10.43 backoff-notify

Specifies email addresses to notify when any of the queues for the given domain enters or leaves backoff mode. Multiple addresses may be specified in a comma separated list.

Use empty string ("" ) to clear any inherited value.

##### Syntax:

```
backoff-notify EMAIL-ADDRESS-LIST
```

##### Example:

```
<domain *>
  backoff-notify two@host.domain,one@host.domain
</domain>
```

---

#### 3.3.10.44 disabled-source-ip-notify

A comma separated list of email addresses that will be notified when a source IP is disabled, either via a command or through an smtp-pattern-list match, and when a source IP is enabled back again.

Use empty string ("" ) to clear any inherited value.

##### Syntax:

```
disabled-source-ip-notify EMAIL-ADDRESS-LIST
```

##### Example:

```
<domain *>
  disabled-source-ip-notify two@host.domain,one@host.domain
</domain>
```

---

#### 3.3.10.45 backoff-to-normal-after

Specifies a time interval after which a queue automatically goes back to normal mode. Example:

##### Syntax:

```
backoff-to-normal-after { TIME-INTERVAL | never }
```

**Default:** never

##### Example:

```
<domain hotmail.com>
  backoff-to-normal-after 6h
</domain>
```

---

#### 3.3.10.46 backoff-to-normal-after-delivery

If enabled, this puts a queue back into normal mode after a successful delivery.

**Default:** false

##### Example:

```
<domain hotmail.com>
  backoff-to-normal-after-delivery true
</domain>
```

---

#### 3.3.10.47 backoff-upon-all-sources-disabled

If enabled, switches a queue into backoff mode when all source IPs become disabled, whether via command line command or when using the `<smtp-pattern-list>` function “disable-source-ip”. Note: SparkPost recommends using the “backoff-to-normal-after” directive in conjunction with this directive, since PowerMTA does not automatically put a queue back into normal mode when an IP address reverts back to being enabled.

**Default:** false

##### Example:

```
<domain hotmail.com>
  backoff-upon-all-sources-disabled true
</domain>
```

---

#### 3.3.10.48 reenable-source-ip-after

Sets the interval of time for when an IP disabled with the `<smtp-pattern-list>` function “disable-source-ip” is to be re-enabled, when no “reenable-after=” option is defined in the pattern action. This directive is ignored however when the “disable-source-ip” actions are defined with the accompanying “reenable-after=” option.

##### Syntax:

```
reenable-source-ip-after { TIME-INTERVAL | never }
```

**Default:** never

**Example:**

```
<domain hotmail.com>
  reenable-source-ip-after 1h
</domain>
```

### 3.3.10.49 max-cold-virtual-mta-msg

A per-domain "max-cold-virtual-mta-msg N/day", default of 0, specifies how many messages go to the "foo-cold" vmta when VirtualMTA "foo" is selected. Re-selection occurs at message reception and over SMTP only. The value specified is the first X messages for that domain. The defined amount is set for delivery upon injection.

**Note: When MX rollup rules are also active in the configuration, PowerMTA will use the max-cold-virtual-mta-msg setting from the rollup queue instead of the individual queues that got rolled up.**

Note: PowerMTA may route less traffic through cold vmtas when an <mx-rollup-list> tag exists in the configuration, even for domains that aren't rolled up.

Please note that this setting is per domain, so 1000 emails will go to yahoo, 1000 will go to hotmail, etc. Auto increments are supported in the following manner:

```
<domain yahoo.com>
  max-cold-virtual-mta-msg 1/day,2/day,3/day,4/day
</domain>
```

In the example above, all cold vmtas will have 1 as their first day's limit for yahoo.com, 2 as the limit for second day to yahoo.com and so on. There is no restriction on the number of per-day limits that can be configured. The last limit value will continue to be used as the per-day limit after all the preceding limits have been used.

**Syntax:**

```
max-cold-virtual-mta-msg DAILY-LIMITS
```

where DAILY-LIMITS is a list of one or more rates of the form N/{d|day}

**Default:** 0/d

**Example:**

```
<virtual-mta vmta1>
  smtp-source-host 1.2.3.4 vmta1.example.com
  cold-virtual-mta vmta2
  <domain *>
    max-cold-virtual-mta-msg 1000/day
  </domain>
</virtual-mta>

<virtual-mta vmta2>
  smtp-source-host 5.6.7.8 vmta2.example.com
  <domain *>
    max-msg-rate 100/h
  </domain>
</virtual-mta>
```

### 3.3.10.50 reroute-to-virtual-mta

Specifies that PowerMTA should reroute messages to the given VirtualMTA. This can be set globally.

A reroute to a VirtualMTA pool is not supported.

Use empty string ("") to clear any inherited value.

**Syntax:**

```
reroute-to-virtual-mta VMTA-NAME
```

**Example:**

```
<virtual-mta vmta1>
  ...
</virtual-mta>

<virtual-mta vmta2>
  ...
</virtual-mta>

<domain aol.com>
  reroute-to-virtual-mta vmta2
</domain>
```

Or, it may also make sense to use this directive on a per VirtualMTA basis:

```
<virtual-mta vmta1>
  ...
  <domain aol.com>
    reroute-to-virtual-mta vmta2
  </domain>
</virtual-mta>

<virtual-mta vmta2>
  ...
</virtual-mta>
```

### 3.3.10.51 backoff-reroute-to-virtual-mta

Specifies that PowerMTA should reroute messages to the given VirtualMTA in case the queue enters backoff mode. For example, if

messages are queued for example.port25.com/vmta1 and that queue enters backoff mode while backoff-reroute-to-virtual-mta is set to vmta2, all of its messages would be rerouted to the queue example.port25.com/vmta2. New messages entering the queue are also rerouted, as long as the queue remains in backoff mode.

Use empty string ("") to clear any inherited value.

It may also make sense to use this directive on a per VirtualMTA basis:

```
<virtual-mta vmta1>
...
<domain *>
    backoff-reroute-to-virtual-mta vmta2
</domain>
</virtual-mta>

<virtual-mta vmta2>
...
</virtual-mta>
```

**WARNING:** PowerMTA performs full loop detection on this directive. If a loop is detected, messages are left in the last queue in the loop, and an error is written to the log file. A reroute from one VirtualMTA to itself is silently ignored, and a reroute to a VirtualMTA pool is not supported.

**Syntax:**

```
backoff-reroute-to-virtual-mta VMTA-NAME
```

**Example:**

```
<virtual-mta vmta1>
</virtual-mta>

<domain *>
    backoff-reroute-to-virtual-mta vmta1
</domain>
```

### 3.3.10.52 type

Determines whether email for this domain is to be delivered over SMTP, through the pipe delivery API, to a file, or discarded. The default type for a queue is `smtp`, which means email from the queue is delivered over SMTP. The other choices are

1. `file`, in which case, the queue writes its messages to files.
2. `pipe`, in which case the queue delivers its messages to a pipe program.
3. `discard` in which case, the queue discards the email writing the corresponding accounting record
4. `http-delivery`, in which case, the queue will deliver its messages to a HTTP(S) endpoint as a JSON object.

**Note:** Change from one type of queue to another with messages in the given queue requires a restart of PowerMTA.

**Syntax:**

```
type { smtp | pipe | file | discard | http-delivery }
```

**Default:** `smtp`

**Example:**

```
<domain hotmail.com>
    type file
</domain>
```

### 3.3.10.53 allow-cancel-during-transfer

Specifies whether PowerMTA should allow connections to this domain to be canceled (e.g., in order to shut down PowerMTA) while a message's contents (headers and body) are being transferred. Some mailers have been known to still deliver messages if the connection breaks before they are fully received, lastly causing partial and duplicate messages to be delivered as PowerMTA retries delivering. Setting this directive to `false` prevents PowerMTA from Canceling connections while messages are being transferred on it, but also possibly causes PowerMTA to take longer to shut down.

**Default:** `true`

**Example:**

```
<domain hotmail.com>
    allow-cancel-during-transfer true
</domain>
```

### 3.3.10.54 auth-username

Specifies the user name to use when authenticating to the remote mailer. Authentication is attempted if the AUTH extension as well as a SASL authentication mechanism supported by PowerMTA are available at the remote mailer, and either `auth-username` or `auth-password` have non-empty values.

Use empty string ("") to clear any inherited value.

PowerMTA currently only supports the CRAM-MD5, LOGIN, and PLAIN authentication mechanisms.

**Example:**

```
<domain hotmail.com>
    auth-username foo
    auth-password 43rusdfdsi
</domain>
```

### 3.3.10.55 smtp-data-termination-timeout

Allows for setting the amount of time that PowerMTA will wait for the final "250 ok" response after sending the message body. Setting this value too low is likely to cause duplicates.

**Syntax:**

```
smtp-data-termination-timeout TIME-INTERVAL
```

**Default:** 10m**Example:**

```
<domain hotmail.com>
  smtp-data-termination-timeout 5m
</domain>
```

**3.3.10.56 auth-password**

Specifies the password to use when authenticating to the remote mailer. Please see the auth-username directive above for more information. auth-password cannot contain a # as it will be interpreted as the start of comments.

**Example:**

```
<domain hotmail.com>
  auth-password password
</domain>
```

**3.3.10.57 use-unencrypted-plain-auth**

Specifies whether PowerMTA should use PLAIN and/or LOGIN for authentication in case the connection is not encrypted (i.e., if STARTTLS is not being used).

**Default:** false**Example:**

```
<domain hotmail.com>
  use-unencrypted-plain-auth true
</domain>
```

**3.3.10.58 smtp-553-means-invalid-mailbox**

Instructs PowerMTA whether to consider a 553 reply to a RCPT command as an 5.1.1 (invalid mailbox) error when enhanced status code is not returned by the remote gateway.

**Default:** yes**Example:**

```
<domain hotmail.com>
  smtp-553-means-invalid-mailbox true
</domain>
```

**3.3.10.59 smtp-421-means-mx-unavailable**

If set to "yes", PowerMTA will immediately break the connection upon receiving a 421 error and subsequently attempt a new connection to the next MX record listed in DNS. The default setting of "no" will cause PowerMTA to reset the connection and then reuse the same connection for additional recipients.

**Default:** no**Example:**

```
<domain hotmail.com>
  smtp-421-means-mx-unavailable true
</domain>
```

**3.3.10.60 replace-smtp-421-service-response**

When enabled, this replaces the SMTP response after the 421 code and after any enhanced status code with "service unavailable", thus preventing any email addresses, etc. that may be included in that response from being used in bounces. The original message is still passed in for pattern matching, so that these continue to work normally.

**Default:** false**Example:**

```
<domain hotmail.com>
  replace-smtp-421-service-response true
</domain>
```

**3.3.10.61 bounce-upon-no-mx**

If set to "true" and no MX record(s) are found for the domain, all emails in the queue bounce immediately.

If a time interval is specified, such as 30m, only messages that have been queued for that amount of time are bounced if no MX record(s) are found for the domain. Defining a time interval allows PowerMTA to attempt delivery to the domain by looking up its A/AAAA records.

Note that this directive applies only to cases where a MX lookup is performed; if a host list is specified with the smtp-hosts directive, no MX lookup is performed.

**Syntax:**

```
bounce-upon-no-mx ( true | false | TIME-INTERVAL | never )
```

**Default:** false

**Example:**

```
<domain example.com>
  bounce-upon-no-mx true
</domain>

<domain *>
  bounce-upon-no-mx 30m
</domain>
```

**3.3.10.62 bounce-upon-5xx-greeting**

Specifies whether PowerMTA should immediately bounce all messages for the domain when receiving a 5xx SMTP greeting. email protocols mandate this default behavior, however there exist many misconfigured mailers on the network, causing email to be lost unnecessarily. If this directive is set to false, PowerMTA will regard 5xx greetings as a temporary error and try delivering the messages to any secondary MX mailers, as well continue retrying delivery until the bounce-after interval expires.

**Default:** true

**Example:**

```
<domain hotmail.com>
  bounce-upon-5xx-greeting true
</domain>
```

**3.3.10.63 bounce-upon-pix-transfer-failure**

This directive is identical to the bounce-upon-transfer-failure directive below, except that the message is only bounced if a Cisco PIX Firewall is detected at the receiving site. Detection is performed based on the SMTP greeting line: if it contains at least ten asterisks, a PIX firewall is detected.

**Default:** false

**Example:**

```
<domain hotmail.com>
  bounce-upon-pix-transfer-failure true
</domain>
```

**3.3.10.64 bounce-upon-transfer-failure**

Specifies whether PowerMTA should immediately bounce a message whose delivery failed while its contents (message headers and body) were being transferred. Normally, PowerMTA would not immediately bounce the message, but continue trying until it becomes older than the bounce-after interval. Due to deficiencies in the SMTP protocol (See RFC 1047) and in the receiving mail servers, if the TCP connection fails while the contents are being transmitted, there is some risk that various copies of the same message will be delivered, some of which possibly with only partial contents. By enabling this directive, you instruct PowerMTA to bounce the message immediately after the first failed transfer attempt. This effectively prevents sending out duplicates, but may also cause the recipient to never receive the message, or receive only one (possibly partial) copy of it.

**Default:** false

**Example:**

```
<domain hotmail.com>
  bounce-upon-transfer-failure true
</domain>
```

**3.3.10.65 connect-timeout**

Specifies the maximum amount of time that PowerMTA waits for an outgoing SMTP connection to be established.

**Syntax:**

```
connect-timeout TIME-INTERVAL
```

**Default:** 2m

**Example:**

```
<domain hotmail.com>
  connect-timeout 5m
</domain>
```

**3.3.10.66 smtp-greeting-timeout**

Specifies the amount of time to wait for the initial SMTP 220 greeting message to be received after the connection is accepted.

**Syntax:**

```
smtp-greeting-timeout TIME-INTERVAL
```

**Default:** 5m

**Example:**

```
<domain hotmail.com>
  smtp-greeting-timeout 10m
</domain>
```

**3.3.10.67 data-send-timeout**

Specifies the maximum amount of time that PowerMTA waits for a chunk of data (message contents and body) to be sent over an outgoing SMTP connection. For full compliance with RFC 2821, this timeout needs to be at least three minutes.

**Syntax:**

```
data-send-timeout TIME-INTERVAL
```

**Default:** 3m**Example:**

```
<domain hotmail.com>
  data-send-timeout 5m
</domain>
```

**3.3.10.68 ignore-8bitmime**

Specifies whether PowerMTA should ignore the 8BITMIME extension even when supported by the receiving mailer. Setting this directive to true allows you to work around the problem of messages bouncing because a site's external MTA accepting 8BITMIME messages as such but unable to convert them to 7BIT when passing to an internal MTA which does not support 8BITMIME. The directive doesn't change the encoding of the message, but rather, by default if a message has 8bit encoding and the feeding application passed that information along in the MAIL FROM command, PowerMTA passes that info along to the remote mail server via the MAIL FROM command. Setting the value to true tells PowerMTA to not pass along the information. This has nothing to do with anything declared in the body of the message. See RFC 1652 for more information <http://www.ietf.org/rfc/rfc1652.txt>.

**Default:** false**Example:**

```
<domain hotmail.com>
  ignore-8bitmime true
</domain>
```

**3.3.10.69 ignore-chunking**

Specifies whether PowerMTA should ignore the CHUNKING extension even when supported by the receiving mailer. Setting this directive to true may be useful for debugging message transfer problems.

**Default:** false**Example:**

```
<domain hotmail.com>
  ignore-chunking true
</domain>
```

**3.3.10.70 ignore-pipelining**

Specifies whether PowerMTA should ignore the PIPELINING extension even when supported by the receiving mailer. Setting this directive to true may be useful for debugging message transfer problems.

**Default:** false**Example:**

```
<domain hotmail.com>
  ignore-pipelining true
</domain>
```

**3.3.10.71 ignore-dsn**

Specifies whether PowerMTA should ignore the DSN extension even when supported by the receiving mailer (seen in the "250" responses after connecting).

**Default:** false**Example:**

```
<domain hotmail.com>
  ignore-dsn true
</domain>
```

**3.3.10.72 max-msg-per-connection**

Specifies the maximum number of messages delivered within a single connection. Normally it is most efficient to deliver as many messages as possible per connection, but in special circumstances it may be desirable to have PowerMTA close the connection and reconnect before delivering more messages. A value of zero means no limit is imposed.

**Default:** 0**Example:**

```
<domain hotmail.com>
  max-msg-per-connection 5
</domain>
```

**3.3.10.73 max-rcpt-per-transaction**

Specifies the maximum number of recipients passed with each message. For example, if a message is submitted including 5,000 recipients for a single domain, and max-rcpt-per-message for that domain is set to 3,000, PowerMTA will transfer the message twice, once for 3,000 recipients and again for the remaining 2,000 recipients. Smaller settings increase parallelism by allowing the same message to be transferred over several connections to the same domain, but also reduce efficiency, since the message contents are transferred several times. max-rcpt-per-transaction is an alias to max-rcpt-per-message.

**Default:** 1000**Example:**

```
<domain hotmail.com>
  max-rcpt-per-transaction 100
</domain>
```

### 3.3.10.74 max-rcpt-per-message

Specifies the maximum number of recipients accepted in each message. If set to zero or `unlimited`, no explicit limit is enforced.

#### Syntax:

```
max-rcpt-per-message { N | unlimited }
```

**Default:** 0

#### Example:

```
<domain yahoo.com>
  max-rcpt-per-message 10
</domain>
```

### 3.3.10.75 max-smtp-out

Specifies the *maximum* number of simultaneous connections to be opened for this queue. This limit is independent of connections to the same domain but from a different VMTA.

**Default:** 20

#### Example:

```
<domain hotmail.com>
  max-smtp-out 30
</domain>
```

### 3.3.10.76 backoff-max-smtp-out

Specifies the *maximum* number of simultaneous connections to be opened for this domain when in backoff mode.

**Default:** 20

#### Example:

```
<domain hotmail.com>
  backoff-max-smtp-out 30
</domain>
```

### 3.3.10.77 max-smtp-out-per-source-ip

**Deprecated** Use `source-ip-max-smtp-out` instead

### 3.3.10.78 source-ip-max-smtp-out

Specifies the *maximum* number of simultaneous connections to be opened for this domain for a given source IP.

#### Syntax:

```
source-ip-max-smtp-out CIDR N
```

**Default:** unlimited

#### Example:

```
<domain comcast.net>
  source-ip-max-smtp-out 1.2.3.4 10
</domain>
```

### 3.3.10.79 smtp-out-rampup-interval

Specifies the minimum amount of time PowerMTA waits before opening another connection to deliver email from the queue. This can reduce the number of connections needed and help avoid hitting a connection limit.

#### Syntax:

```
smtp-out-rampup-interval TIME-INTERVAL
```

**Default:** 0

#### Example:

```
<domain hotmail.com>
  smtp-out-rampup-interval 1s
</domain>
```

### 3.3.10.80 max-connect-rate

Specifies the *maximum* number of connections to be opened for this domain during the specified time period.

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

#### Syntax:

```
max-connect-rate { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<domain hotmail.com>
  max-connect-rate 10/m
</domain>
```

### 3.3.10.81 source-ip-max-connect-rate

Specifies the *maximum* number of connections to be opened during the specified time period when using the given source IP for this domain/vmta (i.e. queue).

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Syntax:**

```
source-ip-max-connect-rate CIDR { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<domain hotmail.com>
  source-ip-max-connect-rate 1.2.3.4 60/m
</domain>
```

### 3.3.10.82 connection-idle-timeout

Specifies the maximum amount of time that PowerMTA waits for another email to be submitted for delivery once there is no remaining email to deliver in the queue. This can reduce the number of connections needed and help avoid hitting a connection limit for opening too many connections within a short interval. The maximum value is 10m. See also [smtp-out-rampup-interval](#).

**Syntax:**

```
connection-idle-timeout TIME-INTERVAL
```

**Default:** 0

**Example:**

```
<domain hotmail.com>
  connection-idle-timeout 1m
</domain>
```

### 3.3.10.83 source-ip-max-msg-rate

Specifies the *maximum* number of messages to be attempted for delivery during the specified time period when using the given source IP for this domain/vmta (i.e. queue).

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Syntax:**

```
source-ip-max-msg-rate CIDR { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<domain comcast.net>
  source-ip-max-msg-rate 1.2.3.4 600/m
</domain>
```

### 3.3.10.84 source-ip-max-data-volume-rate

Specifies *maximum* amount of data that will be transferred to this domain in the given time interval when using the given source IP for this domain/vmta (i.e. queue).

See [max-data-volume-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Syntax:**

```
source-ip-max-data-volume-rate CIDR { unlimited | 0 | N [{b|B|k|K|m|M|g|G|t|T}] / [X]
{d|day|h|hr|m|min|s|sec} }
```

For the numerator, "b" or "B" means bytes, "k" or "K" means kilobytes, "m" or "M" means megabyte, "g" or "G" means gigabyte, "t" or "T" means terabyte. A number N must be specified in the numerator. If a data size suffix is not specified, it is assumed to mean bytes.

For the denominator, "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. X is optional, but must be a number if specified. If X is not specified, it defaults to 1.

See [max-data-volume-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Default:** unlimited

**Example:**

```
<domain comcast.net>
  source-ip-max-data-volume-rate 1.2.3.4 600/m
</domain>
```

### 3.3.10.85 backoff-max-connect-rate

Specifies the *maximum* number of connections to be opened for this domain during the specified time period when in backoff mode.

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

#### Syntax:

```
backoff-max-connect-rate { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** If max-connect-rate is also configured for the same queue, the default is equal to the configured max-connect-rate. Else, it is unlimited.

#### Example:

```
<domain hotmail.com>
  # backoff-max-connect-rate is 600/h
  backoff-max-connect-rate 600/h
</domain>

<domain port25.com>
  # backoff-max-connect-rate is unlimited
</domain>

<domain foo.com>
  max-connect-rate 100/h

  # backoff-max-connect-rate defaults to 100/h
</domain>
```

### 3.3.10.86 retry-upon-new-mail

Specifies if PowerMTA should immediately schedule delivery of a message when it is received if the queue to which it is injected is already in retry mode. If set to true and a queue is in retry mode, new mail injected into that queue will cause the queue to be scheduled for delivery.

**Default:** false

#### Example:

```
<domain hotmail.com>
  retry-upon-new-mail true
</domain>
```

### 3.3.10.87 pix-bug-workaround

Some versions of the Cisco PIX firewall have a bug (bug ID CSCds90792) in its "fixup smtp" command in that it rejects email (causing duplicates to be delivered) if the final . sequence is split across various TCP frames. This directive instructs PowerMTA, in case it detects a PIX firewall, to send the final . in a frame of its own, making it almost certain to arrive at the receiving site in a single frame. Detection is performed based on the SMTP greeting line: if it contains at least ten asterisks, a PIX firewall is detected.

**Default:** true

#### Example:

```
<domain hotmail.com>
  pix-bug-workaround true
</domain>
```

### 3.3.10.88 route

Overrides or modifies the normal DNS-based routing, instructing PowerMTA to deliver to specific hosts or to look up an MX record for a different domain.

**Deprecated** Use smtp-hosts instead.

### 3.3.10.89 default-smtp-port

Allows specifying the TCP port to use for delivering over SMTP (unless overridden by smtp-hosts). This can be used to simplify the setup when multiple PowerMTAs communicate through a NAT (network address translation) device, since the NAT configuration can key on the destination port rather than require individual local source IPs for each externally visible IP address.

**Default:** 25

#### Example:

```
<virtual-mta vmta1>
  <domain *>
    default-smtp-port 9001
  </domain>
</virtual-mta>
<virtual-mta vmta2>
  <domain *>
    default-smtp-port 9002
  </domain>
</virtual-mta>
```

### 3.3.10.90 smtp-hosts

Overrides or modifies the normal DNS-based routing, instructing PowerMTA to deliver to specific hosts or to look up an MX record for a different domain.

Multiple host names or IP addresses can be entered. Commas separate hosts within a preference level, and semicolons separate preference levels. Email delivery is attempted by preference level (first to last), exactly as though those hosts had been obtained from MX records.

An IP address may be entered in dotted-decimal notation instead of a host name, for example [10.1.0.2] for IPv4 or [IPv6:2001::1] for IPv6. Each host name or IP address may be followed by : and a number specifying an alternative TCP port number, as in h1.test.com:1025.

To specify that a MX lookup be performed instead, specify lookup-mx: followed by the domain name to look up.

Use the empty string ("") to clear any inherited value.

#### Examples:

```
<domain hotmail.com>
  smtp-hosts lookup-mx:example.com
</domain>
```

In the following example, commas and semicolons are used to create a primary preference level containing hosts h1 and h2 (which will be used in round-robin fashion), a secondary preference level containing the IP address [10.1.1.1] and specifying an alternate TCP port, and a tertiary preference level (only used if unable to deliver to the first two levels) containing host last:

```
<domain email.test.com>
  smtp-hosts h1.test.com, h2.test.com; [10.1.1.1]:2525; last.test.com
</domain>
```

### 3.3.10.91 smtp-pattern-list

Specifies that the given SMTP pattern list is to be used in connections to the given domain. See [<smtp-pattern-list> Directives](#) for more information.

#### Syntax:

```
smtp-pattern-list NAME
```

#### Example:

```
<smtp-pattern-list foo>
</smtp-pattern-list>

<domain hotmail.com>
  smtp-pattern-list foo
</domain>
```

### 3.3.10.92 assume-delivery-upon-data-termination-timeout

If enabled and the connection times out while waiting for the reply to the "." after a DATA or for the reply to the last BDAT segment, PowerMTA assumes that the delivery was successful. May result in duplicate messages if used when not needed.

**Default:** false

#### Example:

```
<domain hotmail.com>
  assume-delivery-upon-data-termination-timeout true
</domain>
```

### 3.3.10.93 disable-mx-rollback

Controls whether MX rollback should be performed for the domain. When set to true, DNS results for the domain are not checked against any [MX rollback patterns](#) configured in the system.

**Default:** false

#### Example:

```
<domain hotmail.com>
  disable-mx-rollback true
</domain>
```

### 3.3.10.94 include

This directive can be used at domain level to specify an additional configuration file to process. This can be used, for example, to facilitate maintenance of the configuration files across multiple hosts, by storing those settings which differ from host to host in a separate file and including it from the main (common) file. Wildcards may be used. An include may be used at global level and within <source>, <domain>, <virtual-mta> or <smtp-pattern-list> tag.

#### Example:

```
<domain yahoo.com>
  include /etc/pmta/dkim-keys.txt
</domain>
```

### 3.3.10.95 SSL/TLS Directives

#### 3.3.10.95.1 use-starttls

Specifies whether PowerMTA should use the STARTTLS extension (if supported by the receiving mailer). Setting this directive to true instructs PowerMTA to use that extension to encrypt the connection to the remote mailer.

**Default:** true

**Example:**

```
<domain hotmail.com>
  use-starttls true
</domain>
```

---

#### 3.3.10.95.2 require-starttls

Only relevant if "use-starttls" is enabled. The STARTTLS extension may not be available at the remote host, or its use may fail for various reasons. If "require-starttls" is set to "true", PowerMTA will abort the connection, thus avoiding to send the message through a less secure connection. Otherwise, it will send the message anyway.

**Default:** false

**Example:**

```
<domain hotmail.com>
  require-starttls true
</domain>
```

---

#### 3.3.10.95.3 use-dane

Specifies whether PowerMTA should use DANE DNS records to verify whether the remote site requires the use of STARTTLS extension when transferring email to it. Setting this directive to true instructs PowerMTA to look up DANE DNS records for the remote site and attempt to verify the certificate presented by the remote site during STARTTLS based on the methods specified in the DANE DNS records. If the remote site requires the use of STARTTLS in its DANE DNS records, and the STARTTLS attempt to the remote site fails for any reason, PowerMTA will not transfer email to the remote site over plain text (in other words, downgrades to plain text on STARTTLS failures is not allowed). Note that PowerMTA has to be configured with a DNS resolver that supports DNSSEC. See [DANE](#) for more information about the use of DANE in PowerMTA.

**Default:** true

**Example:**

```
# PowerMTA has to be configured with a DNS resolver that supports DNSSEC in order to use DANE.
name-server 8.8.8.8

<domain hotmail.com>
  use-dane true
</domain>
```

---

#### 3.3.10.95.4 reuse-ssl-session

Session reuse is a mechanism to improve performance when establishing SSL/TLS sessions with a remote host. If PowerMTA is re-connecting to the same host within a short duration, it may be possible to reuse the previous SSL/TLS session details speeding up the initial handshake via an abbreviated protocol.

Session reuse is not supported for TLSv1.3 connections. TLSv1.3 session establishment is faster than prior SSL/TLS versions. This directive will be ignored for TLSv1.3 sessions.

**This directive depends on the remote host's ability to support reusing SSL/TLS sessions. At present, only gmail.com is known to support. We suggest enabling the directive for only those domains that are known to support this feature.**

**Default:** false

**Example:**

```
<domain gmail.com>
  reuse-ssl-session true
</domain>
```

---

#### 3.3.10.95.5 smtp-tls-allow-tlsv1

Allows PowerMTA to use TLSv1 when establishing a secure connection.

**Default:** false

**Example:**

```
<domain hotmail.com>
  smtp-tls-allow-tlsv1 false
</domain>
```

---

#### 3.3.10.95.6 smtp-tls-allow-tlsv1.1

Allows PowerMTA to use TLSv1.1 when establishing a secure connection.

**Default:** true

**Example:**

```
<domain hotmail.com>
  smtp-tls-allow-tlsv1.1 true
</domain>
```

---

#### 3.3.10.95.7 smtp-tls-allow-tlsv1.2

Allows PowerMTA to use TLSv1.2 when establishing a secure connection.

**Default:** true

**Example:**

```
<domain hotmail.com>
```

---

```
    smtp-tls-allow-tlsv1.2 true
</domain>
```

---

### 3.3.10.95.8 smtp-tls-allow-tlsv1.3

Allows PowerMTA to use TLSv1.3 when establishing a secure connection.

**Default:** true

**Example:**

```
<domain hotmail.com>
    smtp-tls-allow-tlsv1.3 true
</domain>
```

---

### 3.3.10.95.9 smtp-tls-allow-sslv3

This directive is no longer valid as PowerMTA no longer supports SSLv3.

---

### 3.3.10.95.10 smtp-tls-ca-file directive

Specifies the location of a local text file containing one or more root certificates that will be used to verify remote server certificate chains. Used by senders to authenticate remote gateways when using TLS based delivery. If using opportunistic TLS and the certificate verification fails, PowerMTA will fall back to using plain text for the delivery. In contrast, if TLS is required and the certificate verification fails, PowerMTA will return a temporary 4xx error indicating the failure.

**Example:**

```
<domain hotmail.com>
    smtp-tls-ca-file /etc/pmta/certchain.pem
</domain>
```

---

### 3.3.10.95.11 smtp-tls-ciphers

Specifies the ciphers that PowerMTA is allowed to use when negotiating a session using a TLSv1.2 or earlier protocol with remote client during STARTTLS. Please see documentation for OpenSSL command 'ciphers' for valid cipher names that can be used with this directive.

**Default:** "HIGH:!ADH:@STRENGTH"

**Example:**

```
<domain example.com>
    smtp-tls-ciphers HIGH:!ADH:@STRENGTH
</domain>
```

---

### 3.3.10.95.12 smtp-tls-tlsv1.3-ciphersuites

Specifies the ciphersuites that PowerMTA is allowed to use when negotiating a session using the TLSv1.3 protocol with remote client during STARTTLS. Please see documentation for OpenSSL command 'ciphers' for valid ciphersuite names that can be used with this directive.

**Default:**

TLS\_AES\_256\_GCM\_SHA384:TLS\_CHACHA20\_POLY1305\_SHA256:TLS\_AES\_128\_GCM\_SHA256:TLS\_AES\_128\_CCM\_SHA256:TLS\_AES\_128\_CCM\_8\_SHA256

**Example:**

```
<domain example.com>
    smtp-tls-tlsv1.3-ciphersuites TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
</domain>
```

---

### 3.3.10.95.13 log-tls

Specifies whether PowerMTA should log information during negotiation of a TLS session with the remote during STARTTLS.

**Default:** false

**Example:**

```
<domain hotmail.com>
    log-tls true
</domain>
```

---

## 3.3.10.96 Discard Delivery Directives

The directives in this Section are ignored if specified in non-discard delivery domains.

### 3.3.10.96.1 discard-as-bounce

Controls whether emails in a discard type queue should be recorded as bounces('b'). When set to true for a domain with type discard, all emails in the queue are recorded as bounces in the accounting file. When set to false, all emails in the queue are recorded as deliveries('d') in the accounting file.

**Default:** false

**Example:**

```
<domain undeliverable-domain.com>
    type discard
    discard-as-bounce true
</domain>
```

---

## 3.3.10.97 File Delivery Directives

### 3.3.10.97.1 file-format

Specifies the format of the file(s) to which PowerMTA delivers the messages:

- **newfile-plain:** A new file is created for each message, in the directory indicated by file-destination. The entire message is written to the file, including all headers, but no envelope information.
- **newfile-pickup:** A new file is created for each message, in the directory indicated by file-destination. In addition to the entire message body and headers, the special headers x-sender and x-receiver are included, containing the originator and recipient addresses from the message envelope.
- **append-mbox:** Messages are appended to the file indicated by file-destination, in the unix "mbox" format.

PowerMTA creates new files with the initial "extension" of .tmp and renames them to .msg once done. If post-processing these files as they are created, you should code to ignore any .tmp files, to avoid processing a file which hasn't yet been completely written.

When appending to a file (for append-mbox), interlocking is performed using flock on Unix-like platforms and with share options passed to CreateFile on Windows. In order to safely access the file while PowerMTA is running, SparkPost recommends using the [pmtagefile](#) application.

#### Syntax:

```
file-format { newfile-plain | newfile-pickup | append-mbox }
```

**Default:** newfile-plain

#### Example:

```
<domain bounce.mydomain.com>
  type file
  file-destination /opt/pickup/
  file-format newfile-pickup
</domain>
```

### 3.3.10.97.2 file-destination

Specifies the directory or file name for file delivery. The directory or file name is subjected to the same macro expansion as in the pipe delivery's command line:

- **\$bodytype:** The message body type (7BIT, 8BITMIME or BINARYMIME).
- **\$ctime:** The current date in Unix ctime format.
- **\$envid:** The DSN envelope ID (from MAIL ... ENVID=...).
- **\$from:** The envelope sender (MAIL FROM) address.
- **\$isodate:** The current date in the format YYYY-MM-DD.
- **\$timet:** The current time in numeric (time\_t) format.
- **\$rfctime:** The current date in RFC 822 format.
- **\$to:** The envelope recipient (RCPT TO) address.
- **\$user:** The recipient user. That is the local part (mailbox) in the recipient address, or, if the local part contains a '+' character, the portion of the mailbox to the left of the '+'. For example, in [joe+xyzzy@company.com](#), the user would be joe.
- **\$domain:** The recipient's domain.

Whether \$from and \$to are lowercased depends on the setting of [macro-variables-preserve-case](#). \$user and \$domain macros always expand in lower case, avoiding issues when those are used to create directories and files on case-sensitive file systems.

#### Syntax:

```
file-destination DIRECTORY
```

#### Example:

```
<domain bounce.mydomain.com>
  type file
  file-destination /opt/pickup/
</domain>
```

### 3.3.10.98 Pipe Delivery Directives

The directives in this Section are ignored if specified in non-pipe domains.

#### 3.3.10.98.1 command

Specifies the command to use to start the program which will read from the pipe. This directive is required for pipe domains. PowerMTA substitutes the following macros in the command line:

- **\$bodytype:** The message body type (7BIT, 8BITMIME or BINARYMIME).
- **\$ctime:** The current date in Unix ctime format.
- **\$envid:** The DSN envelope ID (from MAIL ... ENVID=...).
- **\$from:** The envelope sender (MAIL FROM) address.
- **\$isodate:** The current date in the format YYYY-MM-DD.
- **\$timet:** The current time in numeric (time\_t) format.
- **\$rfctime:** The current date in RFC 822 format.
- **\$to:** The envelope recipient (RCPT TO) address.
- **\$user:** The recipient user. That is the local part (mailbox) in the recipient address, or, if the local part contains a '+' character, the portion of the mailbox to the left of the '+'. For example, in [joe+xyzzy@company.com](#), the user would be joe.

- **\$domain**: The recipient's domain.

To prevent any one of these macros from being interpreted as multiple arguments if they contain spaces, it is a good idea to enclose them in double quotes (like in the example below).

**Syntax:**

```
command STRING
```

where STRING is the complete command that PowerMTA will use to start the program.

**Example:**

```
<domain bounces.xyz.com>
  type pipe
  command "/my/bounce/processor --envid \"$envid\" \"$user\""
</domain>
```

### 3.3.10.98.2 macro-variables-preserve-case

When set to true prevents \$to and \$from from being changed to lowercase. \$user and \$domain are always changed to lowercase, as they are intended to facilitate integration with Linux environments.

**Default:** false

**Example:**

```
<domain bounces.mydomain.com>
  type pipe
  command "/my/bounce/processor --envid \"$envid\" \"$user\""
  macro-variables-preserve-case true
</domain>
```

### 3.3.10.99 HTTP Delivery Directives

#### 3.3.10.99.1 http-delivery-url

For a queue with its [type](#) set to http-delivery, this directive can be used to configure an HTTP destination endpoint where messages from the queue will be delivered as a JSON object. See [HTTP Delivery Webhooks](#) for more details on the JSON object format.

The URL specified with this directive must be an absolute URL, not relative, must be prefixed by either http or https, and should not contain any userinfo, query-string or fragment components.

**Note:** This directive is required when the queue type is http-delivery.

**Syntax:**

```
http-delivery-url scheme://host[:port][/path]
```

where scheme can be one of https or http, the host must be either a hostname, or an IP address. IPv4 addresses must be in dot-decimal notation, and IPv6 addresses must be enclosed in brackets ([ ]). port is optional - if not specified, it defaults to 443 when scheme is https and 80 when scheme is http. The path component is optional, and if specified must consist of a sequence of path segments separated by a slash (/).

**Example:**

```
<domain yahoo.com>
  type http-delivery
  http-delivery-url http://webhooks.endpoint.com:8181/api/v1/email
</domain>
```

#### 3.3.10.99.2 http-delivery-extra-headers

For a queue with its [type](#) set to http-delivery, this optional directive can be used to configure a file containing HTTP headers that will be included in the headers of the HTTP request that PowerMTA will make when delivering messages from the queue. This could be useful, for example, for including a header such as Authorization as part of the HTTP request. See [HTTP Delivery Webhooks](#) for more details on the JSON object format.

**Syntax:**

```
http-delivery-extra-headers <file-name>
```

where <file-name> is the name of a file containing the extra HTTP headers to include, each header on a line on its own.

**Example:**

```
<domain yahoo.com>
  type http-delivery
  http-delivery-url http://webhooks.endpoint.com:8181/api/v1/email
  http-delivery-extra-headers /etc/pmta/http-delivery-headers.txt
</domain>
```

Sample content of the file: /etc/pmta/http-delivery-headers.txt:

```
Authorization: 14ac5499cfd2bb2859e4476d2e5b1d2bad079bfd
```

**Note** PowerMTA does not check or interpret the content of the headers included in the file. It only does basic syntax checking of the header format.

### 3.3.10.100 SparkPost Traffic Redirection Support Directives

PowerMTA has special support for redirecting traffic for delivery through SparkPost. This may be desirable for various reasons, such as to seasonally add extra delivery capacity. Once your SparkPost setup is ready, you can have PowerMTA connect to it by creating a domain named {sparkpost} and setting it to authenticate using a SparkPost API key:

```
<domain {sparkpost}>
```

```
auth-password d41d8cd98f00b204e9800998ecf8427e # API key
</domain>
```

You can then redirect specific domains (or entire Virtual MTAs, as in the example below) to SparkPost by having PowerMTA place messages in the {sparkpost} queue:

```
<virtual-mta ExtraCapacity>
  <domain *>
    queue-to {sparkpost}
  </domain>
</virtual-mta>
```

#### 3.3.10.100.1 add-x-msys-api-header

Instructs PowerMTA to add a special X-MSYS-API header required to set SparkPost specific options, also setting SparkPost's campaign to the messages' JobID, if any. This directive is enabled by default for the {sparkpost} domain.

**Default:** true for {sparkpost} domain, false for all other domains.

**Example:**

```
<domain {sparkpost}>
  add-x-msys-api-header true
</domain>
```

#### 3.3.10.100.2 sparkpost-sandbox

Sets the sandbox SparkPost option.

**Default:** false

**Example:**

```
<domain {sparkpost}>
  sparkpost-sandbox true
</domain>
```

#### 3.3.10.100.3 sparkpost-ip-pool

Sets the IP pool from which SparkPost is to send messages.

**Syntax:**

```
sparkpost-ip-pool STRING
```

**Example:**

```
<domain {sparkpost}>
  sparkpost-ip-pool shared-pool
</domain>
```

### 3.3.11 <acct-file> Directives

The accounting file in CSV format can be written to a file or pipe. For more information, see [Accounting Files and Statistics](#).

#### 3.3.11.1 delete-after

This directive tells PowerMTA how long to keep the accounting file before deleting it from the server. "delete-after" only applies if "move-to" isn't active" (and move-to must point to a different location than the directory where the file was originally created).

**Syntax:**

```
delete-after { Nd | never }
```

where N is the number of days to retain the accounting file for.

**Default:** 8d

**Example:**

```
<acct-file /var/log/pmta/acct.csv>
  delete-after 5d
</acct-file>
```

#### 3.3.11.2 move-to

This directive tells PowerMTA where to move the accounting file when using move-interval or max-size. If move-to isn't set, no moving is done (move-interval, if configured, is silently ignored). Directory must be on the same volume as the accounting files. Moving across a network is not supported.

The directive is off by default, and the files are not moved unless this directive is configured.

**Syntax:**

```
move-to DIRECTORY
```

**Example:**

```
<acct-file /var/log/pmta/acct.csv>
  move-to /var/log/archive
</acct-file>
```

#### 3.3.11.3 move-interval

This directive tells PowerMTA at what time interval to move the accounting file and start a new one.

**Syntax:**

```
move-interval TIME-INTERVAL
```

**Default:** 1d

**Example:**

```
<acct-file /var/log/pmta/acct.csv>
  move-to /var/log/archive
  move-interval 1h
</acct-file>
```

### 3.3.11.4 max-size

The max size of the accounting file in Bytes, Kilobytes, Megabytes, Gigabytes, or Terabytes before PowerMTA moves the accounting file and starts a new one. Minimum size is 1M (1024K). This directive applies both with and without "move-to". If "move-to" is not specified, a new file is started in place (with a higher -NNNN in the file name).

**Syntax:**

```
max-size N{b|k|m|g|t}
```

**Default:** 250M

**Example:**

```
<acct-file /var/log/pmta/acct.csv>
  max-size 1G
</acct-file>
```

### 3.3.11.5 queue-filter

Can be set to a comma separated list of queue names, each of which can be optionally prefixed with ! to indicate negation. If set, the **queue** field of accounting records is compared against the queue names in the filter list and is written if it matches a non-negated (without a ! prefix) queue name, or not written if it matches a negated (with a ! prefix) queue name. The filtering applies to only records that have a 'queue' field in them (d/b/t/tq). Any f/rb/rs/r records that are configured in the acct-file will be written unconditionally (i.e. all records of that type) since they don't have a 'queue' field.

The matching is performed in the order in which the queue names were specified in the queue-filter directive.

**Note that if queue-filter is active for an accounting file, any records that are filtered out (i.e., any d/b/t/tq with a queue that does not match one of the queue names listed in the directive) will be lost forever (unless they are configured to be written to other accounting files).** Always make sure that the queue-filter is set exactly to your needs.

If all queue names in queue-filter are negated, no records will match. All the queue names listed in the "queue-filter" directive must obey syntax rules for a valid queue name, namely:

- The domain part must be non-empty, must obey domain name syntax, and must be a qualified domain name. It cannot be a reserved domain name (i.e, cannot start with "{" and end with "}"). This rules out domain names like "{feedbackLoop}", "{bounceProcessor}" etc.
- If a VMTA is specified, it must be a syntactically valid VMTA name, or {default}. Note that there is no check for whether such a VMTA is defined in the configuration. It is sufficient if the VMTA name is syntactically valid.
- If no VMTA name is specified, it is assumed to be the name of the default VMTA ({default}), so foo.com will be treated as foo.com/{default}

By default, filtering is not turned on (in other words, accounting records for all queues are written to the accounting file).

**Syntax:**

```
queue-filter QUEUE-NAME-LIST
```

where QUEUE-NAME-LIST is a list of comma separated list of queue names, optionally prefixed with !.

**Example:**

```
<acct-file a.csv>
  records d,b,t,tq,f,rb,rs,r
  queue-filter a.com/vmta1,![*.]b.com/*,*c.com/vmta2,!*/vmta3,*/vmta4
</acct-file>
```

### 3.3.11.6 retry-interval

Retries the pipe program in a regular interval in case of a failure. This should only be used when using pipe to write an accounting file.

**Syntax:**

```
retry-interval TIME-INTERVAL
```

**Default:** 5m

**Example: (Windows)**

```
<acct-file |C:\pmta\bin\customCode.exe --thisOption --thatOption>
  retry-interval 10m
</acct-file>
```

**Example: (Unix)**

```
<acct-file |myscript.pl --thisOption --thatOption>
  retry-interval 10m
</acct-file>
```

### 3.3.11.7 iso-times

Requests that PowerMTA writes time stamps in the accounting file (or pipe) in ISO format.

**Default:** yes

**Example:**

```
<acct-file /var/log/pmta/acct.csv>
    iso-times yes
</acct-file>
```

### 3.3.11.8 user-string

Specifies a custom string to be stored along with other MTA information in the accounting file. The field “userstring” also needs to be added to the record-fields directive for this to show up in the accounting file.

**Syntax:**

```
user-string STRING
```

**Example:**

```
<acct-file /var/log/pmta/acct.csv>
    user-string unique-id
</acct-file>
```

### 3.3.11.9 write-timeout

Allows for setting write timeouts in accounting pipes, so that an unresponsive accounting application doesn't hold up PowerMTA indefinitely. Only used when piping accounting data to another file.

**Syntax:**

```
write-timeout TIME-INTERVAL
```

**Default:** 1m

**Example:**

```
<acct-file |C:\pmta\bin\customCode.exe --thisOption --thatOption>
    write-timeout 2m
</acct-file>
```

### 3.3.11.10 map-header-to-field

Allows for mapping a header to a specific field in the accounting file. This is useful if for example the From header needs to be mapped to the SMTP MAIL FROM address (“orig” field in the accounting file). An example would be like the following:

In the example below, header\_x-foo gets mapped to the rcpt field in bounced records, and header\_x-bar gets mapped to the rcpt field for delivered records. Multiple entries may be specified to map multiple headers to the same field. The tq record type cannot be used. The order of the directives is significant when using multiple ones for the same record type and field. In the above example the rcpt field will contain the content of the x-foo header if x-foo exists, if not it will contain the content of the x-bar header if that one exists. Otherwise the content of the x-baz header. If that one also doesn't exist it will contain the original value of the orig field (i.e. the rcpt field will not be overwritten). The same header can be mapped to multiple fields (e.g., header\_x-foo above) and will be written to all fields mapped.

**Syntax:**

```
map-header-to-field RECORD-TYPE header_NAME ACCT-FIELD-NAME
```

**Default:** none

**Example:**

```
<acct-file /var/log/pmta/acct.csv>
    map-header-to-field b header_x-foo rcpt
    map-header-to-field d header_x-bar rcpt
    map-header-to-field d header_x-baz rcpt
    map-header-to-field d header_x-foo orig
</acct-file>
```

### 3.3.11.11 records

This directive tells PowerMTA which record types to include in the accounting file. There are 8 record types: d, b, t, tq, r, f, rb and rs. It may only be defined once per <acct-file> with multiple types separated by commas.

- “d” is for delivered records and will include all messages that were successfully delivered. “delivery” can be used in place of “d”.
- “b” is for bounced records and will include all locally generated bounces (synchronous bounces). Note that bounces generated by remote mail servers (asynchronous bounces) will not be included. “bounce” can be used in place of “b”.
- “t” is for transient recipient level errors only (such as mailbox full). “transient” can be used in place of “t”.
- “tq” is for queue wide transient errors and will include information about delivery attempts for a given domain/vmta. “transient-queue” can be used in place of “t”.
- “r” is for inbound records and will include information about messages submitted to PowerMTA for delivery. “receipt” can be used in place of “r”.
- “f” is for feedback loop emails if PowerMTA is configured to look for feedback loop emails with the <feedback-loop-processor>. “feedback-loop” may be used in place of “f”. See [Processing Feedback Loop Emails](#) for more information.
- “rb” is for remote bounce emails if PowerMTA is configured to look for remote bounce emails with the <bounce-processor>. “remote-bounce” may be used in place of “rb”. See [Processing Asynchronous \(Remote\) Bounces and remote status reports](#) for more information.

- “rs” is for remote status emails if PowerMTA is configured to look for remote status emails with the <bounce-processor>.
- “remote-status” may be used in place of “rs”. See [Processing Asynchronous \(Remote\) Bounces and remote status reports](#) for more information.

Due to the fact that there can be many delivery attempts for one message, it is suggested that type “t” and/or “tq” should only be used when needed (debugging, troubleshooting, etc.) as it may cause large file sizes. It is also recommended when using this setting to have the “t” or “tq” type records written to their own file, as so not to cause unwanted data in the primary accounting file.

#### Syntax:

```
records RECORD-TYPE-LIST
```

where RECORD-TYPE-LIST is a comma separated list of record types({d|b|t|tq|r|rb|rs|f})

**Default:** d,b

#### Example:

```
<acct-file /var/log/pmta/acct.csv>
  records d, b, t, tq, r, rb, rs, f
</acct-file>
```

### 3.3.11.12 record-fields

This directive specifies the fields to be included in the records defined via the "record" directive, in the accounting file. By default all per record type are included. To include only a subset of all the fields, simply add the desired fields in a comma separated list. To include a custom header, add the record field header \_XXX, where XXX is the name of the custom field you wish to log. For example, to log all fields and a custom header, the directive would look like this:

```
record-fields delivery *,header_Message-Id
```

To include only a subset of the fields, list the required fields similar to the following:

```
record-fields bounce timeLogged,orig,rcpt,dlvSourceIp,vmta
```

To exclude fields from the default, you can use an “!” to exclude the unwanted fields. For example, to include all fields except orcpt you would use:

```
record-fields d *,!orcpt
record-fields b *,!orcpt
```

#### Syntax:

```
record-fields RECORD-TYPE FIELD-LIST
```

where RECORD-TYPE is one of {d|b|t|tq|r|rb|rs|f}

**Default:** \*

#### Example:

```
<acct-file /var/log/pmta/acct.csv>
  record-fields d *,header_Message-Id
  record-fields b *,!orcpt
</acct-file>
```

### 3.3.11.13 world-readable

This directive tells PowerMTA to provide read permissions to all users on the system when it creates a new accounting file.

**Default:** no

#### Example:

```
<acct-file /var/log/pmta/acct.csv>
  world-readable yes
</acct-file>
```

### 3.3.11.14 count-moved-records

This directive tells PowerMTA whether or not to count the records in CSV files moved. The record counts are written to the logging file along with the message that indicates that the file was moved.

**Default:** false

#### Example:

```
<acct-file /var/log/pmta/acct.csv>
  count-moved-records yes
</acct-file>
```

### 3.3.11.15 sync

Determines whether the OS' buffers for the accounting file are flushed regularly. This helps ensure that the accounting data isn't corrupted in case of a system crash.

**Default:** false

#### Example:

```
<acct-file /var/log/pmta/acct.csv>
  sync true
</acct-file>
```

### 3.3.11.16 privacy-policy

This directive, when set to anything but `retain`, instructs PowerMTA to replace the "local part" (left hand side of the "@") in the recipient email addresses, as well as any occurrences found in SMTP responses and in standard VERPED originator addresses, with either the redaction string or a hash of that local part (after normalizing to lower-case). The directive accepts the following:

- `retain`
- `redact` with an optional redaction string (defaults to "...")
- `siphhash` with a key string in plain text
- `siphhash-hexkey` with hex-formatted key string
- `hmac-sha256` with a key string in plain text
- `hmac-sha256-hexkey` with a hex-formatted key string

For Siphhash hash calculations, the key string can only be up to 16 bytes in length. If less than 16 bytes are specified, PowerMTA will append null bytes (`\0`) to the end of the key string to make it 16 bytes in length. The 16 byte length is a requirement of the SipHash algorithm.

For hmac-sha256 hash calculations, RFC 2104 recommends that the key string be at least 32 bytes in length for security reasons. PowerMTA will emit a warning if the key string is less than 32 bytes in length, however, it does not reject such key strings because RFC 2104's suggestion is only a recommendation rather than a requirement.

In the example below, the local part is replaced with X:

```
<acct-file /var/log/pmta/redacted.csv>
  privacy-policy redact "X"
  records b, d
</acct-file>
```

In this, the local part is replaced with the hex-formatted (64-bit) SipHash of the local part, using the key "secret":

```
<acct-file /var/log/pmta/siphhashed.csv>
  privacy-policy siphhash "secret"
  records b, d
</acct-file>
```

See <https://131002.net/siphhash/> for a description of SipHash.

In this, the local part is replaced with the hex-formatted HMAC-SHA256 hash of the local part, using the key "secret":

```
<acct-file /var/log/pmta/hmacsha256-hashed.csv>
  privacy-policy hmac-sha256 "secret"
  records b, d
</acct-file>
```

To manually compute the hash of a local part, the `digest` command of the PowerMTA command line tool can be used. Be sure to supply just the local part of the email address to the command.

**Syntax:**

```
privacy-policy { retain | redact | siphhash | siphhash-hexkey | hmac-sha256 | hmac-sha256-hexkey } [STRING]
```

**Default:** `retain`

**Example:**

```
<acct-file /var/log/pmta/siphhash.csv>
  privacy-policy siphhash "secret"
  records b, d
</acct-file>
```

### 3.3.12 <virtual-mta> Directives

These directives specify behavior specific to a "VirtualMTA". VirtualMTA directives only apply when that VirtualMTA has been selected through one of the supported selection methods.

```
<virtual-mta mta1>
  smtp-source-host 1.2.3.4 mta1.company.com
  <domain *>
    dkim-sign yes
  </domain>
</virtual-mta>
```

See [VirtualMTA Support](#) for more information.

#### 3.3.12.1 use-proxy

Specifies a comma separate list of proxy servers (see [proxy](#) for more details) that the virtual MTA can use when connecting to a remote destination.

The proxy servers in the list are used in the order in which they are defined. The names in the list are strictly used for failover only - the second proxy on the list is used only if the connection to the first proxy fails, the third proxy is used only when connection to both first and second proxy have failed, and so on.

Every proxy server mentioned in this directive must be defined in the configuration with a [proxy](#) tag.

**Syntax:**

```
use-proxy PROXY-SERVER-LIST
```

**Default:** `None`

**Example:**

```
<virtual-mta customer1>
  use-proxy mta-proxy-a,mta-proxy-b # "mta-proxy-a" is always used unless connections to it fail. "mta-proxy-b" is used only when connecting to "mta-proxy-a" fails
</virtual-mta>

<proxy mta-proxy-a>
  server 10.1.2.3:5000
  client 16.70.55.233 client.hostname.com
```

```
</proxy>

<proxy mta-proxy-b>
  server mta-proxy-b.internal.hostname.com:5000
  client 16.80.77.192 client2.hostname.com
</proxy>
```

### 3.3.12.2 max-smtp-msg-rate

Specifies *maximum* attempted recipients per time interval for this VirtualMTA.

Both deliveries and attempted deliveries are counted towards the limit. The following error will be displayed in the web monitor when the rate limit is reached:

```
message rate limit reached (based on max-smtp-msg-rate in configuration)
```

When using rate limiting, it is best to establish a baseline without rate limiting to ensure that the limits set are having a positive impact on deliverability.

See [max-msg-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Syntax:**

```
max-smtp-msg-rate { unlimited | 0 | N/[M]{d|day|h|hr|m|min|s|sec} }
```

where "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. N is required and must be non-zero, but M is optional. If M is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<virtual-mta customer1>
  max-smtp-msg-rate 6/sec
</virtual-mta>
```

### 3.3.12.3 max-smtp-data-volume-rate

Specifies *maximum* amount of data that will be transferred through this virtual MTA in the given time interval.

Only the volume of data for successful deliveries are counted towards the limit.

See [max-data-volume-rate](#) for more examples on rate limit syntax that can be used with this directive.

**Syntax:**

```
max-smtp-data-volume-rate { unlimited | 0 | N [{b|B|k|K|m|M|g|G|t|T}] /[X]{d|day|h|hr|m|min|s|sec} }
```

For the numerator, "b" or "B" means bytes, "k" or "K" means kilobytes, "m" or "M" means megabyte, "g" or "G" means gigabyte, "t" or "T" means terabyte. A number N must be specified in the numerator. If a data size suffix is not specified, it is assumed to mean bytes.

For the denominator, "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. X is optional, but must be a number if specified. If X is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<virtual-mta customer1>
  max-smtp-data-volume-rate 6g/10h
</virtual-mta>
```

### 3.3.12.4 cold-virtual-mta

This directive takes the name of a manually configured VirtualMTA. The directive can be repeated to configure multiple cold VirtualMTAs, in which case these are used in round-robin fashion and the per-domain limit applies individually for each cold VirtualMTA.

**Syntax:**

```
cold-virtual-mta VMTA-NAME
```

**Example:**

```
<virtual-mta vmta-cold>
  smtp-source-host 1.2.3.4 mail2.yourdomain.com
</virtual-mta>

<virtual-mta vmta1>
  smtp-source-host 2.3.4.5 mail3.yourdomain.com
  cold-virtual-mta vmta-cold
  <domain *>
    max-cold-virtual-mta-msg 1000/d
  </domain>
</virtual-mta>
```

### 3.3.12.5 host-name

See [host-name](#) definition in [Global Directives](#) for more information.

**Example:**

```
<virtual-mta vmta>
  host-name mail.foo.com
</virtual-mta>
```

### 3.3.12.6 smtp-source-host

See [smtp-source-host](#) directive in [Global Directives](#) for more information.

**Example:**

```
<virtual-mta mta2.port25.com>
  smtp-source-host 2.3.4.5 mta2.port25.com
</virtual-mta>
```

### 3.3.12.7 domain-key

See [domain-key](#) definition in [General Queuing and Delivery Directives](#) for more information.

**Example:**

```
<virtual-mta mta1>
  domain-key s1,*,/etc/pmta/key.pem
</virtual-mta>
```

### 3.3.12.8 include-headers-from

See [include-headers-from](#) definition in [General Queuing and Delivery Directives](#) for more information.

**Example:**

```
<virtual-mta mta1>
  include-headers-from /etc/pmta/headers.txt
</virtual-mta>
```

### 3.3.12.9 max-smtp-out

Specifies the *maximum* number of simultaneous connections to be opened across all queues from within this virtual MTA.

**Example:**

```
<virtual-mta mta1>
  max-smtp-out 100
</virtual-mta>
```

### 3.3.12.10 rewrite-list

See [<rewrite-list> Directives](#) for more information.

**Syntax:**

```
rewrite-list NAME
```

**Example:**

```
<rewrite-list abc>
  mail-from *@foo.com *@bar.com
</rewrite-list>

<virtual-mta mta1>
  rewrite-list abc
</virtual-mta>
```

### 3.3.12.11 include

This directive can be used at VirtualMTA level to specify an additional configuration file to process. This can be used, for example, to facilitate maintenance of the configuration files across multiple hosts, by storing those settings which differ from host to host in a separate file and including it from the main (common) file. Wildcards may be used. An include may be used at global level and within `<source>`, `<domain>`, `<virtual-mta>` or `<smtp-pattern-list>` tag.

**Example:**

```
<virtual-mta mta1>
  include /etc/pmta/domain-keys.txt
</virtual-mta>
```

## 3.3.13 <virtual-mta-pool> Directives

VirtualMTA pools are groups of VirtualMTAs. They share the same name space as VirtualMTAs, so you can select a pool instead of a specific VirtualMTA. When you select a VirtualMTA pool, the VirtualMTAs in that pool are used in round-robin fashion to deliver the message. See [VirtualMTA Support](#) for more information on VirtualMTAs and VirtualMTA pools.

A `<domain>` may be defined inside of a `<virtual-mta-pool>`. If this is done, the defined directives will be inherited by any VirtualMTA that is part of the pool. If a conflicting directive is defined directly in the given VirtualMTA, the directive in the VirtualMTA overrides the one defined in the `<virtual-mta-pool>`.

The following directives are also supported:

- cold-virtual-mta
- domain-key
- max-smtp-out
- max-smtp-msg-rate
- include-headers-from
- max-smtp-data-volume-rate

### 3.3.13.1 virtual-mta

Specifies the name of a VirtualMTA to include in the pool. To specify multiple VirtualMTAs, repeat this directive with the name of each VirtualMTA to add.

**Syntax:**

```
virtual-mta VMTA-NAME
```

**Example:**

```
<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<virtual-mta-pool pool>
  virtual-mta mta1
  virtual-mta mta2
</virtual-mta-pool>
```

**3.3.13.2 domain**

Specifies the settings for a domain that will be inherited by all the Virtual MTAs defined in the pool.

Within the inheritance hierarchy, a domain defined inside a Virtual MTA will first inherit its settings from domain definitions in the pool that the Virtual MTA is part of and then subsequently inherit from global domain definitions.

**It is an error to define a VMTA in 2 different pools which have <domain> definitions inside them.**

**Example:** In this example, both yahoo.com/mta1 and yahoo.com/mta2 will inherit max-msg-rate 10/h setting from the pool.

```
<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<virtual-mta-pool pool>

  <domain yahoo.com>
    max-msg-rate 10/h
  </domain>

  virtual-mta mta1
  virtual-mta mta2
</virtual-mta-pool>
```

Below is an example of an invalid configuration.

```
# Invalid configuration since mta1 is part of both pool-one and pool-two,
# each of which have domain definitions inside them

<virtual-mta mta1>
</virtual-mta>

<virtual-mta-pool pool-one>

  <domain yahoo.com>
    max-msg-rate 10/h
  </domain>

  virtual-mta mta1
</virtual-mta-pool>

<virtual-mta-pool pool-two>

  <domain yahoo.com>
    max-msg-rate 15/h
  </domain>

  virtual-mta mta1
</virtual-mta-pool>
```

**3.3.13.3 domain-key**

**Syntax:** See [domain-key](#) definition in [General Queueing and Delivery Directives](#) for more information.

**Example:**

```
<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<virtual-mta-pool pool>

  domain-key s1,*,etc/pmta/key.pem

  virtual-mta mta1
  virtual-mta mta2
</virtual-mta-pool>
```

**3.3.13.4 include-headers-from**

**Syntax:** See [include-headers-from](#) definition in [General Queueing and Delivery Directives](#) for more information.

**Example:**

```

<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<virtual-mta-pool pool>

    include-headers-from /etc/pmta/headers.txt

    virtual-mta mta1
    virtual-mta mta2

</virtual-mta-pool>

```

### 3.3.13.5 max-smtp-msg-rate

See [max-smtp-msg-rate](#) definition in [<virtual-mta> Directives](#).

**Example:**

```

<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<virtual-mta-pool pool>

    max-smtp-msg-rate 100/m

    virtual-mta mta1
    virtual-mta mta2

</virtual-mta-pool>

```

### 3.3.13.6 max-smtp-out

**Syntax:** See [max-smtp-out](#) definition in [<domain> Directives](#).

**Example:**

```

<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<virtual-mta-pool pool>

    max-smtp-out 100

    virtual-mta mta1
    virtual-mta mta2

</virtual-mta-pool>

```

### 3.3.13.7 rewrite-list

See [<rewrite-list> Directives](#) for more information.

**Syntax:**

```
rewrite-list NAME
```

**Example:**

```

<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<rewrite-list abc>
    mail-from *@foo.com *@bar.com
</rewrite-list>

<virtual-mta-pool pool>

    rewrite-list abc

    virtual-mta mta1
    virtual-mta mta2

</virtual-mta-pool>

```

### 3.3.13.8 max-smtp-data-volume-rate

See [max-smtp-data-volume-rate](#) for more information about the use of this directive within a Virtual MTA .

**Syntax:**

```
max-smtp-data-volume-rate { unlimited | 0 | N [{b|B|k|K|m|M|g|G|t|T}] /[X]{d|day|h|hr|m|min|s|sec} }
```

For the numerator, "b" or "B" means bytes, "k" or "K" means kilobytes, "m" or "M" means megabyte, "g" or "G" means gigabyte, "t" or "T" means terabyte. A number N must be specified in the numerator. If a data size suffix is not specified, it is assumed to mean bytes.

For the denominator, "d" or "day" means per-day, "h" or "hr" means per-hour, "m" or "min" means per-minute and "s" or "sec" means per-second. X is optional, but must be a number if specified. If X is not specified, it defaults to 1.

**Default:** unlimited

**Example:**

```
<virtual-mta mta1>
</virtual-mta>

<virtual-mta mta2>
</virtual-mta>

<virtual-mta-pool pool>

    max-smtp-data-volume-rate 6g/10h

    virtual-mta mta1
    virtual-mta mta2

</virtual-mta-pool>
```

### 3.3.14 <pattern-list> Directives

Pattern lists allow you to select a VirtualMTA or set recipient priority based on the message's originator, the message's recipient, a message header, or a combination of the three.

The directives described in this section define a list of regular expression patterns. Each entry in a pattern list has the following form:

```
DIRECTIVE PARAMETERS /PATTERN/ RESULT
```

where:

- **DIRECTIVE** is one of the directives below, specifying what is matched.
- **PARAMETERS** is additional information depending on the chosen directive; currently only the *header* directive supports a parameter, which is the message header you want to match.
- **/PATTERN/** is a perl-compatible regular expression (REGEX). For more information on perl-compatible regular expressions, please refer to the [Perl documentation](#) or the [PCRE library package](#).
- **RESULT** is one or more of the following:
  - **virtual-mta=NAME** selects a VirtualMTA.
  - **recipient-priority=N** sets the priority of the recipient using a scale of 0-100 with 0 being the lowest priority and 100 being the highest with a default of 50 when not specified. This may be used without selecting a VirtualMTA.
  - **call-to:PATTERN-LIST-NAME** selects another pattern list.

If multiple entries match, the first match is used.

An *x-virtual-mta* header may be used in conjunction with a <pattern-list> to set a recipient-priority.

#### 3.3.14.1 header

Adds a pattern match against the specified header. Matching is performed against the entire header contents; in particular, any *white space* characters such as blanks, tabs and CR/LFs will still be present. Because of this, when anchoring the regular expression with the caret (^) and dollar (\$) characters, it is important to allow for white space following the caret and preceding the dollar signs, respectively. The \s metacharacter is particularly useful for this, since it will match all white space characters.

**Syntax:**

```
header HEADER-NAME REGEX RESULT
```

**Example:**

```
<pattern-list myList>
  header x-eval /^\\s*eval/ virtual-mta=evaluation
  header from /jsmith/ virtual-mta=evaluation
  header subject /hello world/ virtual-mta=evaluation
</pattern-list>

<source 127.0.0.1>
...
  pattern-list myList # this selects the pattern list for messages
</source> # from 127.0.0.1
```

#### 3.3.14.2 mail-from

Adds a pattern match against the message's originator. This is the SMTP MAIL FROM address and is the address to which bounces are sent. This is not the same as the from header seen in mail clients.

**Syntax:**

```
mail-from REGEX RESULT
```

**Example:**

```
<pattern-list myList>
  mail-from /^customerservice/ virtual-mta=service
</pattern-list>

<source 127.0.0.1>
...
  pattern-list myList # this selects the pattern list for messages
</source> # from 127.0.0.1
```

#### 3.3.14.3 rcpt-to

Adds a pattern match against the message's recipient (SMTP RCPT TO).

#### Syntax:

```
rcpt-to REGEX RESULT
```

#### Example:

```
<pattern-list myList>
  rcpt-to /customer/ virtual-mta=customers
</pattern-list>

<source 127.0.0.1>
  ...
  pattern-list myList # this selects the pattern list for messages
</source> # from 127.0.0.1
```

If multiple recipients are specified in a message and more than one match a rcpt-to directive, the first match's VirtualMTA will apply to the entire message. This behavior may change in the future, so rcpt-to should only be used if you only intend to submit single-recipient messages.

Another example, showing a combination of matching patterns:

```
<pattern-list myList>
  header x-eval /eval/ virtual-mta=evaluation
  header from /jsmith/ virtual-mta=evaluation
  header subject /hello world/ virtual-mta=evaluation
  mail-from /newsletter/ virtual-mta=newsletters
  rcpt-to /^stud.*@somesite.com$/ virtual-mta=students
  rcpt-to /password-reset@customer.com/ recipient-priority=90
  rcpt-to /special@customer.com/ virtual-mta=students,recipient-priority=90
</pattern-list>

<source 127.0.0.1>
  ...
  pattern-list myList # this selects the pattern list for messages
</source> # from 127.0.0.1
```

Here, each email is first checked for the x-eval header. If it is present and its content contains *eval*, the *evaluation* VirtualMTA is used. If the x-eval header isn't given or does not match the pattern, the next pattern is checked. Each pattern is checked from top to bottom, with the first matching pattern being used. If none of the above matched, PowerMTA will use the default VirtualMTA.

This example shows conditional pattern matching:

```
<pattern-list patterns>
  mail-from /newsletter/ <pattern-list>
    rcpt-to /^stud.*@somesite.com$/ virtual-mta=students
    rcpt-to /^staff.*@somesite.com$/ virtual-mta=staff
    * virtual-mta=anything
  </pattern-list>

  rcpt-to /admin/ <pattern-list>
    mail-from /thisdomain.com/ virtual-mta=this
    header x-info /newdata/ virtual-mta=other
    * virtual-mta=anything
  </pattern-list>
</pattern-list>

<source 127.0.0.1>
  ...
  pattern-list patterns # this selects the pattern list for messages from 127.0.0.1
</source>
```

Here, first the MAIL FROM is checked for each email. If the MAIL FROM matches "newsletter", a sub-pattern is then used. In this sub-pattern, the RCPT TO is checked, and if it matches the pattern, the emails will go to the "students" VirtualMTA, or the "staff" VirtualMTA. Anything with "newsletter" in the MAIL FROM that does not match the sub-patterns get sent to the wildcard defined "anything" VirtualMTA.

The second pattern is similar to the above, but first checks the RCPT TO. If the RCPT TO matches, the email is then put through the sub-patterns, first checking the MAIL FROM, and then for the "x-info" header. Once again, emails with a RCPT matching "admin", but not the sub-patterns will be sent to the VirtualMTA "anything".

Emails not matching any patterns will not be sent to a VirtualMTA based on pattern matching.

#### 3.3.14.4 Nested pattern-list support

In some cases it may be better to use a <pattern-list> to select another <pattern-list> instead of selecting a VirtualMTA directly. This can be accomplished in PowerMTA with the new "call-to" option defined within a pattern.

Assuming a message has the following header:

```
x-custom-header: abc123
```

and you wanted to use that header and content to select another pattern to then select the correct VirtualMTA. The following would be defined in PowerMTA's config file:

```
<pattern-list specialSelector1>
  mail-from /newsletters/ virtual-mta=vm1a1
</pattern-list>

<pattern-list selections>
  header x-custom-header /abc123/ call-to: specialSelector1
</pattern-list>

<source 0/0>
  pattern-list selections
</source>
#
```

In the above example, <pattern-list selections> would match the x-custom-header: header, and then select <pattern-list specialSelector1> to then apply next to the message, given the "call-to:" definition. If the SMTP MAIL FROM address for this message included "newsletters", PowerMTA would then use VirtualMTA vm1a1 for the message.

### 3.3.15 <smtp-pattern-list> Directives

SMTP pattern lists allow you to have PowerMTA change a delivery queue's mode based on pattern matching on the SMTP replies received while attempting deliveries.

SMTP pattern lists currently only accept a single directive (reply). Each entry in a pattern list has the following form:

```
reply /PATTERN/ ACTION-LIST
```

Where:

- /PATTERN/ is a perl-compatible regular expression (REGEX). For more information on perl-compatible regular expressions, please refer to the [Perl documentation](#) or the [PCRE library package](#).
- ACTION-LIST is one or more of the following:
  - mode=normal
  - mode=backoff
  - skip-mx
  - bounce-queue
  - bounce-rcpt
  - disable-source-ip
  - reenable-after
  - defer-queue

If multiple conflicting entries match, the first match is used. The following is not allowed:

- both mode=backoff and mode=normal
- both skip-mx and bounce-queue
- reenable-after without a preceding disable-source-ip (in the same rule)
- more than one reenable-after

**skip-mx** is intended to help PowerMTA from give up delivering to one or more MXes which is just giving a transient recipient-level error. Any patterns matched will result in the next MX being used and mail being held in queue, even in the case of a 5xx error.

**bounce-queue** will bounce the entire queue immediately if the pattern is matched. The original error, as well as the fact that the queue was bounced due to this setting, will be recorded in the accounting file.

**bounce-rcpt** will bounce the recipient immediately if the pattern is matched. This only works for 4xx responses to the RCPT command. The original error, as well as the fact that the recipient was bounced due to this setting, will be recorded in the accounting file.

**disable-source-ip** will disable the IP that received the error. The source IP will remain disabled until the time period defined by the <domain> directive reenable-source-ip-after, unless using the reenable-after option.

**reenable-after** Allows for defining the time after which an IP that has been disabled with disable-source-ip will be reenabled. The default is never. Takes priority over the domain's reenable-source-ip-after setting when used.

**defer-queue** When a queue goes in backoff mode, it does not automatically go into retry mode as well. Not all SMTP errors will have PowerMTA putting the queue into retry mode immediately vs., for example, having PowerMTA reconnect on the next MX. Since it may be desirable to have the queue put into retry mode immediately for certain errors, a new defer-queue option was added in SMTP pattern lists.

For example, to put any queues in backoff mode upon reception of certain SMTP replies, while bouncing some messages immediately or skipping some MX records:

```
<smtp-pattern-list blockList>
  reply /generating high volumes of.* complaints from AOL/ mode=backoff
  reply /Excessive unknown recipients - possible Open Relay/ mode=backoff
  reply /^421 .* too many errors/ mode=backoff
  reply /permanently deferred/ bounce-queue
  reply /too many connections/ skip-mx
  reply /account over quota/ bounce-rcpt
  reply /TS03/ mode=backoff,disable-source-ip,reenable-after=1h
  reply /resources temporarily unavailable/ defer-queue
</smtp-pattern-list>

<domain *>
  ...
  smtp-pattern-list blockList
</domain>
```

To match on a # (the comment symbol in the configuration file) hexadecimal code may be used. For example, to match the pattern "Sender address deferred by rule #CR-IN-DEF-2":

```
<smtp-pattern-list blockList>
  reply /Sender address deferred by rule \x23CR-IN-DEF-2 / mode=backoff
</smtp-pattern-list>

<domain *>
  ...
  smtp-pattern-list blockList
</domain>
```

Backslashes (\) and quotes (") will need to be escaped with a backslash. All patterns are case insensitive. If case sensitivity is needed, it can be forced by adding (?-i) like the following example:

```
<smtp-pattern-list blockList>
  reply /(?-i)TS03/ mode=backoff
</smtp-pattern-list>
```

#### 3.3.15.1 include

This directive can be used at smtp-pattern-list level to specify an additional configuration file to process. This can be used, for example, to facilitate maintenance of the configuration files across multiple hosts, by storing those settings which differ from host to host in a separate file and including it from the main (common) file. Wildcards may be used. An include may be used at global level and within <source>, <domain>, <virtual-mta> or <smtp-pattern-list> tag.

**Example:**

```
<domain yahoo.com>
```

```
include /etc/pmta/dkim-keys.txt
</domain>
```

---

### 3.3.16 <dns> Directives

#### 3.3.16.1 precached-domains-file

Specifies the name of a text file containing a list of domains, one domain per line, that you want PowerMTA to always have current DNS information for in its DNS cache, regardless of whether recipients are in the queue for the domain or not. Typically PowerMTA would look up domain routing information in DNS when recipients are submitted into PowerMTA for a domain (assuming recipients are not currently queued for the domain already). Depending on the DNS infrastructure and how many other domains are in the queue, this may cause some slight delay in the immediate delivery attempts for these recipients. Having DNS routing information precached in PowerMTA for certain key domains before submission minimizes any delay that may take place due to this required DNS resolution.

**Syntax:**

```
precached-domains-file FILENAME
```

**Example:**

```
<dns>
  precached-domains-file /etc/pmta/precached-domains.txt
</dns>
```

---

#### 3.3.16.2 precached-refresh-interval

Defines the time frame in which the DNS records for all pre-cached domains are refreshed (looked back up in DNS). Similar to a DNS record's TTL (time to live), however it manually sets the time interval to refresh, and which overrides the actual records' TTL in DNS. The regular TTL records for pre-cached domains are ignored.

**Syntax:**

```
precached-refresh-interval TIME-INTERVAL
```

**Default:** 5m

**Example:**

```
<dns>
  precached-domains-file /etc/pmta/precached-domains.txt
  precached-refresh-interval 2m
</dns>
```

---

#### 3.3.16.3 precached-max-domains

Specifies the maximum number of domains that can be pre-cached. This number cannot exceed the number of domains defined in the "precached-domains-file". Defining a very large number will result in lots of frequent DNS queries, so a scalable DNS infrastructure is required.

**Syntax:**

```
precached-max-domains N
```

where N is the maximum number of domains to precache.

**Default:** 500

**Example:**

```
<dns>
  precached-domains-file /etc/pmta/precached-domains.txt
  precached-max-domains 1000
</dns>
```

---

### 3.3.17 <smtp-listener> Directives

#### 3.3.17.1 await-slot

Specifies whether new connections should wait or be refused if PowerMTA is currently at its total-max-smtp-in limit., applies to connections accepted at this <smtp-listener> address.

**Default:** false

**Example:**

```
<smtp-listener 127.0.0.1:25>
  await-slot true
</smtp-listener>
```

---

#### 3.3.17.2 immediate-tls

Specifies whether to immediately start TLS when a connection is accepted (implementing "SMTPS")., applies to connections accepted at this <smtp-listener>'s address.

**Default:** false

**Example:**

```
<smtp-listener 127.0.0.1:25>
  immediate-tls true
</smtp-listener>
```

---

#### 3.3.17.3 source

Specifies a <source> entry with settings to override those obtained from source IP address matching., applies to connections accepted at this <smtp-listener>'s address.

**Example:**

```
<source named-source>
  max-message-size 100K
</source>

<smtp-listener 127.0.0.1:25>
  source named-source
</smtp-listener>
```

### 3.3.17.4 tls-certificate

Specifies a file containing the X.509 certificate and private key to use if TLS is started on inbound SMTP connections, as well as the password protecting the private key. Applies to connections accepted at this <smtp-listener> address. Inherited from smtp-server-tls-certificate.

**Syntax:**

```
tls-certificate FILENAME [PASSWORD]
```

**Example:**

```
<smtp-listener 127.0.0.1:25>
  tls-certificate /var/lib/pmta/smtp.pem aaaa
</smtp-listener>
```

### 3.3.17.5 tls-allow-tlsv1.3

Allows PowerMTA to use TLSv1.3 when negotiating a secure connection with a remote client during STARTTLS.

**Default:** true

**Example:**

```
<smtp-listener 0/0:25>
  tls-allow-tlsv1.3 true
</smtp-listener>
```

### 3.3.17.6 tls-allow-tlsv1.2

Allows PowerMTA to use TLSv1.2 when negotiating a secure connection with a remote client during STARTTLS.

**Default:** true

**Example:**

```
<smtp-listener 0/0:25>
  tls-allow-tlsv1.2 true
</smtp-listener>
```

### 3.3.17.7 tls-allow-tlsv1.1

Allows PowerMTA to use TLSv1.1 when negotiating a secure connection with a remote client during STARTTLS.

**Default:** true

**Example:**

```
<smtp-listener 0/0:25>
  tls-allow-tlsv1.1 true
</smtp-listener>
```

### 3.3.17.8 tls-allow-tlsv1

Allows PowerMTA to use TLSv1 when negotiating a secure connection with a remote client during STARTTLS.

**Default:** false

**Example:**

```
<smtp-listener 0/0:25>
  tls-allow-tlsv1 true
</smtp-listener>
```

### 3.3.17.9 tls-allow-sslsv3

This directive is no longer valid as PowerMTA no longer supports SSLv3.

### 3.3.17.10 tls-ciphers

Specifies the ciphers that PowerMTA is allowed to use when negotiating a session using a TLSv1.2 or earlier protocol with a remote client during STARTTLS. Please see documentation for OpenSSL command 'ciphers' for valid cipher names that can be used with this.

**Default:** HIGH:!ADH:@STRENGTH

**Example:**

```
<smtp-listener 0/0:25>
  tls-ciphers HIGH:!ADH:@STRENGTH
</smtp-listener>
```

### 3.3.17.11 tls-tls1.3-ciphersuites

Specifies the ciphersuites that PowerMTA is allowed to use when negotiating a session using the TLSv1.3 protocol with a remote client during STARTTLS. Please see documentation for OpenSSL command 'ciphers' for valid ciphersuite names that can be used with this.

**Default:**

TLS\_AES\_256\_GCM\_SHA384:TLS\_CHACHA20\_POLY1305\_SHA256:TLS\_AES\_128\_GCM\_SHA256:TLS\_AES\_128\_CCM\_SHA256:TLS\_AES\_128\_CCM\_8\_SHA256

**Example:**

```
<smtp-listener 0/0:25>
  tls-tls1.3-ciphersuites TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
</smtp-listener>
```

### 3.3.17.12 tls-ca-file

Specifies a "CA file" containing Certification Authority certificates to allow verifying any certificates presented by clients on inbound SMTP connections. The certificates must be in PEM format.

**Example:**

```
<smtp-listener 0/0:25>
  tls-ca-file /etc/pmta/certchain.pem
</smtp-listener>
```

### 3.3.18 <rewrite-list> Directives

Rewrite lists can be used to rewrite the MAIL FROM address (envelope address) of an email and the From header in the body of email. Rewrite lists are created using the tag <rewrite-list>. To use <rewrite-list>, you must configure a source with the name of the rewrite-list to use as shown below.

**Syntax:**

```
<rewrite-list abc>
  mail-from SOURCE-ADDRESS TARGET-ADDRESS
  from-domain SOURCE-DOMAIN TARGET-DOMAIN
</rewrite-list>

<source 0/0>
  rewrite-list abc
</source>
```

#### 3.3.18.1 mail-from

Use mail-from rules to rewrite the MAIL FROM address (envelope address) of an email.

**Syntax:**

```
mail-from SOURCE-ADDRESS TARGET-ADDRESS
```

where SOURCE-ADDRESS can be one of:

| SOURCE-ADDRESS | matches                                |
|----------------|----------------------------------------|
| abcd@xyz.com   | this exact email address               |
| abcd@*         | any address with this local part       |
| *@xyz.com      | any address at this domain             |
| *@*            | all addresses except the empty address |
| <>             | the empty SMTP MAIL FROM address       |

Within a source address, the wildcard \* has the meaning "match-all".

and TARGET-ADDRESS can be one of:

| TARGET-ADDRESS | resulting rewritten address                                       |
|----------------|-------------------------------------------------------------------|
| abcd@xyz.com   | this specific email address                                       |
| abcd@*         | the local part abcd and the same domain as the source address     |
| *@xyz.com      | the same local part as the source address with the domain xyz.com |
| <>             | the empty address                                                 |

Within a target address, the \* has the meaning "replace from source address".

The TARGET-ADDRESS cannot have non-ASCII UTF8 characters in them. For example:

```
mail-from abc@xyz.com foo@bücher.de # this is an invalid rule
```

The wildcard \* cannot be used in a TARGET-ADDRESS if the SOURCE-ADDRESS is empty. For example:

```
mail-from <> *@foo.com # this is an invalid rule
```

**Example:**

```
<rewrite-list abc>
  mail-from foo@* *@bar.com # foo@xyz.com will be rewritten to foo@bar.com
</rewrite-list>
```

#### 3.3.18.2 from-domain

Use from-domain rules to rewrite the domain of the email address contained in the From header of an email.

**Syntax:**

```
from-domain SOURCE-DOMAIN TARGET-DOMAIN
```

where SOURCE-DOMAIN must be a domain pattern (see DOMAIN definition) and TARGET-DOMAIN must be a specific domain name

**Example:**

```
<rewrite-list foo>
  from-domain *.port25.com bouncedomain.com # rewrites foo@mail.port25.com to foo@bouncedomain.com
</rewrite-list>
```

### 3.3.19 <domain-key-list> Directives

domain-key-list can be used to organize a list of domain keys and refer to them using a single name. <domain-key-list> can only be defined in global scope. <domain-key-list> can be used to restrict the search for the exact key to DKIM sign an email to a specific list of keys.

**Syntax:**

```
<domain-key-list NAME>
  domain-key ...
  domain-key ...
</domain-key-list>
```

The x-dkim-options header can be used to specify the domain-key-list that PMTA will use to search for a key from. For example:

```
x-dkim-options: s=foo;key-list=xyz
```

In the above example, PMTA will search for the key with the selector foo in the domain-key-list named xyz. If not found, then PMTA will not sign the message.

If the key-list option is not used in x-dkim-options, then PMTA will fallback to the default behavior of selecting a key from those that were defined in the VMTA scope, or were inherited from global scope.

**Example:**

```
<domain-key-list keylist-customer-one>
  domain-key m,mail.port25.com,c:\pmta\m.pem
  domain-key base,*,c:\pmta\base.pem
</domain-key-list keylist-customer-one>
```

#### 3.3.19.1 domain-key

See [domain-key](#) for description.

### 3.3.20 <transmissions-api> Directives

#### 3.3.20.1 log-requests

Specifies whether HTTP data exchanges during [Transmissions API](#) request handling are logged to PowerMTA's log file or not.

**Default:** false

**Example:**

```
<transmissions-api>
  log-requests yes
</transmissions-api>
```

#### 3.3.20.1.1 api-key

This directive configures authentication keys for applications using the [Transmissions API](#); it can be repeated to create multiple keys. Applications passing a valid key in the Authorization header are allowed to use the HTTP Transmissions API.

This directive may be repeated any number of times to add additional API keys.

**Syntax:**

```
api-key STRING
```

where STRING is a set of alphanumeric characters or the underscore character, and between 32 and 128 characters in length.

**Example:**

```
<transmissions-api>
  api-key 14ac5499cfdd2bb2859e4476d2e5b1d2bad079bf
  api-key 476d2e5b1d2bad079bf14ac5499cfdd2bb2859e4
</transmissions-api>
```

#### 3.3.20.1.2 log-data

Specifies whether HTTP data exchanges during [Transmissions API](#) request handling are logged to PowerMTA's log file or not. (More verbose than log-requests within <transmissions-api>)

**Default:** false

**Example:**

```
<transmissions-api>
  log-data yes
</transmissions-api>
```

#### 3.3.20.1.3 log-tls

Specifies whether PowerMTA should log information about negotiation of a TLS session with the remote client when handling [HTTP Transmissions API](#) requests.

**Default:** false

**Example:**

```
<transmissions-api>
  log-data yes
</transmissions-api>
```

### 3.3.21 <transmissions-api-listener> Directives

#### 3.3.21.1 tls-certificate

Specifies a file containing the X.509 certificate and private key to use if TLS is started on inbound [HTTP Transmissions API](#) connections, as well as the password protecting the private key.

**Syntax:**

```
tls-certificate FILENAME [PASSWORD]
```

**Example:**

```
<transmissions-api-listener 127.0.0.1:443>
  tls-certificate /var/lib/pmta/http.pem aaaa
</transmissions-api-listener>
```

#### 3.3.21.2 tls-allow-tls1.3

Allows PowerMTA to use TLSv1.3 when negotiating a secure connection with a remote client during [HTTP Transmissions API](#) request handling.

**Default:** true

**Example:**

```
<transmissions-api-listener 0/0:443>
  tls-allow-tls1.3 true
</transmissions-api-listener>
```

#### 3.3.21.3 tls-allow-tls1.2

Allows PowerMTA to use TLSv1.2 when negotiating a secure connection with a remote client during [HTTP Transmissions API](#) request handling.

**Default:** true

**Example:**

```
<transmissions-api-listener 0/0:443>
  tls-allow-tls1.2 true
</transmissions-api-listener>
```

#### 3.3.21.4 tls-allow-tls1.1

Allows PowerMTA to use TLSv1.1 when negotiating a secure connection with a remote client during [HTTP Transmissions API](#) request handling.

**Default:** true

**Example:**

```
<transmissions-api-listener 0/0:443>
  tls-allow-tls1.1 true
</transmissions-api-listener>
```

#### 3.3.21.5 tls-allow-tls1

Allows PowerMTA to use TLSv1 when negotiating a secure connection with a remote client during [HTTP Transmissions API](#) request handling.

**Default:** false

**Example:**

```
<transmissions-api-listener 0/0:443>
  tls-allow-tls1 true
</transmissions-api-listener>
```

#### 3.3.21.6 tls-allow-ssl3

This directive is no longer valid as PowerMTA no longer supports SSLv3.

#### 3.3.21.7 tls-ciphers

Specifies the ciphers that PowerMTA is allowed to use when negotiating a session using a TLSv1.2 or earlier protocol with a remote client during [HTTP Transmissions API](#) request handling. Please see documentation for OpenSSL command 'ciphers' for valid cipher names that can be used with this.

**Default:** HIGH:!ADH:@STRENGTH

**Example:**

```
<transmissions-api-listener 0/0:443>
  tls-ciphers HIGH:!ADH:@STRENGTH
</transmissions-api-listener>
```

### 3.3.21.8 tls-tls1.3-ciphersuites

Specifies the ciphersuites that PowerMTA is allowed to use when negotiating a session using the TLSv1.3 protocol with a remote client during [HTTP Transmissions API](#) request handling. Please see documentation for OpenSSL command 'ciphers' for valid ciphersuite names that can be used with this.

**Default:**

TLS\_AES\_256\_GCM\_SHA384:TLS\_CHACHA20\_POLY1305\_SHA256:TLS\_AES\_128\_GCM\_SHA256:TLS\_AES\_128\_CCM\_SHA256:TLS\_AES\_128\_CCM\_8\_SHA256

**Example:**

```
<transmissions-api-listener 0/0:443>
  tls-tls1.3-ciphersuites TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
</transmissions-api-listener>
```

### 3.3.21.9 tls-ca-file

Specifies a "CA file" containing Certification Authority certificates to allow verifying any certificates presented by clients during [HTTP Transmissions API](#) request handling. The certificates must be in PEM format.

**Example:**

```
<transmissions-api-listener 0/0:443>
  tls-ca-file /etc/pmta/certchain.pem
</transmissions-api-listener>
```

## 3.3.22 <proxy> Directives

The proxy tag is used to configure a proxy server for use by a virtual MTA. It must be named. A proxy tag must have both `server` and `client` directives defined within it.

### 3.3.22.1 server

The server directive within a proxy tag is used to configure the hostname or the IP address and the port of the proxy server.

**Syntax:**

```
server {IP | hostname }:PORT
```

**Example**

```
<proxy mta-proxy-a>
  server 10.1.2.3:5000
  client 16.80.99.192:0 client.hostname.com
</proxy>

<proxy mta-proxy-b>
  server proxy.internal.hostname.com:9999
  client 16.80.77.101:0 client2.hostname.com
</proxy>
```

### 3.3.22.2 client

The client directive within a proxy tag is be used to configure the IP address that the proxy server must use when connecting to the remote destinations. A port number is not required, and if unspecified, the port number will default to zero. A hostname must be defined.

**Syntax:**

```
client IP[:PORT] hostname
```

**Example**

```
<proxy mta-proxy-a>
  server 10.1.2.3:5000
  client 16.80.99.192 client.hostname.com
</proxy>

<proxy mta-proxy-b>
  server proxy.internal.hostname.com:9999
  client 16.80.77.101 client2.hostname.com
</proxy>
```

## 4 PowerMTA Management

Although PowerMTA has been designed for being as low maintenance as possible, it does include various tools for management and troubleshooting. This chapter is intended as a guide to daily operations with PowerMTA, for those who wish to actively manage and monitor, as well as to troubleshooting the most commonly encountered problems.

### 4.1 Management

#### 4.1.1 Monitoring

There are a number of ways in which you can monitor PowerMTA. It has a web-based monitor built in, so you can simply point your web browser to the monitor's TCP port on PowerMTA's machine to view the mailer's current status and display information about its queue. This comes particularly handy when monitoring over longer periods of time. The web monitor is described in detail in [Web-Based Monitoring](#).

The same information from the web monitor also available through the `pmta` command line tool. In addition to displaying the command results in formatted plain text, `pmta` can also output these in XML, JSON or as DOM-Style variable lists, making it relatively easy to post-process and to integrate PowerMTA in existing management infrastructures. The `pmta` command line tool and the commands it accepts are described in [Command Line Tool](#).

The available [SparkPost Signals](#) service can analyze your email sending and response data to help you identify potential problems.

#### 4.1.2 Logging and Accounting

For greatest flexibility and best performance, PowerMTA maintains its own logging file. The operating system's log (i.e., the Event Log on Windows or the `syslog` on Unix) is only used for recording information on startup errors and similar situations where it may not be possible for PowerMTA to write to its own logging file.

The logging file is used for recording various information about PowerMTA, such as startup and shutdown messages, any errors found during execution, connection traces, etc. In general, the log file is the place where the mailer writes any error messages, and you should look there first when any problems arise.

In addition to the logging file, PowerMTA records detailed information on e-mails processed in a separate *accounting file*. Storing accounting data separate from other, more general, logging data allows for a more efficient accounting file format for easier post-processing. It also makes it easier to send feedback about problems with PowerMTA while keeping the accounting data undisclosed. The format of the accounting files is described in [Accounting Files and Statistics](#).

Both logging and accounting files are *rotated* automatically, that is, a new file is started and the older ones are kept for a number of days. The parameters for the rotation, including the number of (historical) files to keep can be described in [Logging Directives](#) and in [<acct-file> Directives](#).

#### 4.1.3 PowerMTA Startup and Shutdown

On Windows systems, PowerMTA runs as a service and as such is started up automatically by the operating system after booting. It also can be started and stopped both via the Control Panel and by using the `net start pmta` and `net stop pmta` commands from a Command Prompt window. Currently, PowerMTA must run under the SYSTEM account.

On Unix systems, PowerMTA runs as a daemon, i.e., a background process. A standard startup/shutdown script is provided in the system's `init.d` directory.

On either platform, two processes are started: `pmtad`, which is PowerMTA itself, and `pmtawatch`, which is PowerMTA's watchdog process. As the name suggests, `pmtawatch` watches over PowerMTA and re-starts it in case of a (PowerMTA) crash.

## 4.2 Troubleshooting

The following sections guide through troubleshooting the most commonly encountered problems.

#### 4.2.1 Troubleshooting Startup Problems

Startup problems typically occur right after installing PowerMTA or after having changed its configuration. If PowerMTA won't start up, you should look in the logging file a separate logging file called `pmta-startuperr.log` for an error message. On Windows, this file is located in the log subdirectory of the folder where PowerMTA was installed. On Unix systems, the logging file is located on `/var/log/pmta/log`.

If no logging file is present or if it does not seem to contain an applicable error message, you can try running PowerMTA on the command line. On Windows, open a Command Prompt window and change to PowerMTA's bin subdirectory. On Unix, ensure that `/usr/sbin` is in your path. Then start PowerMTA by using the command

```
pmtad --debug
```

That will start up PowerMTA and will have it write any error messages to the screen, rather than to the logging file. Most probably, the startup will fail displaying an error message, like in the example below:

```
# /usr/sbin/pmtad --debug
2018-07-03 18:05:37 Startup error: No logging file specified
```

You can then fix the problem and restart PowerMTA.

A common cause of startup problems when PowerMTA is first installed is TCP port conflicts. Applications that listen for (i.e., accept) connections on a TCP port need exclusive access to the port, so if you are running any other SMTP server software on the same machine, either PowerMTA or the other software will fail to obtain access to port number 25 (the standard port for the SMTP protocol). The same might also happen with the TCP port used for the built-in web monitor (port 8080 by default). In the case of the the web monitor, since 8080 is just a local convention, you can resolve the conflict by changing the [configuration](#) and specifying a different port that the web-based monitor could listen on.

The SMTP port conflict may be more difficult to solve. If you are *not* using PowerMTA for handling inbound traffic, i.e., you are just feeding PowerMTA through SMTP or the submission APIs and inbound traffic is handled by some other software, you can configure PowerMTA to use an alternative [SMTP port](#) and change your feeding software to use that port. You can normally pick any free TCP port above 1024; port numbers 2500 and 2525 are popular choices.

If, on the other hand, both PowerMTA and the other SMTP server software are to receive inbound traffic, you will need to give each of them a different IP address. Normally, applications listening for (i.e., accepting) connections on a TCP port will do so for all locally available IP addresses, so *both* PowerMTA and the other SMTP software need to be configured to use a specific [IP address](#) only for PowerMTA's configuration). Naturally, you will first need to allocate a new IP address and configure your operating system to respond to it.

#### 4.2.2 Troubleshooting Email Delivery Problems

Email delivery problems constitute by far the most common type of problem encountered. It is quite normal for some email not to be delivered immediately and for the delivery to have to be retried. The causes of such problems are often invalid email addresses, network or server failures or misconfigured software at the recipient's site.

If you notice email building up in the queue for a specific domain, the first step is to check the (DNS-based) routing information for the domain. If that information is available and seemingly correct, the next step would be to check whether any of the listed mailers is reachable. PowerMTA's `pmta` command line tool has a `resolve` option that allows you to do all of that in a single step:

```
$ pmta resolve --connect outlook.com
Querying 127.0.0.1 over UDP about MX outlook.com
Read response from 127.0.0.1
answers:      ttl=60
             outlook.com. 60 IN MX 5 outlook-com.olc.protection.outlook.com.

Querying 127.0.0.1 over UDP about A outlook-com.olc.protection.outlook.com
Read response from 127.0.0.1
answers:      ttl=2
             outlook-com.olc.protection.outlook.com. 2 IN A 104.47.48.33
             outlook-com.olc.protection.outlook.com. 2 IN A 104.47.49.33

status = StatusOk
pref host name                               IP addresses
```

```
-----
5 outlook-com.olc.protection.outlook.com 104.47.48.33 104.47.49.33

connecting from localhost (0.0.0.0) to outlook-com.olc.protection.outlook.com (104.47.48.33)
>>> 220 C01NAM05FT041.mail.protection.outlook.com Microsoft ESMTP MAIL Service ready at Thu, 7 Feb 2019
15:18:50 +0000
<<< EHLO localhost
>>> 250-C01NAM05FT041.mail.protection.outlook.com Hello
>>> 250-SIZE 49283072
>>> 250-PIPELINING
>>> 250-DSN
>>> 250-ENHANCEDSTATUSCODES
>>> 250-STARTTLS
>>> 250-8BITMIME
>>> 250-BINARYMIME
>>> 250-CHUNKING
>>> 250-SMTPUTF8
<<< STARTTLS
>>> 220 2.0.0 SMTP server ready
TLSv1.2 connected with 256-bit ECDHE-RSA-AES256-SHA384
Cert: /C=US/ST=Washington/L=Redmond/O=Microsoft Corporation/CN=mail.protection.outlook.com;
issuer=/C=BE/O=GlobalSign nv-sa/CN=GlobalSign Organization Validation CA - SHA256 - G3; verified=no
<<< EHLO localhost
>>> 250-C01NAM05FT041.mail.protection.outlook.com Hello
>>> 250-SIZE 49283072
>>> 250-PIPELINING
>>> 250-DSN
>>> 250-ENHANCEDSTATUSCODES
>>> 250-8BITMIME
>>> 250-BINARYMIME
>>> 250-CHUNKING
>>> 250-SMTPUTF8
<<< QUIT
>>> 221 2.0.0 Service closing transmission channel
closed outlook-com.olc.protection.outlook.com (104.47.48.33) in=601 out=48
```

In the example above, routing information for the domain outlook.com could be resolved. It listed one mail host, with preference level 5 with two IPv4 addresses. The --connect option specified that pmta attempts connecting to the mailhost (which it did, to "104.47.48.33").

When you enter this command during troubleshooting, either part of the command may fail. You may get an error during the DNS resolution part, like, for example "No such domain name" for a domain that is not registered in the DNS. Similarly, if the routing information is available but if the mailhosts themselves are not reachable, the connection attempt fails.

Simply connecting to the mailhosts does not preclude other problems delivering email, for the problem may be related to the specific messages queued, or the mailhosts may be temporarily rejecting email (e.g., returning a 4XX code to SMTP commands). To verify if this is happening, use the pmta command called [trace](#). The trace command instructs PowerMTA to retry email delivery to the domain immediately and log the connection. This way, you can see (in the logging file) exactly what happens when delivery is attempted.

The pmta trace command is useful for quickly checking on a domain. If you should need to trace connections to a domain over a longer period of time, enable logging for it in the configuration file by adding a <domain> entry, like in the example below:

```
<domain outlook.com>
  log-connections yes
  log-commands yes
</domain>
```

### 4.2.3 Troubleshooting DNS Name Resolution Problems

PowerMTA requires a recursive name service, i.e., a name server that accepts queries and performs the full DNS resolution.

Some name servers fail to do the recursive lookup and return non-recursive (referral-only) responses instead. These responses don't contain the information sought by PowerMTA, so it discards them and retries the query in the hope that whatever caused them is temporary, or that some other server will provide the full response. When such responses are received a number of times from the same name server, PowerMTA also writes a warning to its logging file.

Generally speaking, such non-recursive responses may have various causes. However, some DNS server software versions are known to send them when configured to refuse recursive service, or when its limit of concurrent recursive queries has been reached.

Non-recursive responses waste time and may have an impact on performance. If you find a name server refusing recursive queries too often, check the name server configuration for any quotas or access control lists. If you do not control the name server, you should consider no longer using it for PowerMTA and setting up your own.

### 4.2.4 Troubleshooting Email Reception/SMTP Feeding Problems

Problems receiving email are quite rare and happen most commonly when writing custom applications to feed PowerMTA. The most common mistake when writing such applications is to terminate lines with just a line feed, rather than a carriage return and line feed pair as mandated by the email standards.

The simplest way to debug incoming SMTP connections into PowerMTA is to enable debugging for a specific source IP address. You do so by adding a <source> entry in the configuration file, like in the example below,

```
<source 127.0.0.1>
  log-connections yes
  log-commands yes
  log-data yes
</source>
```

and reload the configuration by using the pmta reload command. This will cause PowerMTA to trace incoming connections to the logging file, which helps determine in what phase of the transaction (for example, in a RCPT command) the problem is occurring. If you suspect the problem to lay in non-printable characters or in the message's body, you can additionally enable log-data, which prints all data sent or received through the connection along with each byte's hexadecimal code.

### 4.2.5 In Case of a Crash

If PowerMTA crashes, its pmtawatch watchdog program will attempt to restart it and you should get an email informing you about the crash. Please forward this email to us at support@port25.com, so that we can look into the problem. Feel free to include any information you may find relevant, such as recent changes to the configuration or workload, etc.

On Unix systems, the core dump will be stored, compressed if possible, in PowerMTA's subdirectory on the /opt filesystem.

### 4.2.6 If All Else Fails

Check out <http://www.port25.com/support/> for technical support options from SparkPost. Evaluators and customers with a support agreement may also get support via email at [support@port25.com](mailto:support@port25.com) or over the phone at +1.410.750.7687.

## 5 Web-Based Monitoring

### 5.1 Overview

PowerMTA has a web-based monitoring console built in that provides real time display of PowerMTA's status and allows you to view information on the current message queue based on various criteria.

To access the web monitor, point your web browser to the monitor's TCP port on the machine you installed PowerMTA. For example, if you are reading this document on the machine running PowerMTA, you would use the URL <http://127.0.0.1:8080>. This also assumes that you did not change the default TCP port in the [configuration](#).

When you try the URL, you may get a message saying that the access is forbidden. This is because, to protect your privacy, access to the web monitor is denied unless authorized in the configuration. Authorization is done on a per source IP address basis, so you need to enter the IP address of the host running your web browser into the configuration.

By default, PowerMTA auto-creates a self signed certificate during installation to enable the use of HTTPS on the web monitor. The use of self signed certificate may trigger warnings from the browser when you access the web monitor. You can eliminate these warnings shown by browsers by installing a signed certificate for use with the web monitor. You can obtain such a signed certificate from any well known certificate signer entity (for example, from *Lets Encrypt*). Once you obtain a signed certificate, overwrite the existing certificate in the file `/var/lib/pmta/webmon.pem` on Unix, or `<PowerMTA install directory>\db\webmon.pem` on Windows, and restarting the web monitor. The web monitor will now use and present the signed certificate.

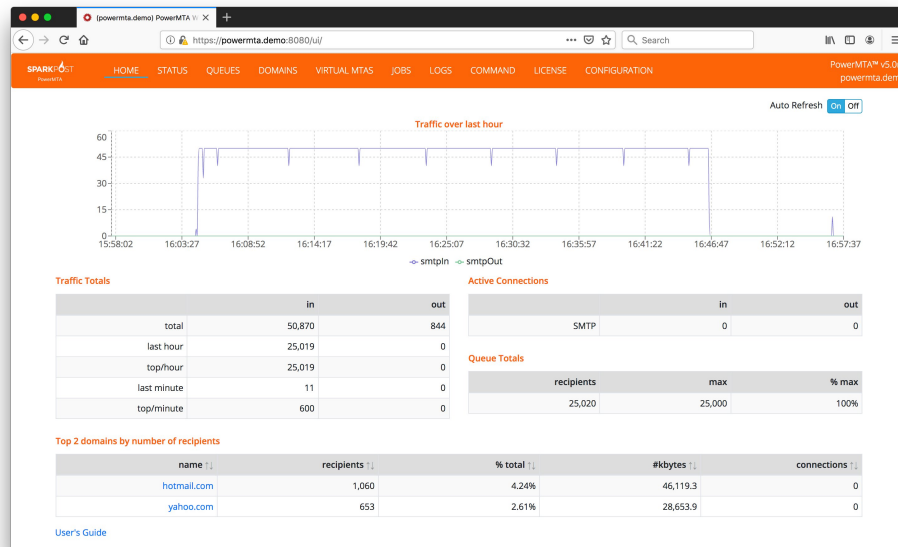
#### Note: Accessing the web monitor using Internet Explorer on Windows

During installation of PowerMTA, the web monitor will configure a self-signed certificate for enabling HTTPS support. We have noticed that accessing web monitor pages on Windows using Internet Explorer is very slow. This issue stems from the use of self-signed certificates for providing HTTPS. A workaround is to install a signed certificate from a well known signing entity such as *Lets Encrypt* and configuring the web monitor to present the certificate as documented above.

#### Note: Accessing the web monitor with XML and JSON formats

The web monitor URL <http://127.0.0.1:8080> by default is redirected to the URL with suffix `/ui`. To obtain XML or JSON formatted output, you need to use the URL without `/ui` suffix. Refer to the documentation of individual pages below for exact URLs.

### 5.2 Home Display



The initial "Home" page displays basic information about your installation. In addition to this information, there are some useful links:

From the home page, you have the option of changing to other displays:

- **Status**, the status monitor page, which displays various information about PowerMTA's status, updated every few seconds.
- **Queues**, which presents you with various options for listing queues to which email is queued.
- **Domains**, which presents you with various options for listing domains to which email is queued.
- **VirtualMTAs**, which displays a summary of the VirtualMTAs in use.
- **Jobs**, which presents you with summary information on any jobs (groups of messages) queued.
- **Logs**, allows for the viewing and downloading of PowerMTA's various log files.
- **Command** – This link allows you to run "pmta" commands as if you were logged in a the command/shell prompt of the PowerMTA server. This link is displayed only when PowerMTA service is *running* on the machine.
- **License** – This links to a form where you can view/add/change the PowerMTA license.
- **Configuration** – This allows you to view and edit the configuration file from within the web monitor window. After saving changes, a "pmta reload" will not be required as it is done automatically. If the change requires a restart of the service, that will have to be done manually.
- **User's Guide** – Opens UsersGuide.pdf in a new window for ease of looking up PowerMTA information.

This page can also be retrieved in XML or JSON format for processing by a program. You specify the format by using `format`

parameter.

E.g. use the URL: <http://127.0.0.1:8080?format=xml>

The following sections describe these pages in more detail.

## 5.3 Status Monitor Display

The status monitor displays data from various components within PowerMTA.

| Inbound Traffic |        |       |           | Outbound Traffic |       |      |        |
|-----------------|--------|-------|-----------|------------------|-------|------|--------|
|                 | rcpts  | msgs  | kbytes    |                  | rcpts | msgs | kbytes |
| total           | 50,870 | 5,348 | 229,928.2 | total            | 844   | 844  | 33,956 |
| last hour       | 25,019 | 2,633 | 112,997.5 | last hour        | 0     | 0    | 0      |
| top/hour        | 25,019 | 2,633 | 112,997.5 | top/hour         | 0     | 0    | 0      |
| last minute     | 0      | 0     | 0         | last minute      | 0     | 0    | 0      |
| top/minute      | 600    | 69    | 2,904.9   | top/minute       | 0     | 0    | 0      |

| Connections |    |       | Domain Names |       | Spool          |       |
|-------------|----|-------|--------------|-------|----------------|-------|
|             | in | out   |              | total |                | total |
| active      | 0  | 0     | cached       | 0     | files          | 2,634 |
| top         | 1  | 0     | pending      | 0     | directories    | 1     |
| max         | 5  | 1,200 |              |       | initialization | 100%  |

| Queues  |        | domains |       | kbytes |             |
|---------|--------|---------|-------|--------|-------------|
|         | rcpts  |         |       |        |             |
| SMTP    | 24,966 |         | 2,223 |        | 1,092,768.5 |
| discard | 1      |         | 1     |        | 40          |
| file    | 0      |         | 0     |        | 0           |
| pipe    | 0      |         | 0     |        | 0           |

Status: running Started On: 2019-09-13 16:56:41 Uptime: 0:00:01:10

The "Total" counters show the total data received and sent since the mailer was started. Similarly, the "Last Hour" and "Last Minute" counters show the total data received and sent for the respective time intervals.

The "Top/Hour" and "Top/Minute" lines show the highest throughput observed per hour and minute. The various counters are measured independently of each other, i.e., the top number of inbound recipients may or may not have happened at the same time as the top number of inbound kilobytes.

The "Connections" section shows the number of currently active connections, the top number of simultaneous connections made since last restart, and the maximum number of inbound and outbound simultaneous connections available given your license activation key and configured limits.

The "Domain Names" section shows the number of DNS entries currently in PowerMTA's internal cache, as well as the number of DNS queries pending, i.e., currently taking place or scheduled and waiting to take place.

The "Spool Files" section shows the number of spool files currently being used for storing queued messages. And, since the spool management code recycles files for efficiency instead of having to recreate as needed, this section also tracks the number of files that have been recycled and are waiting to be reused.

The "Spool Initialization" section shows whether the initialization is in progress or has been completed. This initialization takes place at startup and consists of reading the message files in the spool and loading queue information into memory.

The "Queues" section provides top level counts for each of the delivery methods available: SMTP or through the pipe delivery API. The counts include the total number of recipients currently in the queue, total number of destination domains for these recipients, and the total amount of data in the mail queues to be delivered, for each method.

This page can also be retrieved in XML format for processing by a program. You specify the XML format by using `format=xml`. For example, you could use the `wget` utility to retrieve it:

(Actual output will not be pretty-printed)

```
$ wget -O- -q http://127.0.0.1:8080/status?format=xml
<rsp>
  <data>
    <mta>
      <product>
        <name>PowerMTA</name>
        <version>5.0r1</version>
        <buildDate>2019-09-15 14:27:54</buildDate>
        <revision>v5_0r1@190914.919ffeb</revision>
        <bits>64</bits>
      </product>
      <lak>
        <serial>123456</serial>
        <instances>1</instances>
      </lak>
      <os>
        <name>Linux</name>
        <version>3.10.0-957.1.3.el7.x86_64</version>
        <build>CentOS Linux release 7.6.1810 (Core) </build>
      </os>
      <cpu>
        <type>x86_64</type>
        <count>2</count>
      </cpu>
      <ram>
        <real>3878876</real>
      </ram>
      <fullHostName>hazmat.port25.com</fullHostName>
      <userString/>
      <hostId>0</hostId>
      <webUI>
```

```
<protocol>1</protocol>
</webUI>
</mta>
<status>
  <timeNow>2019-02-07 17:06:02</timeNow>
  <startupTime>2019-01-31 02:48:06</startupTime>
  <shuttingDown>0</shuttingDown>
  <traffic>
    <total>
      <out>
        <rcp>85402</rcp>
        <msg>85402</msg>
        <kb>2736957.8</kb>
      </out>
      <in>
        <rcp>100640</rcp>
        <msg>100544</msg>
        <kb>2177515.3</kb>
      </in>
      <bounceProcessed>0</bounceProcessed>
      <feedbackLoopProcessed>0</feedbackLoopProcessed>
    </total>
    <lastHr>
      <out>
        <rcp>772</rcp>
        <msg>772</msg>
        <kb>25208.2</kb>
      </out>
      <in>
        <rcp>984</rcp>
        <msg>984</msg>
        <kb>21202.5</kb>
      </in>
      <bounceProcessed>0</bounceProcessed>
      <feedbackLoopProcessed>0</feedbackLoopProcessed>
    </lastHr>
    <lastMin>
      <out>
        <rcp>7</rcp>
        <msg>7</msg>
        <kb>54.0</kb>
      </out>
      <in>
        <rcp>10</rcp>
        <msg>10</msg>
        <kb>79.4</kb>
      </in>
      <bounceProcessed>0</bounceProcessed>
      <feedbackLoopProcessed>0</feedbackLoopProcessed>
    </lastMin>
    <topPerHr>
      <out>
        <rcp>883</rcp>
        <msg>883</msg>
        <kb>73238.2</kb>
      </out>
      <in>
        <rcp>1039</rcp>
        <msg>1039</msg>
        <kb>52393.0</kb>
      </in>
      <bounceProcessed>0</bounceProcessed>
      <feedbackLoopProcessed>0</feedbackLoopProcessed>
    </topPerHr>
    <topPerMin>
      <out>
        <rcp>158</rcp>
        <msg>158</msg>
        <kb>27369.0</kb>
      </out>
      <in>
        <rcp>188</rcp>
        <msg>187</msg>
        <kb>27269.5</kb>
      </in>
      <bounceProcessed>0</bounceProcessed>
      <feedbackLoopProcessed>0</feedbackLoopProcessed>
    </topPerMin>
  </perf>
  <lastMin>
    <in>
      <count>0</count>
      <mailFromTime>0</mailFromTime>
      <dataTime>0</dataTime>
      <totalTime>0</totalTime>
    </in>
  </lastMin>
</perf>
</traffic>
<conn>
  <smtpIn>
    <cur>3</cur>
    <max>200</max>
    <top>20</top>
  </smtpIn>
  <smtpOut>
    <cur>11</cur>
    <max>100</max>
    <top>76</top>
  </smtpOut>
</conn>
<resolver>
```

```

<namesCached>81</namesCached>
<queriesPending>1</queriesPending>
<queriesActive>1</queriesActive>
</resolver>
<queue>
  <smtp>
    <rcp>132</rcp>
    <dom>92</dom>
    <kb>7944.7</kb>
  </smtp>
  <pipe>
    <rcp>0</rcp>
    <dom>0</dom>
    <kb>0.0</kb>
  </pipe>
  <discard>
    <rcp>0</rcp>
    <dom>0</dom>
    <kb>0.0</kb>
  </discard>
  <file>
    <rcp>0</rcp>
    <dom>0</dom>
    <kb>0.0</kb>
  </file>
  <alias>
    <rcp>0</rcp>
    <dom>0</dom>
    <kb>0.0</kb>
  </alias>
  <bounceProcessor>
    <rcp>0</rcp>
    <dom>0</dom>
    <kb>0.0</kb>
  </bounceProcessor>
  <feedbackLoopProcessor>
    <rcp>0</rcp>
    <dom>0</dom>
    <kb>0.0</kb>
  </feedbackLoopProcessor>
</queue>
<spool>
  <initPct>100</initPct>
  <dirs>3</dirs>
  <files>
    <inUse>134</inUse>
    <recycled>0</recycled>
    <total>12288</total>
  </files>
  <totalRcp>134</totalRcp>
  <maxRcp>288000</maxRcp>
</spool>
<status>running</status>
</data>
<succ/>
</rsp>

```

Similarly, this page can also be retrieved in JSON format for processing by a program. You specify the JSON format by using `format=json`. For example, you could use the `wget` utility to retrieve it:

(Actual output will not be pretty-printed)

```

$ wget -O- -q http://127.0.0.1:8080/status?format=json
{
  "data": {
    "mta": {
      "product": {
        "name": "PowerMTA",
        "version": "5.0b1",
        "buildDate": "2018-12-17 04:13:56",
        "revision": "trunk@18274",
        "bits": 64
      },
      "lak": {
        "serial": "123456",
        "instances": 0
      },
      "os": {
        "name": "Linux",
        "version": "3.10.0-862.3.2.el7.x86_64",
        "build": "CentOS Linux release 7.5.1804 (Core) "
      },
      "cpu": {
        "type": "x86_64",
        "count": 2
      },
      "ram": {
        "real": 3879568
      },
      "fullHostName": "foo.int.port25.com",
      "userString": "",
      "hostId": 0
    },
    "status": {
      "timeNow": "2019-02-19 23:00:46",
      "startupTime": "2019-02-19 17:15:07",
      "shuttingDown": false,
      "traffic": {
        "total": {
          "out": {
            "rcp": 2007007,

```

```

        "msg": 2007007,
        "kb": 1370100.3
    },
    "in": {
        "rcp": 2007845,
        "msg": 28046,
        "kb": 8799.7
    },
    "bounceProcessed": 0,
    "feedbackLoopProcessed": 0
},
"lastHr": {
    "out": {
        "rcp": 348704,
        "msg": 348704,
        "kb": 142245.3
    },
    "in": {
        "rcp": 348700,
        "msg": 4936,
        "kb": 33.7
    },
    "bounceProcessed": 0,
    "feedbackLoopProcessed": 0
},
"lastMin": {
    "out": {
        "rcp": 5809,
        "msg": 5809,
        "kb": 2372.6
    },
    "in": {
        "rcp": 5800,
        "msg": 81,
        "kb": 0.6
    },
    "bounceProcessed": 0,
    "feedbackLoopProcessed": 0
},
"topPerHr": {
    "out": {
        "rcp": 348947,
        "msg": 348947,
        "kb": 693034.3
    },
    "in": {
        "rcp": 348700,
        "msg": 4936,
        "kb": 8633.6
    },
    "bounceProcessed": 0,
    "feedbackLoopProcessed": 0
},
"topPerMin": {
    "out": {
        "rcp": 6219,
        "msg": 6219,
        "kb": 214286.9
    },
    "in": {
        "rcp": 5900,
        "msg": 99,
        "kb": 3480.8
    },
    "bounceProcessed": 0,
    "feedbackLoopProcessed": 0
},
"perf": {
    "lastMin": {
        "in": {
            "count": 0,
            "mailFromTime": 0,
            "dateTime": 0,
            "totalTime": 0
        }
    }
}
},
"conn": {
    "smtpIn": {
        "cur": 1,
        "max": 5,
        "top": 1
    },
    "smtpOut": {
        "cur": 0,
        "max": 1200,
        "top": 157
    }
}
},
"resolver": {
    "namesCached": 589,
    "queriesPending": 0,
    "queriesActive": 0
},
"queue": {
    "smtp": {
        "rcp": 805,
        "dom": 226,
        "kb": 5.5
    },
    "pipe": {

```

```

        "rcp": 0,
        "dom": 0,
        "kb": 0
    },
    "discard": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
    },
    "file": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
    },
    "alias": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
    },
    "bounceProcessor": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
    },
    "feedbackLoopProcessor": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
    }
},
"spool": {
    "initPct": 100,
    "dirs": 1,
    "files": {
        "inUse": 504,
        "recycled": 0,
        "total": 504
    },
    "totalRcp": 805,
    "maxRcp": 25000
},
"status": "running"
}
},
"status": "success",
"message": ""
}

```

## 5.4 Queues Display

Similar to the "Top Domains" page, the "Top Queues" page provides a listing of the top ten queues ranked by number of recipients.

For greatest flexibility and control, PowerMTA groups messages for a particular domain for a particular VirtualMTA in its own separate queue. For example, mail destined for msn.com and handled by VirtualMTA mta5 would be in a different queue than mail destined for msn.com and handled by VirtualMTA mta10.

Whereas the "Top Domains" page provides a global view with the top ten domains across all queues, the "Top Queues" page provides a listing and counts for specific queues.

Top 2 queues by number of recipients

| Name                | #Rcpt | #KBytes  | #Conn | Paused | Mode   | Last Error |
|---------------------|-------|----------|-------|--------|--------|------------|
| hotmail.com/vmta110 | 1,060 | 46,119.3 | 0     | yes    | normal |            |
| gmail.com/vmta402   | 6     | 284.9    | 0     | yes    | normal |            |

2 of 23029 matching queues.

Queue name:

Sort by: ☐ name ☒ # of recipients ☐ queue size

Connections: ☐ connected only ☐ unconnected only ☒ both

Paused: ☐ paused only ☐ running only ☒ both

Mode: ☐ normal only ☐ backoff only ☒ any

Maximum # of items to display:

The individual queue pages also include data on the last 50 recipient events, the last 10 queue errors and the last 10 mode change (backoff/normal) events.

Queue Detail

| Name                | #Rcpt | #KBytes  | #Conn | Paused | Mode   | Backoff Reason | Next Retry |
|---------------------|-------|----------|-------|--------|--------|----------------|------------|
| hotmail.com/vmta110 | 1,060 | 46,119.3 | 0     | yes    | normal |                |            |

Last Errors

| Time   | Last Error |
|--------|------------|
| (none) |            |

Mode Change Events

Show

Recipient Events

Show

VMTA Settings

Show

Queue Settings

Show

This information is also available in the output of the `pmta show queues` command.

This page can also be retrieved in XML or JSON format for processing by a program. You specify the format by using `format` parameter.

E.g. for obtaining top queues, use the URL: <http://127.0.0.1:8080/queues?format=xml>

E.g. for obtaining an individual queue's detail, use the URL: <http://127.0.0.1:8080/queueDetail?format=xml&queue=domain.com/VirtualMTA>

## 5.5 Queues “View Options”

The "Queues" page consists of a Queues “view options” form which after submitting, generates an output similar in format to the "Top Queues" page, but for the specific queue(s) specified in conjunction with relevant filters.

The filters include:

- connected queues only, unconnected queues only, or both
- queues in normal mode, backoff mode, or any mode
- the desired number of queues to list

One can then choose how to sort the output, either by queue name, number of recipients, or queue size.

Queue name

\*/\*

Sort by

☐ name
 ☒ # of recipients
 ☐ queue size

Connections

☐ connected only
 ☐ unconnected only
 ☒ both

Paused

☐ paused only
 ☐ running only
 ☒ both

Mode

☐ normal only
 ☐ backoff only
 ☒ any

Maximum # of items to display

10

Update

## 5.6 Domains Display

The "Top Domains" page provides a listing of the top ten domains in the queue ranked by number of recipients.

SPARKPOST

HOMESTATUSQUEUESDOMAINSVIRTUAL MTASJOBSLOGSCOMMANDLICENSECONFIGURATION

PowerMTA™ v5.0r1  
powermta demo

Auto Refresh ☒ On ☐ Off

Top 2 domains by number of recipients

| Name ↕      | #Rcpt ↕ | #KBytes ↕ | #Conn ↕ | Last Error ↕ |
|-------------|---------|-----------|---------|--------------|
| hotmail.com | 1,060   | 46,119.3  | 0       |              |
| yahoo.com   | 653     | 28,653.9  | 0       |              |

2 of 2226 matching domains.

Domain name

\*/

VMTA

\*/

Sort by

☐ name
 ☒ # of recipients
 ☐ queue size

Connections

☐ connected only
 ☐ unconnected only
 ☒ both

Maximum # of items to display

10

Update

For each domain listed, the number of recipients, amount of data (kilobytes), the number of open connections and the last error occurred while delivering to the domain are displayed.

## 5.7 Domains “View Options”

The "Domains" page consists of a Domains “view options” form which after submitting, generates an output similar in format to the

"Top Domains" page, but for the specific domain(s) and based on the other parameters specified.

|                                       |                                |                                                  |                                        |
|---------------------------------------|--------------------------------|--------------------------------------------------|----------------------------------------|
| Domain name                           | VMTA                           | Sort by                                          | Connections                            |
| <input type="text" value="*"/>        | <input type="text" value="*"/> | <input type="radio"/> name                       | <input type="radio"/> connected only   |
| Maximum # of items to display         |                                | <input checked="" type="radio"/> # of recipients | <input type="radio"/> unconnected only |
| <input type="text" value="10"/>       |                                | <input type="radio"/> queue size                 | <input checked="" type="radio"/> both  |
| <input type="button" value="Update"/> |                                |                                                  |                                        |

This page can also be retrieved in XML or JSON format for processing by a program. You specify the format by using `format` parameter.

E.g. for obtaining top domains, use the URL: <http://127.0.0.1:8080/domains?format=xml>

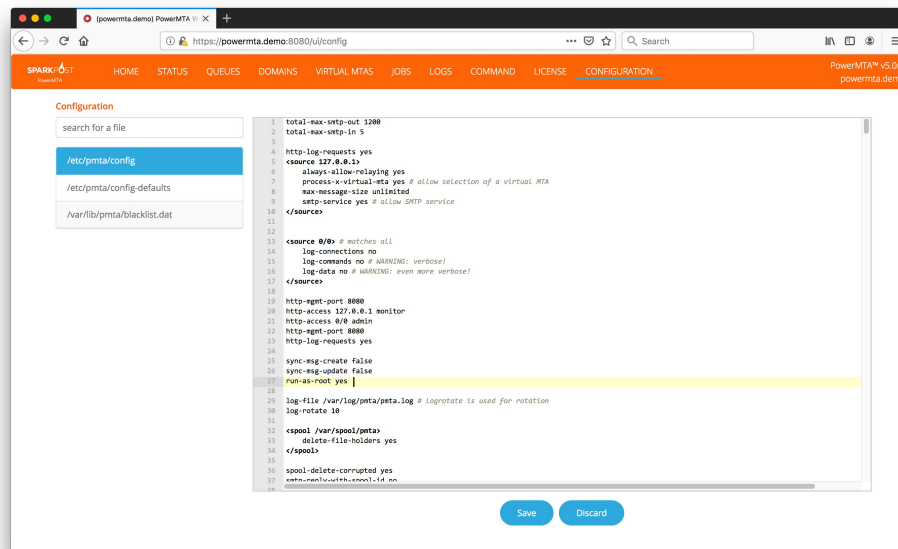
E.g. for obtaining an individual domain's detail, use the URL: <http://127.0.0.1:8080/domainDetail?format=xml&domain=domain.com>

## 5.8 Configuration Page

The "Configuration" page can be used to view and edit PowerMTA's main configuration as well as any other files that were included in the configuration through the [include](#).

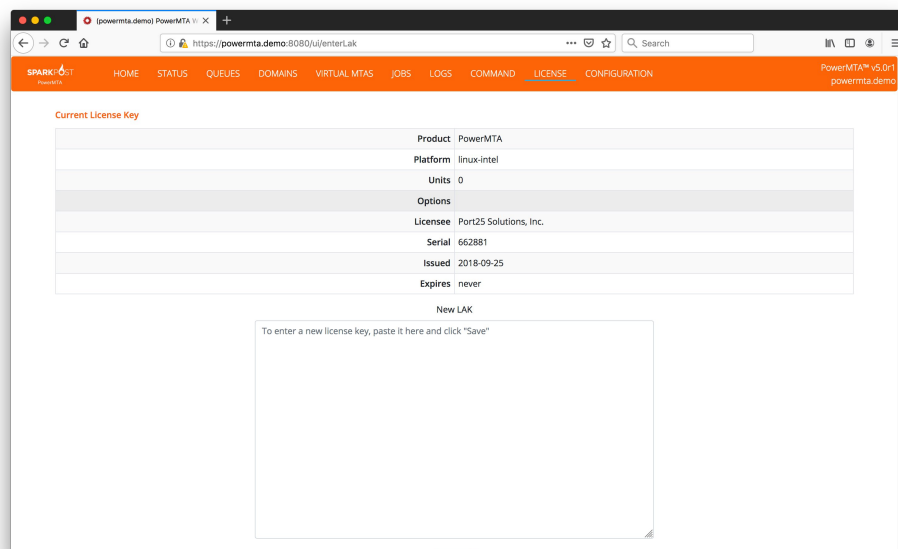
### Note

While the `config-defaults` file is also shown in the output, you should not modify the contents of this file since they get overwritten during PowerMTA installation or upgrade.



## 5.9 License Page

The "License" page can be used to view and edit PowerMTA's license file.



## 5.10 Common "Last Error"s

Included below is a list of the most common "Last Error" messages, with an explanation about what may cause them. On Windows, system status codes may be prefixed with WSA, but indicate generally the same condition.

### ETIMEDOUT in connection to *domain (IP)*

This error indicates that the connection failed because, while connected, the remote host did not properly respond within a period of time. It possibly indicates that the remote host is overwhelmed or that there are connectivity problems. PowerMTA's connection timeout parameter is 10 minutes, and is not configurable.

### ECONNREFUSED connecting to *domain (IP)*

This error indicates that no connection could be made because the remote host actively refused it. This usually results when the remote mailer software is temporarily down or inactive, or if the remote site is actively refusing connections (usually with a packet filter) from your company and/or IP address.

### EDISCON in connection to *domain (IP)*

This error indicates that the remote host has unexpectedly closed the connection. It may result from the remote mail server being taken down, crashing, or simply closing the connection, possibly as a "last resort" kind of response to message contents it could not process.

### StatusDnsQueryFailed resolving domain

This error indicates that the DNS query for the domain could not be completed. This could be due to more than one reason, specifically timeouts when PowerMTA attempts to talk to your local DNS server, but more likely, timeouts in your local DNS server itself when attempting to talk to other DNS servers. Although these errors can and will be seen for some domains that appear to not have valid MX or A records, the fact that the actual DNS queries timeout before receiving any verification of this precludes PowerMTA from making any assumptions and thus, for example, bouncing mail to these domains immediately.

### EHLO failed

These are errors that are returned by the remote mail server during the initial SMTP handshake (at the HELO or EHLO SMTP commands). Usually the text included in the error messages is self-explanatory. Here are a few examples:

```
421 domain Service not available - too busy
421 domain Sorry, you are not authorized to make this connection.
421 domain temporarily not accepting mail
452 domain Too busy, please try later.
```

If you see domains in the queue with no errors specified, along with no active connections, this would mean that the queue is most likely to be waiting for DNS resolution to complete. For these domains, we recommend running the [trace](#) command with the `--log-resolution` option from the command line interface or enable one or more of the various types of [logging](#) available.

## 5.11 VirtualMTAs Display

| Name    | Domains | #Rcpt | %Total | #KBytes  | #Conn |
|---------|---------|-------|--------|----------|-------|
| vmta110 | 34      | 1,093 | 4%     | 47,471.9 | 0     |
| vmta318 | 71      | 77    | 0%     | 3,293.5  | 0     |

2 of 964 virtual MTAs.

The VirtualMTAs page provides a listing of the VirtualMTAs selected that currently have messages in the queue. The VirtualMTAs are listed sorted by number of recipients.

Each VirtualMTA listed also links to a top domains page that subsequently lists the top 10 domains in the queue for the specific VirtualMTA, ranked by number of recipients in the queue. Also included on this page is the volume of data in the queue, number of current connections open to the domain, and the last error received for the domain, again for the specific VirtualMTA.

This page can also be retrieved in XML or JSON format for processing by a program. You specify the format by using `format` parameter.

E.g. for obtaining top Virtual MTAs, use the URL: <http://127.0.0.1:8080/vmtas?format=xml>

E.g. for obtaining an individual Virtual MTAs's detail, use the URL: [http://127.0.0.1:8080/queues?format=xml&queue=\\*/VirtualMTA](http://127.0.0.1:8080/queues?format=xml&queue=*/VirtualMTA)

## 5.12 Jobs Display

| Name                | Domains | #Rcpt | #KBytes  | Paused | Scheduled Start Time |
|---------------------|---------|-------|----------|--------|----------------------|
| jobid35589020763324 | 207     | 315   | 13,806.7 | no     |                      |
| jobid35589020763326 | 160     | 234   | 10,152   | no     |                      |

2 of 15 jobs.

For each job listed, the number of recipients and amount of data (kilobytes) are displayed.

This page can also be retrieved in XML or JSON format for processing by a program. You specify the format by using `format` parameter.

E.g. for obtaining top jobs, use the URL: <http://127.0.0.1:8080/jobs?format=xml>

E.g. for obtaining an individual job's detail, use the URL: <http://127.0.0.1:8080/jobDetail?format=xml&jobid=123>

## 5.13 Logs Display

The Logs page displays the various log and accounting files currently available in PowerMTA. By choosing the download link, these files can be downloaded directly to your local machine for viewing. In addition, these files can be accessed programmatically via the URL format: <http://127.0.0.1:8080/getFile?file=log.txt> where `log.txt` is the name of the file you want to download.

| File           | Date                | Size | Download                 |
|----------------|---------------------|------|--------------------------|
| pmta.log       | 2019-09-13 17:01:41 | 958K | <a href="#">download</a> |
| pmtahttp.log   | 2019-09-13 17:01:12 | 348K | <a href="#">download</a> |
| pmta.log.1     | 2019-09-13 00:00:00 | 233M | <a href="#">download</a> |
| pmtahttp.log.1 | 2019-09-13 00:00:00 | 503K | <a href="#">download</a> |
| pmta.log.2     | 2019-09-12 00:00:00 | 219K | <a href="#">download</a> |
| pmtahttp.log.2 | 2019-09-12 00:00:00 | 350K | <a href="#">download</a> |
| pmta.log.3     | 2019-09-11 00:00:00 | 218K | <a href="#">download</a> |
| pmtahttp.log.3 | 2019-09-11 00:00:00 | 350K | <a href="#">download</a> |
| pmta.log.4     | 2019-09-10 00:00:00 | 336K | <a href="#">download</a> |
| pmtahttp.log.4 | 2019-09-10 00:00:00 | 351K | <a href="#">download</a> |
| pmta.log.5     | 2019-09-09 00:00:00 | 217K | <a href="#">download</a> |
| pmtahttp.log.5 | 2019-09-09 00:00:00 | 350K | <a href="#">download</a> |
| pmta.log.6     | 2019-09-08 00:00:00 | 217K | <a href="#">download</a> |
| pmtahttp.log.6 | 2019-09-08 00:00:00 | 350K | <a href="#">download</a> |
| pmta.log.7     | 2019-09-07 00:00:00 | 216K | <a href="#">download</a> |
| pmtahttp.log.7 | 2019-09-07 00:00:00 | 350K | <a href="#">download</a> |
| pmta.log.8     | 2019-09-06 00:00:00 | 216K | <a href="#">download</a> |
| pmtahttp.log.8 | 2019-09-06 00:00:00 | 350K | <a href="#">download</a> |
| pmta.log.9     | 2019-09-05 00:00:00 | 216K | <a href="#">download</a> |

This page can also be retrieved in XML or JSON format for processing by a program. You specify the format by using `format` parameter.

E.g. for obtaining logs, use the URL: <http://127.0.0.1:8080/logs?format=xml>

## 5.14 Listing of local IPs

To get a list of the IPs currently configured on your machine, use the following URL in your web browser:

<http://127.0.0.1:8080/getlocalips>

```
-<rsp>
  -<localIps>
    <ip>127.0.0.1</ip>
    <ip>10.59.14.69</ip>
    <ip>::1</ip>
    <ip>fe80::42f:6eff:fe85:8eec</ip>
  </localIps>
  <succ/>
</rsp>
```

## 5.15 Listing of emails in queue

To get a listing of the messages that are currently in the queue, use a URL similar to the following in your web browser:

<http://127.0.0.1:8080/listqueue?format=xml&queue=domain.com/VirtualMTA&jobid=12>

[http://127.0.0.1:8080/listqueue?format=xml&queue=yahoo.com/\\*&jobid=12&envid=3](http://127.0.0.1:8080/listqueue?format=xml&queue=yahoo.com/*&jobid=12&envid=3)

[http://127.0.0.1:8080/listqueue?format=xml&queue=\\*/vmta12&rcpt=test@port25.com](http://127.0.0.1:8080/listqueue?format=xml&queue=*/vmta12&rcpt=test@port25.com)

[http://127.0.0.1:8080/listqueue?format=xml&queue=/\\*&orig=test@port25.com](http://127.0.0.1:8080/listqueue?format=xml&queue=/*&orig=test@port25.com)

Use of jobid is optional. Replace domain.com with the domain for which you are looking. Also replace VirtualMTA with the name of the VirtualMTA in question. Also, use \* for all domains or all VirtualMTAs.

```
-<rsp>
  -<data>
    -<msg>
      <queue>öbb.de/vmta979</queue>
      <orig>پنجابی@bounce.port25.com</orig>
      <rcpt>H@öbb.de</rcpt>
      <file>/var/spool/pmta/0/000001D6</file>
      <timeQueued>2018-11-06 16:19:22</timeQueued>
      <header_subject/>
      <priority>50</priority>
    </msg>
    -<msg>
      <queue>öbb.de/vmta979</queue>
      <orig>پنجابی@bounce.port25.com</orig>
      <rcpt>B@öbb.de</rcpt>
      <file>/var/spool/pmta/0/000001D6</file>
      <timeQueued>2018-11-06 16:19:22</timeQueued>
      <header_subject/>
      <priority>50</priority>
    </msg>
    -<msg>
      <queue>öbb.de/vmta979</queue>
      <orig>پنجابی@bounce.port25.com</orig>
      <rcpt>A@öbb.de</rcpt>
      <file>/var/spool/pmta/0/000001D6</file>
      <timeQueued>2018-11-06 16:19:22</timeQueued>
      <header_subject/>
      <priority>50</priority>
    </msg>
  +<msg></msg>
  +<msg></msg>
</data>
-<succ>
  1 queue busy while listing; consider using the --pause option.
</succ>
</rsp>
```

## 6 Command Line Tool

### 6.1 Overview

PowerMTA comes with a command line tool that allows you to execute commands, display various information and also generally aids in investigating delivery problems. On Windows, this tool can be found under bin\pmta.exe in the folder where PowerMTA was installed. On Unix, it is installed as /usr/sbin/pmta. At the command line, you can get a quick listing of the commands available by invoking pmta --help:

PowerMTA(TM) v5.0b1 command tool  
Copyright(c) 1999-2018, Port25 Solutions, Inc. All Rights Reserved.

```

Usage: pmta [options] command

where 'options' are:
  --help          display this message
  --xml           display result in XML
  --dom          display result as DOM-like "variables"
  --json         display result in JSON

and where 'command' is one of:
  clear dnscache domainname
  check mfrom [--tcp] [--dumppackets] mfrom ip
  check spf [--tcp] [--dumppackets] ip heloDomain [mailFrom]
  delete [--dsn] [--queue=domain[/vmta]] [--orig=addr] [--rcpt=addr] [--jobId=id] [--envId=id] [--older-
than=time-interval] [--no-accounting]
  deregister [--user=name] [--local-only] [--retain-unloaded]
  digest [--hmac-sha256] [--siphhash] [--key=string] [--hexkey=string] string
  disable source [--reenable-after=interval] ip domain[/vmta]
  enable source ip domain[/vmta]
  list [--queue=domain[/vmta]] [--orig=addr] [--rcpt=addr] [--jobId=id] [--envId=id] [--maxitems=n] [--
older-than=time-interval] [--pause] [--priority] [--schedule]
  pause queue domain[/vmta]
  pause job jobid
  register [--user=name] [--label=name] [--webmon-ip=ip] [--webmon-port=number] [--pmtmc-port=number] [-
reuse-label] pmtmc-hostname
  reload
  reset counters
  reset status
  resolve [--tcp] [--connect] [--use-dane] [--dumppackets] [--interactive] [--server=ip] [--source=
[host,]ip] domainname
  resume queue domain[/vmta]
  resume job jobid
  rotate acct [file]
  rotate log
  schedule [--retry-recipients] domain[/vmta]
  set priority [--queue=domain[/vmta]] [--orig=addr] [--rcpt=addr] [--jobId=id] [--envId=id] number
  set queue --mode={normal|backoff} domain[/vmta]
  show disabled sources [domain[/vmta]]
  show domains [--vmta=name] [--connected={yes|no}] [--maxitems=n] [--errors] [--sort={name|rcpt|size}]
[name]
  show jobs [--maxitems=n]
  show license
  show precache
  show queues [--connected={yes|no}] [--paused={yes|no}] [--mode={normal|backoff}] [--maxitems=n] [--
errors] [--sort={name|rcpt|size}] [--no-rcpt-events] [--mode-events] [domain[/vmta]]
  show registration
  show status
  show settings domain[/vmta]
  show topdomains [--vmta=name] [--connected={yes|no}] [--maxitems=n] [--errors]
  show topqueues [--connected={yes|no}] [--paused={yes|no}] [--mode={normal|backoff}] [--maxitems=n] [--
errors] [--no-rcpt-events] [domain[/vmta]]
  show version
  show vmtas [--maxitems=n]
  test config
  trace [--log-data] [--log-resolution] [--to-log] [--retry-recipients] [--source-ip=ip] domain[/vmta]

```

Note: On Windows, you can also use the "/" character instead of "--" (but "--" works on all platforms).

## 6.2 Command Output Formats

pmta supports four output formats:

- formatted plain text, intended for human consumption, e.g., for the system or services administrator executing PowerMTA commands. It is the default format if no options are passed to pmta.
- XML format, intended for XML-enabled applications. It is selected by specifying the --xml option to pmta.
- DOM-style variable listing, intended for non XML-capable applications and for quick Perl or shell scripts. It is selected by specifying the --dom option to pmta.
- JSON format, intended for JSON-enabled applications. It is selected by specifying the --json option to pmta.

### 6.2.1 Sample Command Outputs

Included below are three sample command outputs along with the commands and options used to generate each. For comparison purposes, all three are based on the show status command.

**Formatted Plain Text (pmta show status):**

```

PowerMTA v5.0 status on example.port25.com on 2018-09-10 10:16:19

Traffic      -----inbound-----      -----outbound-----
              rcpts      msgs      kbytes      rcpts      msgs      kbytes
Total        939437      5940      233409.5    939430      939430      34550887.2
Last Hour    939437      5940      233409.5    939263      939263      34544775.4
Top/Hour     939437      5940      233409.5    939263      939263      34544775.4
Last Min.    45063      298      11632.6     45021      45021      1658056.5
Top/Min.     69881      458      18005.6     69115      69115      2489680.7

Connections  active      top      maximum  Domain      cached      pending
Inbound      0           1         5         Names
Outbound     10000      10000     10000

Queues       rcpts      domains      kbytes      Spool      in use      recycled
SMTP         99459      2895        3985272.8   Files      5667        0
other        0           0           0.0         Init.      complete

Status       running     Started  2018-09-10 20:02:12  Uptime  0 0:20:01

```

**XML Format (pmta --xml show status):**

(Actual output will not be pretty-printed)

```

<rsp>
  <data>
    <mta>
      <product>
        <name>PowerMTA</name>
        <version>5.0</version>
        <buildDate>2015-08-31 16:07:03</buildDate>
        <revision>tags/v4_5c1@15693</revision>
        <bits>64</bits>
      </product>
      <lak>
        <serial>112345</serial>
        <instances>1</instances>
      </lak>
      <os>
        <name>Linux</name>
        <version>3.10.0-229.1.2.el7.x86_64</version>
        <build>CentOS Linux release 7.1.1503 (Core) </build>
      </os>
      <cpu>
        <type>x86_64</type>
        <count>12</count>
      </cpu>
      <ram>
        <real>32750628</real>
      </ram>
      <fullHostName>example.port25.com</fullHostName>
      <userString/>
      <hostId>0</hostId>
    </mta>
    <status>
      <timeNow>2015-09-01 10:17:07</timeNow>
      <startupTime>2015-08-31 16:22:01</startupTime>
      <shuttingDown>0</shuttingDown>
      <traffic>
        <total>
          <out>
            <rcp>5317436</rcp>
            <msg>5317436</msg>
            <kb>2788394.0</kb>
          </out>
          <in>
            <rcp>5104905</rcp>
            <msg>128951</msg>
            <kb>1951.8</kb>
          </in>
          <bounceProcessed>0</bounceProcessed>
          <feedbackLoopProcessed>0</feedbackLoopProcessed>
        </total>
        <lastHr>
          <out>
            <rcp>295376</rcp>
            <msg>295376</msg>
            <kb>97626.3</kb>
          </out>
          <in>
            <rcp>295358</rcp>
            <msg>7366</msg>
            <kb>111.4</kb>
          </in>
          <bounceProcessed>0</bounceProcessed>
          <feedbackLoopProcessed>0</feedbackLoopProcessed>
        </lastHr>
        <lastMin>
          <out>
            <rcp>4776</rcp>
            <msg>4776</msg>
            <kb>1669.2</kb>
          </out>
          <in>
            <rcp>4772</rcp>
            <msg>124</msg>
            <kb>1.9</kb>
          </in>
          <bounceProcessed>0</bounceProcessed>
          <feedbackLoopProcessed>0</feedbackLoopProcessed>
        </lastMin>
        <topPerHr>
          <out>
            <rcp>450637</rcp>
            <msg>450637</msg>
            <kb>1176613.1</kb>
          </out>
          <in>
            <rcp>303999</rcp>
            <msg>7591</msg>
            <kb>114.9</kb>
          </in>
          <bounceProcessed>0</bounceProcessed>
          <feedbackLoopProcessed>0</feedbackLoopProcessed>
        </topPerHr>
        <topPerMin>
          <out>
            <rcp>213222</rcp>
            <msg>213222</msg>
            <kb>1098011.2</kb>
          </out>
          <in>
            <rcp>8365</rcp>
            <msg>207</msg>
            <kb>3.1</kb>
          </in>

```

```

        <bounceProcessed>0</bounceProcessed>
        <feedbackLoopProcessed>0</feedbackLoopProcessed>
    </topPerMin>
    <perf>
        <lastMin>
            <in>
                <count>0</count>
                <mailFromTime>0</mailFromTime>
                <dataTime>0</dataTime>
                <totalTime>0</totalTime>
            </in>
        </lastMin>
    </perf>
</traffic>
<conn>
    <smtpIn>
        <cur>0</cur>
        <max>150</max>
        <top>1</top>
    </smtpIn>
    <smtpOut>
        <cur>0</cur>
        <max>500</max>
        <top>299</top>
    </smtpOut>
</conn>
<resolver>
    <namesCached>0</namesCached>
    <queriesPending>0</queriesPending>
    <queriesActive>0</queriesActive>
</resolver>
<queue>
    <smtp>
        <rcp>692</rcp>
        <dom>241</dom>
        <kb>10.4</kb>
    </smtp>
    <pipe>
        <rcp>0</rcp>
        <dom>0</dom>
        <kb>0.0</kb>
    </pipe>
    <discard>
        <rcp>0</rcp>
        <dom>0</dom>
        <kb>0.0</kb>
    </discard>
    <file>
        <rcp>0</rcp>
        <dom>0</dom>
        <kb>0.0</kb>
    </file>
    <alias>
        <rcp>0</rcp>
        <dom>0</dom>
        <kb>0.0</kb>
    </alias>
    <bounceProcessor>
        <rcp>0</rcp>
        <dom>0</dom>
        <kb>0.0</kb>
    </bounceProcessor>
    <feedbackLoopProcessor>
        <rcp>0</rcp>
        <dom>0</dom>
        <kb>0.0</kb>
    </feedbackLoopProcessor>
</queue>
<spool>
    <initPct>100</initPct>
    <dirs>1</dirs>
    <files>
        <inUse>491</inUse>
        <recycled>0</recycled>
        <total>491</total>
    </files>
    <totalRcp>692</totalRcp>
    <maxRcp>1000000</maxRcp>
</spool>
<status>running</status>
</status>
</data>
<succ/>
</rsp>

```

DOM-Style (pmta --dom show status):

```

mta.product.name="PowerMTA"
mta.product.version="5.0"
mta.product.buildDate="2015-08-31 16:07:03"
mta.product.revision="tags/v4_5@15693"
mta.product.bits="64"
mta.lak.serial="112345"
mta.lak.instances="1"
mta.os.name="Linux"
mta.os.version="3.10.0-229.1.2.el7.x86_64"
mta.os.build="CentOS Linux release 7.1.1503 (Core) "
mta.cpu.type="x86_64"
mta.cpu.count="12"
mta.ram.real="32750628"
mta.fullHostName="example.port25.com"
mta.userString=""

```

```

mta.hostId="0"
status.timeNow="2015-09-01 10:17:50"
status.startupTime="2015-08-31 16:22:01"
status.shuttingDown="0"
status.traffic.total.out.rcp="5319661"
status.traffic.total.out.msg="5319661"
status.traffic.total.out.kb="2789142.9"
status.traffic.total.in.rcp="5107093"
status.traffic.total.in.msg="129008"
status.traffic.total.in.kb="1952.7"
status.traffic.total.bounceProcessed="0"
status.traffic.total.feedbackLoopProcessed="0"
status.traffic.lastHr.out.rcp="294091"
status.traffic.lastHr.out.msg="294091"
status.traffic.lastHr.out.kb="97231.2"
status.traffic.lastHr.in.rcp="294046"
status.traffic.lastHr.in.msg="7335"
status.traffic.lastHr.in.kb="111.0"
status.traffic.lastHr.bounceProcessed="0"
status.traffic.lastHr.feedbackLoopProcessed="0"
status.traffic.lastMin.out.rcp="3275"
status.traffic.lastMin.out.msg="3275"
status.traffic.lastMin.out.kb="1092.6"
status.traffic.lastMin.in.rcp="3242"
status.traffic.lastMin.in.msg="85"
status.traffic.lastMin.in.kb="1.3"
status.traffic.lastMin.bounceProcessed="0"
status.traffic.lastMin.feedbackLoopProcessed="0"
status.traffic.topPerHr.out.rcp="450637"
status.traffic.topPerHr.out.msg="450637"
status.traffic.topPerHr.out.kb="1176613.1"
status.traffic.topPerHr.in.rcp="303999"
status.traffic.topPerHr.in.msg="7591"
status.traffic.topPerHr.in.kb="114.9"
status.traffic.topPerHr.bounceProcessed="0"
status.traffic.topPerHr.feedbackLoopProcessed="0"
status.traffic.topPerMin.out.rcp="213222"
status.traffic.topPerMin.out.msg="213222"
status.traffic.topPerMin.out.kb="1098011.2"
status.traffic.topPerMin.in.rcp="8365"
status.traffic.topPerMin.in.msg="207"
status.traffic.topPerMin.in.kb="3.1"
status.traffic.topPerMin.bounceProcessed="0"
status.traffic.topPerMin.feedbackLoopProcessed="0"
status.traffic.perf.lastMin.in.count="0"
status.traffic.perf.lastMin.in.mailFromTime="0"
status.traffic.perf.lastMin.in.dataTime="0"
status.traffic.perf.lastMin.in.totalTime="0"
status.conn.smtpIn.cur="0"
status.conn.smtpIn.max="150"
status.conn.smtpIn.top="1"
status.conn.smtpOut.cur="0"
status.conn.smtpOut.max="500"
status.conn.smtpOut.top="299"
status.resolver.namesCached="0"
status.resolver.queriesPending="0"
status.resolver.queriesActive="0"
status.queue.smtp.rcp="655"
status.queue.smtp.dom="243"
status.queue.smtp.kb="9.9"
status.queue.pipe.rcp="0"
status.queue.pipe.dom="0"
status.queue.pipe.kb="0.0"
status.queue.discard.rcp="0"
status.queue.discard.dom="0"
status.queue.discard.kb="0.0"
status.queue.file.rcp="0"
status.queue.file.dom="0"
status.queue.file.kb="0.0"
status.queue.alias.rcp="0"
status.queue.alias.dom="0"
status.queue.alias.kb="0.0"
status.queue.bounceProcessor.rcp="0"
status.queue.bounceProcessor.dom="0"
status.queue.bounceProcessor.kb="0.0"
status.queue.feedbackLoopProcessor.rcp="0"
status.queue.feedbackLoopProcessor.dom="0"
status.queue.feedbackLoopProcessor.kb="0.0"
status.spool.initPct="100"
status.spool.dirs="1"
status.spool.files.inUse="476"
status.spool.files.recycled="0"
status.spool.files.total="476"
status.spool.totalRcp="655"
status.spool.maxRcp="1000000"
status.status="running"

```

JSON-Style (pmta --json show status): (Actual output will not be pretty-printed)

```

{
  "data": {
    "mta": {
      "product": {
        "name": "PowerMTA",
        "version": "5.0b1",
        "buildDate": "2018-12-17 04:13:56",
        "revision": "trunk@18274",
        "bits": 64
      },
      "lak": {
        "serial": "123456",
        "instances": 0
      }
    }
  }
}

```

```
    },
    "os": {
      "name": "Linux",
      "version": "3.10.0-862.3.2.el7.x86_64",
      "build": "CentOS Linux release 7.5.1804 (Core) "
    },
    "cpu": {
      "type": "x86_64",
      "count": 2
    },
    "ram": {
      "real": 3879568
    },
    "fullHostName": "foo.int.port25.com",
    "userString": "",
    "hostId": 0
  },
  "status": {
    "timeNow": "2019-02-19 23:00:46",
    "startupTime": "2019-02-19 17:15:07",
    "shuttingDown": false,
    "traffic": {
      "total": {
        "out": {
          "rcp": 2007007,
          "msg": 2007007,
          "kb": 1370100.3
        },
        "in": {
          "rcp": 2007845,
          "msg": 28046,
          "kb": 8799.7
        },
        "bounceProcessed": 0,
        "feedbackLoopProcessed": 0
      },
      "lastHr": {
        "out": {
          "rcp": 348704,
          "msg": 348704,
          "kb": 142245.3
        },
        "in": {
          "rcp": 348700,
          "msg": 4936,
          "kb": 33.7
        },
        "bounceProcessed": 0,
        "feedbackLoopProcessed": 0
      },
      "lastMin": {
        "out": {
          "rcp": 5809,
          "msg": 5809,
          "kb": 2372.6
        },
        "in": {
          "rcp": 5800,
          "msg": 81,
          "kb": 0.6
        },
        "bounceProcessed": 0,
        "feedbackLoopProcessed": 0
      },
      "topPerHr": {
        "out": {
          "rcp": 348947,
          "msg": 348947,
          "kb": 693034.3
        },
        "in": {
          "rcp": 348700,
          "msg": 4936,
          "kb": 8633.6
        },
        "bounceProcessed": 0,
        "feedbackLoopProcessed": 0
      },
      "topPerMin": {
        "out": {
          "rcp": 6219,
          "msg": 6219,
          "kb": 214286.9
        },
        "in": {
          "rcp": 5900,
          "msg": 99,
          "kb": 3480.8
        },
        "bounceProcessed": 0,
        "feedbackLoopProcessed": 0
      },
      "perf": {
        "lastMin": {
          "in": {
            "count": 0,
            "mailFromTime": 0,
            "dataTime": 0,
            "totalTime": 0
          }
        }
      }
    }
  }
}
```

```

    },
    "conn": {
      "smtpIn": {
        "cur": 1,
        "max": 5,
        "top": 1
      },
      "smtpOut": {
        "cur": 0,
        "max": 1200,
        "top": 157
      }
    },
    "resolver": {
      "namesCached": 589,
      "queriesPending": 0,
      "queriesActive": 0
    },
    "queue": {
      "smtp": {
        "rcp": 805,
        "dom": 226,
        "kb": 5.5
      },
      "pipe": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
      },
      "discard": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
      },
      "file": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
      },
      "alias": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
      },
      "bounceProcessor": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
      },
      "feedbackLoopProcessor": {
        "rcp": 0,
        "dom": 0,
        "kb": 0
      }
    },
    "spool": {
      "initPct": 100,
      "dirs": 1,
      "files": {
        "inUse": 504,
        "recycled": 0,
        "total": 504
      },
      "totalRcp": 805,
      "maxRcp": 25000
    },
    "status": "running"
  }
},
"status": "success",
"message": ""
}

```

## 6.3 Command Reference

### 6.3.1 clear dnscache

This command clears PowerMTA's internal DNS cache, removing all entries for the given domain. If the domain is \* all cached DNS data is discarded. On Unix the asterisk must be escaped with a backslash.

#### Syntax:

```
pmta clear dnscache domain
```

#### Example:

```

$ pmta clear dns *
DNS cache cleared.

$ pmta clear dns port25.com
Cleared port25.com from DNS cache.

```

### 6.3.2 check mfrom

This command has been deprecated because the Sender-ID standard checking performed by this command is obsolete. Please use check spf instead.

### 6.3.3 check spf

This command checks whether a message with the given sender would pass the SPF check if sent from the given IP address. If the `mailFrom` parameter is specified, it will be used as the identity to check, else the `heloDomain` identity is checked.

**Syntax:**

```
pmta check spf [--tcp] [--dumppackets] ip heloDomain [mailFrom]
```

**Options:**

**--tcp**

use TCP (rather than the default UDP) for querying the name server(s).

**--dumppackets**

print out a hexadecimal dump of any DNS queries and responses exchanged with the name servers. It is ignored unless the output is in text format.

---

### 6.3.4 reload

PowerMTA reads its configuration during startup. Use this command to activate the new settings after making changes to the configuration file. The `reload` command is generally preferable to stopping and restarting PowerMTA because it is more efficient and does not cause an interruption in service.

If PowerMTA detects an error in the configuration file, the reload command fails and the previous in-memory configuration is retained.

**Syntax:**

```
pmta reload
```

**Example:**

```
$ pmta reload
Configuration reloaded.
```

---

### 6.3.5 reset counters

The `reset counters` command allows you to reset the various counters for statistics shown by PowerMTA in the output of `show status` command without having to restart the PowerMTA service.

**Syntax:**

```
pmta reset counters
```

**Example:**

```
$ pmta reset count
Traffic counters reset.
```

---

### 6.3.6 reset status

This command is just an alias to `reset counters`.

---

### 6.3.7 resolve

The `resolve` command queries DNS for mail routing information (MX and A/AAAA records) for the specified domain, displays that information, and optionally connects to the first available mailhost. It is intended to facilitate debugging connectivity problems to destination domains. For example, if you notice that the queue is building up for a specific domain, you can run `resolve` against that domain to figure out what problems exist and what if anything can be done to fix the problem.

**Syntax:**

```
pmta resolve [--tcp] [--connect] [--use-dane] [--dumppackets] [--interactive] [--server=ip] [--source=
[host,]ip] domainname
```

**Defaults:**

If no options are specified, DNS resolution is attempted via UDP and any relevant information is displayed, however no attempt is made to connect to a mailhost.

**Options:**

**--tcp**

this tells PowerMTA to use TCP (rather than the default UDP) for querying the name server(s)

**--connect**

this option tells PowerMTA to attempt to connect to the first available mailhost as well, after querying and displaying the specified domain's DNS server for mail routing information.

**--dumppackets**

this option tells PowerMTA to print out a hexadecimal dump of any DNS queries and responses exchanged with the name servers. It is ignored unless the output is in textual format.

**--interactive**

this option allows the user to try various SMTP commands, to help diagnose problems.

**--server**

this option tells PowerMTA which DNS server to connect to when resolving hostnames.

**--source**

this option tells PowerMTA the internal IP address or hostname to use during the connection attempt.

**--use-dane**

this option tells PowerMTA to lookup DANE DNS records for the connection attempt, and use STARTTLS during the connection attempt based on the remote site's preference as listed in its DANE DNS records. Note that PowerMTA has to be configured with a DNS resolver that supports DNSSEC for this flag to be used. See [DANE](#) for more information.

#### Examples:

```
$ pmta resolve --connect outlook.com
Querying 127.0.0.1 over UDP about MX outlook.com
Read response from 127.0.0.1
answers:      ttl=60
             outlook.com. 60 IN MX 5 outlook-com.olc.protection.outlook.com.

Querying 127.0.0.1 over UDP about A outlook-com.olc.protection.outlook.com
Read response from 127.0.0.1
answers:      ttl=2
             outlook-com.olc.protection.outlook.com. 2 IN A 104.47.48.33
             outlook-com.olc.protection.outlook.com. 2 IN A 104.47.49.33

status = StatusOk
pref host name                               IP addresses
-----
5 outlook-com.olc.protection.outlook.com 104.47.48.33 104.47.49.33

connecting from localhost (0.0.0.0) to outlook-com.olc.protection.outlook.com (104.47.48.33)
>>> 220 C01NAM05FT041.mail.protection.outlook.com Microsoft ESMTP MAIL Service ready at Thu, 7 Feb 2019
15:18:50 +0000
<<< EHLO localhost
>>> 250-C01NAM05FT041.mail.protection.outlook.com Hello
>>> 250-SIZE 49283072
>>> 250-PIPELINING
>>> 250-DSN
>>> 250-ENHANCEDSTATUSCODES
>>> 250-STARTTLS
>>> 250-8BITMIME
>>> 250-BINARYMIME
>>> 250-CHUNKING
>>> 250 SMTPUTF8
<<< STARTTLS
>>> 220 2.0.0 SMTP server ready
TLSv1.2 connected with 256-bit ECDHE-RSA-AES256-SHA384
Cert: /C=US/ST=Washington/L=Redmond/O=Microsoft Corporation/CN=mail.protection.outlook.com;
issuer=/C=BE/O=GlobalSign nv-sa/CN=GlobalSign Organization Validation CA - SHA256 - G3; verified=no
<<< EHLO localhost
>>> 250-C01NAM05FT041.mail.protection.outlook.com Hello
>>> 250-SIZE 49283072
>>> 250-PIPELINING
>>> 250-DSN
>>> 250-ENHANCEDSTATUSCODES
>>> 250-8BITMIME
>>> 250-BINARYMIME
>>> 250-CHUNKING
>>> 250 SMTPUTF8
<<< QUIT
>>> 221 2.0.0 Service closing transmission channel
closed outlook-com.olc.protection.outlook.com (104.47.48.33) in=601 out=48

$ pmta --dom resolve yahoo.com
status="StatusOk"
host[0].pref="1"
host[0].name="mta5.am0.yahoodns.net"
host[0].ip[0]="98.137.159.25"
host[0].ip[1]="98.137.159.27"
host[0].ip[2]="98.137.159.28"
host[0].ip[3]="66.218.85.52"
host[0].ip[4]="67.195.228.141"
host[0].ip[5]="67.195.229.58"
host[0].ip[6]="67.195.229.59"
host[0].ip[7]="98.136.102.54"
host[1].pref="1"
host[1].name="mta6.am0.yahoodns.net"
host[1].ip[0]="74.6.137.63"
host[1].ip[1]="74.6.137.64"
host[1].ip[2]="74.6.137.65"
host[1].ip[3]="98.137.159.26"
host[1].ip[4]="98.137.159.28"
host[1].ip[5]="66.218.85.52"
host[1].ip[6]="67.195.228.141"
host[1].ip[7]="67.195.229.58"
host[2].pref="1"
host[2].name="mta7.am0.yahoodns.net"
host[2].ip[0]="67.195.229.58"
host[2].ip[1]="67.195.229.59"
host[2].ip[2]="98.136.101.117"
host[2].ip[3]="98.136.102.54"
host[2].ip[4]="98.137.159.24"
host[2].ip[5]="98.137.159.27"
host[2].ip[6]="98.137.159.28"
host[2].ip[7]="66.218.85.139"
```

### 6.3.8 rotate acct

This command instructs PowerMTA to immediately rotate all configured accounting files. If a filename is specified in the command, only that specified accounting file will be rotated. Immediate rotation does not affect the automatic rotation at midnight, which is still performed.

#### Syntax:

```
pmta rotate acct [file]
```

**Example:**

```
$ pmta rotate acct
Accounting file rotated.
```

**6.3.9 rotate log**

This command instructs PowerMTA to immediately rotate the logging file. Immediate rotation does not affect the automatic rotation at midnight, which is still performed.

**Syntax:**

```
pmta rotate log
```

**Example:**

```
$ pmta rotate log
Logging file rotated.
```

**6.3.10 schedule**

Instructs PowerMTA to immediately request workers to be assigned to the specified queue(s) in order to start delivery attempts for messages queued in them. This command would be used, for example, if connectivity problems to a domain were fixed and the normal retry interval should be bypassed. A wildcard can also be specified in *domain* or *vmta*, allowing multiple domains/vmtas to be scheduled at once.

**Syntax:**

```
pmta schedule [--retry-recipients] domain/vmta
```

**Options:****--retry-recipients**

Recipients that received SMTP 4xx errors (such as "mailbox full" error) are placed in a separate retry bucket where they are held for up to a period of time controlled by the per-domain setting [retry-recipients-after](#). This flag overrides the `retry-after` period for recipients, and immediately makes them available for delivery attempts.

**Example:**

```
$ pmta schedule */*
154 of 3888 matching queues scheduled.
```

**6.3.11 set queue**

Sets the given queue's mode of operation to either normal or backoff. While in backoff mode, various different settings may apply. See the `backoff-...` directives in [<domain> Directives](#) for more information.

Wildcards can be used to modify various queues at the same time.

**Syntax:**

```
pmta set queue --mode={normal|backoff} domain/vmta
```

**6.3.12 show domains**

This command allows you to view current information on the specified destination domain(s).

**Syntax:**

```
show domains [--vmta=name] [--connected={yes|no}] [--maxitems=n] [--errors] [--sort={name|rcpt|size}]
[name]
```

**Options:****--vmta=name**

displays only domains information for the given VirtualMTA. The default is to display (accumulated) information for all VirtualMTAs, including messages for which no VirtualMTA was selected.

**--connected**

displays domains to which PowerMTA has least one open outgoing connection. The default is to display both connected and unconnected domains.

**--maxitems=n**

specifies the maximum number of domains to be displayed in the output. By default, 20 domains are displayed.

**--errors**

displays any errors recorded for the domain.

**--sort=key**

sort output by *key* where *key* may be *name* to order the domain list alphabetically by name, *rcpt* to order the list by number of recipients or *size* to order the list by volume (kbytes).

**name**

domain name or wildcard specifying the domains about which information is to be displayed. If omitted, all domains are displayed (subject to the `maxitems` limit).

**Examples:**

```
$ pmta show domains
-----domain --#rcpt ---kbytes conn last error-----
test1.port25.com      7      30.6      0
```

|                   |    |      |                        |
|-------------------|----|------|------------------------|
| test10.port25.com | 8  | 30.6 | 0                      |
| test11.port25.com | 3  | 30.6 | 0                      |
| test12.port25.com | 13 | 30.6 | 0                      |
| test13.port25.com | 3  | 30.6 | 0                      |
| test14.port25.com | 3  | 30.6 | 0                      |
| test15.port25.com | 3  | 30.6 | 0                      |
| test16.port25.com | 52 | 30.6 | 0                      |
| test17.port25.com | 3  | 30.6 | 0                      |
| test18.port25.com | 3  | 30.6 | 0 ETIMEDOUT connect... |
| test19.port25.com | 3  | 30.6 | 0                      |
| test2.port25.com  | 3  | 30.6 | 0                      |
| test20.port25.com | 3  | 30.6 | 0                      |
| test21.port25.com | 3  | 30.6 | 0                      |
| test22.port25.com | 3  | 30.6 | 0                      |
| test23.port25.com | 3  | 30.6 | 0                      |
| test24.port25.com | 3  | 30.6 | 0                      |
| test25.port25.com | 3  | 30.6 | 0                      |
| test26.port25.com | 3  | 30.6 | 0                      |
| test27.port25.com | 3  | 30.6 | 0                      |

20 of 30 domains shown.

### 6.3.13 show queues

This command allows you to view current information on the specified queues. The output also includes the last 50 delivery/bounce/deferral events by default.

#### Syntax:

```
show queues [--connected={yes|no}] [--paused={yes|no}] [--mode-events] [--mode={normal|backoff}] [--maxitems=n] [--errors] [--sort={name|rcpt|size}] [--no-rcpt-events] [queue]
```

#### Options:

##### --connected

displays queues for which PowerMTA has at least one open outgoing connection. The default is to display both connected and unconnected queues.

##### --mode

displays only queues in the specified mode. The default is to display queues in all modes.

##### --maxitems=n

specifies the maximum number of queues to be displayed in the output. By default, 20 domains are displayed.

##### --errors

displays any errors recorded for the queue.

##### --sort=key

sort output by *key* where *key* may be name to order the queue list alphabetically by name, rcpt to order the list by number of recipients or size to order the list by volume (kbytes).

##### --no-rcpt-events

hides the delivery events, that by default are shown.

##### --paused={yes|no}

allows selecting only paused or running queues for display. By default, both are shown.

##### --mode-events

allows show the recent mode changes

##### queue

name of the queue(s) to display, in the format *domain[/vmta]*, where *domain* is the destination domain name and *vmta* is the VirtualMTA. If omitted, both *domain* and *vmta* default to *\** (i.e., *\*/\**), which matches all queues.

#### Examples:

```
$ pmta show queue
queue      #rcpt kbytes conn st    retry last error
-----
*/ *
example.com/vmta1  2    0.2    0    09:35:42

Recipient Events (deliveries:4, bounces:1, deferrals:1)
-----
delivered | 2014-11-24 09:35:09 | 10.25.25.235 | kf@example.com | jobid10145783566176 | [192.168.10.48]
(192.168.10.48) | smtp;250 DATA ok
delivered | 2014-11-24 09:31:07 | 10.25.25.235 | ft@example.com | jobid10145783566176 | [192.168.10.48]
(192.168.10.48) | smtp;250 DATA ok
deferred  | 2014-11-24 09:29:06 | 10.25.25.235 | np@example.com | jobid8363372138340 | [192.168.10.48]
(192.168.10.48) | smtp;450 4.0.0 (undefined status);smtp;421 Refused. Your reverse DNS entry does not
resolve.
delivered | 2014-11-24 09:26:06 | 10.25.25.235 | ce@example.com | jobid8363372138340 | [192.168.10.48]
(192.168.10.48) | smtp;250 DATA ok
delivered | 2014-11-24 09:26:06 | 10.25.25.235 | rd@example.com | jobid10145783566176 | [192.168.10.48]
(192.168.10.48) | smtp;250 DATA ok
bounced | 2014-11-24 09:36:09 | 10.25.25.235 | vc@example.com | jobid8363372138340 | [192.168.10.48]
(192.168.10.48) | smtp;550 5.0.0 (undefined status);smtp;550 No Such User Here3 of 3 matching queues.
```

### 6.3.14 show jobs

This command allows you to view the jobs currently in PowerMTA's queue.

#### Syntax:

```
show jobs [--maxitems=n]
```

#### Options:

**--maxitems=n**

specifies the maximum number of queues to be displayed in the output. By default all jobs are displayed.

#### Example:

```
$ pmta show jobs
-----job-id ---#rcpt ---kbytes
               weeklyNews22100      431    1236.9

Total of 1 job(s).
```

### 6.3.15 show settings

This command allows you to view the (per-queue) settings for the specified domain and VirtualMTA.

#### Syntax:

```
show settings domain[/vmta]
```

#### Example:

```
$ pmta show settings port25.com/vmta1
Virtual MTA Settings
-----
max-smtp-msg-rate unlimited
max-smtp-out unlimited

Queue Settings
-----
type smtp
add-x-msys-api-header no
allow-cancel-during-transfer yes
allow-priority-interruption yes
allow-priority-interruption-during-transfer no
assume-delivery-upon-data-termination-timeout no
auth-password ""
auth-username ""
backoff-max-connect-rate unlimited
backoff-max-msg-rate unlimited
backoff-max-smtp-out 20
backoff-notify ""
backoff-reroute-to-virtual-mta ""
backoff-retry-after 10m
backoff-to-normal-after never
backoff-to-normal-after-delivery no
backoff-upon-all-sources-disabled no
bcc-upon-delivery ""
bounce-after 4d12h
bounce-upon-5xx-greeting yes
bounce-upon-no-mx no
bounce-upon-pix-transfer-failure no
bounce-upon-transfer-failure no
check-spf-outbound no
connect-timeout 2m
data-send-timeout 3m
default-smtp-port 25
deliver-local-dsn yes
disable-acct-records ""
disable-mx-rollup no
disabled-source-ip-notify ""
dkim-add-body-limit no
dkim-add-timestamp no
dkim-algorithm rsa-sha256
dkim-body-canon relaxed
dkim-disallow-adding-headers ""
dkim-expire-after never
dkim-headers ""
dkim-headers-canon relaxed
dkim-identity sender-or-from
dkim-query-method ""
dkim-sign yes
ignore-8bitmime no
ignore-chunking yes
ignore-dsn no
ignore-pipelining no
log-commands no
log-connections no
log-data no
log-disabled-ips no
log-resolution yes
log-transfer-failures no
max-cold-virtual-mta-msg 0/d
max-connect-rate unlimited
max-errors-per-connection unlimited
max-events-recorded 10
max-msg-per-connection 20
max-msg-rate unlimited
max-rcpt-per-message 1000
max-smtp-out 20
mx-connection-attempts 10
notify-of-delay-every 4h
pix-bug-workaround yes
queue-priority 50
reenable-source-ip-after never
remove-header ""
```

```

replace-smtp-421-service-response no
require-starttls no
retry-after 10m
retry-after-dns-error 1m
retry-recipients-after 600
retry-upon-new-mail no
reuse-ssl-session no
smtp-421-means-mx-unavailable no
smtp-553-means-invalid-mailbox yes
smtp-client-certificate ""
smtp-data-termination-timeout 10m
smtp-greeting-timeout 5m
smtp-hosts ""
smtp-multiple-line-response-timeout 5m
smtp-pattern-list ""
smtp-tls-allow-SSLv3 no
smtp-tls-allow-TLSv1 yes
smtp-tls-ca-file ""
sparkpost-ip-pool ""
sparkpost-sandbox no
track-recipient-events no
use-starttls yes
use-unencrypted-plain-auth no

```

### 6.3.16 show status

Displays various statistics and counters since last restart. The output includes totals, top throughput and queue information among other things.

**Syntax:**

```
pmta show status
```

**Example:**

```

$ pmta show status
PowerMTA v5.0 status on example.port25.com on 2018-09-10 10:16:19

Traffic      -----inbound-----      -----outbound-----
      rcpts      msgs      kbytes      rcpts      msgs      kbytes
Total      939437      5940      233409.5      939430      939430      34550887.2
Last Hour      939437      5940      233409.5      939263      939263      34544775.4
Top/Hour      939437      5940      233409.5      939263      939263      34544775.4
Last Min.      45063      298      11632.6      45021      45021      1658056.5
Top/Min.      69881      458      18005.6      69115      69115      2489680.7

Connections  active      top      maximum      Domain      cached      pending
Inbound      0      1      5      Names      2      0
Outbound      10000      10000      10000

Queues      rcpts      domains      kbytes      Spool      in use      recycled
SMTP      99459      2895      3985272.8      Files      5667      0
other      0      0      0.0      Init.      complete

Status      running      Started      2018-09-10 20:02:12      Uptime      0 0:20:01

```

### 6.3.17 show topdomains

For displaying the top domains in the queue sorted by number of recipients. Since this command is equivalent to `show domains --sort=rcpt`, please see [show-domains](#) for more information on the various options.

**Syntax:**

```
pmta show topdomains [--vmta=name] [--connected={yes|no}] [--maxitems=n] [--errors]
```

### 6.3.18 show topqueues

For displaying the top queues sorted by number of recipients. Since this command is equivalent to `show queues --sort=rcpt`, please see [show-queues](#) for more information on the various options.

**Syntax:**

```

show topqueues [--connected={yes|no}] [--mode={normal|backoff}] [--maxitems=n] [--errors] [--paused=
{yes|no}] [--no-rcpt-events] [domain[/vmta]]

```

### 6.3.19 show version

Displays PowerMTA's version and build date.

**Syntax:**

```
pmta show version
```

**Example:**

```

$ pmta show ver
PowerMTA v5.0r1 (2019-09-13 16:27:31, 64-bit; master@190913.ba26ec5).

```

### 6.3.20 show vmtas

This command displays a summary of the VirtualMTA(s) in use, as well as the number of recipients and message size in kilobytes within the VirtualMTA.

**Syntax:**

```
show vmtas [--maxitems=n]
```

#### Options:

##### --maxitems=n

specifies the maximum number of queues to be displayed in the output. By default all vmtas are displayed.

#### Example:

```
name      domains #rcpt % total kbytes conn
-----
vmta572    13      18      2    0.1  16
vmta103     1       8      1    0.1   8
vmta118     2       8      1    0.1   5
vmta808     2       8      1    0.1   2
vmta261     7       8      1    0.1   3
vmta188     1       7      1    0.0   0
vmta863     4       6      0    0.0   5

7 of 7 virtual MTAs.
```

### 6.3.21 trace

Instructs PowerMTA to open a connection to the specified domain and to attempt delivery of a single message from the queue. This command is similar to the `schedule` command, with the difference that it also automatically enables logging for the domain for the duration of one connection. If the `--to-log` switch is used, it is a shortcut to enabling logging in the configuration file, reloading the configuration, scheduling a call and undoing these changes when debugging is finished. By default, only the SMTP commands and responses are logged. The result of this command is written to the logging file.

#### Syntax:

```
pmta trace [--log-data] [--log-resolution] [--to-log] [--retry-recipients] [--source-ip=ip] domain/vmta
```

#### Options:

##### --log-data

this enables logging of data transfers as well, like the `log-data` option in the configuration file.

##### --log-resolution

this enables logging of the DNS resolution as well, like the `log-resolution` option in the configuration file.

##### --to-log

this enables logging trace info to the log file

##### --retry-recipients

Use of the `retry-recipients` flag will cause all the recipients of the queue that are in Await Retry bucket to become available for retry immediately.

##### --source-ip

Specifies the source IP used for connection attempts from a virtual mta with multiple `smtp-source-hosts` configured in it

#### domain[/vmta]

name of the queue(s) to display, where *domain* is the destination domain name and *vmta* is the VirtualMTA. If omitted, both *domain* and *vmta* default to `*` (i.e., `*/*`), which matches all queues.

#### Example:

```
$ pmta trace yahoo.com
2019-02-07 22:31:24 starting port25.com/{default}
2019-02-07 22:31:24 connecting from foo.port25.com (::) to ASPMX.L.GOOGLE.com (2607:f8b0:400e:c07::1a)
2019-02-07 22:31:24 connected from [2600:1a17:a893:9b59:dac9:5dr1:de28:5324]:39339
2019-02-07 22:31:24 >>> 220 mx.google.com ESMTP u2si198147pgp.592 - gsmt
2019-02-07 22:31:24 <<< EHLO foo.port25.com
2019-02-07 22:31:24 >>> 250-mx.google.com at your service
2019-02-07 22:31:24 >>> 250-SIZE 157286400
2019-02-07 22:31:24 >>> 250-8BITIME
2019-02-07 22:31:24 >>> 250-STARTTLS
2019-02-07 22:31:24 >>> 250-ENHANCEDSTATUSCODES
2019-02-07 22:31:24 >>> 250-PIPELINING
2019-02-07 22:31:24 >>> 250-CHUNKING
2019-02-07 22:31:24 >>> 250 SMTPUTF8
2019-02-07 22:31:24 <<< STARTTLS
2019-02-07 22:31:24 >>> 220 2.0.0 Ready to start TLS
2019-02-07 22:31:24 tls:TLSv1.2 connected with 128-bit ECDHE-RSA-AES128-GCM-SHA256
2019-02-07 22:31:24 tls:Cert: /C=US/ST=California/L=Mountain View/O=Google LLC/CN=mx.google.com;
issuer=/C=US/O=Google Trust Services/CN=Google Internet Authority G3; verified=no
2019-02-07 22:31:24 <<< EHLO foo.port25.com
2019-02-07 22:31:24 >>> 250-mx.google.com at your service
2019-02-07 22:31:24 >>> 250-SIZE 157286400
2019-02-07 22:31:24 >>> 250-8BITIME
2019-02-07 22:31:24 >>> 250-ENHANCEDSTATUSCODES
2019-02-07 22:31:24 >>> 250-PIPELINING
2019-02-07 22:31:24 >>> 250-CHUNKING
2019-02-07 22:31:24 >>> 250 SMTPUTF8
2019-02-07 22:31:24 <<< QUIT
2019-02-07 22:31:24 >>> 221 2.0.0 closing connection u2si198147pgp.592 - gsmt
2019-02-07 22:31:24 closed ASPMX.L.GOOGLE.com (2607:f8b0:400e:c07::1a) in=511 out=68
2019-02-07 22:31:24 done port25.com/{default}
```

### 6.3.22 delete

Deletes recipients from the queue. Although `--queue's` default is `*/*`, the command requires either `--queue` or `--jobId` to be entered explicitly, so that an accidental `pmta delete` command doesn't delete the entire queue. Messages are marked in the accounting file as deleted by administrator.

**Syntax:**

```
pmta delete [--dsn] [--queue=domain[/vmta]] [--orig=addr] [--rcpt=addr] [--jobId=id] [--envId=id] [--older-than=time-interval] [--no-accounting]
```

**Options:****--dsn**

specifies that a DSN report should be sent. Using this option may slow the deletion process. No entry is written to the accounting file unless the --accounting flag is used as well.

**--queue**

specifies which queue(s) to delete from, and defaults to all queues.

**--orig**

specifies that only recipients with the given MAIL FROM address be deleted

**--rcpt**

specifies that only recipient(s) with the given address are to be deleted

**--jobId**

specifies that only recipients with the given Job ID are to be deleted

**--envId**

specifies that only recipients with the given Envelope ID are to be deleted

**--no-accounting**

specifies that deletion information will not be written to the accounting file

**--older-than**

deletes all the recipients in the given queue that are older than the given from the current time. The syntax for the time interval is same as the syntax used to specify bounce-after in the configuration file.

**Example:**

```
$ pmta delete --queue=yahoo.com/*
```

**6.3.23 pause queue**

Pause delivery of messages for the given queue and prevents them from being delivered until they are resumed.

**Syntax:**

```
pmta pause queue domain[/vmta]
```

**Example:**

```
$ pmta pause queue */customer1
Queue(s) paused.
```

**6.3.24 pause job**

Pause delivery of messages for the given jobId, and prevents them from being delivered until they are resumed.

**Syntax:**

```
pmta pause job jobid
```

**Example:**

```
$ pmta pause job campaign1234
paused job.
```

**6.3.25 resume queue**

Resumes delivery of messages in the queue.

**Syntax:**

```
pmta resume queue domain[/vmta]
```

**Example:**

```
$ pmta resume queue */customer1
Queue(s) resumed.
```

**6.3.26 resume job**

Resumes delivery of messages matching the given jobId.

**Syntax:**

```
pmta resume job jobid
```

**Example:**

```
$ pmta resume job campaign1234
resumed job.
```

### 6.3.27 list

Lists recipients matching the provided options.

#### Syntax:

```
pmta list [--queue=domain[/vmta]] [--orig=addr] [--rcpt=addr] [--jobId=id] [--envId=id] [--maxitems=n] [--older-than=time-interval] [--pause] [--priority] [--schedule]
```

#### Options:

##### --queue

specifies which queue(s) to list from, and defaults to all queues.

##### --orig

specifies that only recipients with the given MAIL FROM address be listed

##### --rcpt

specifies that only recipients with the given RCPT TO address are to be listed.

##### --jobId

specifies that only recipients with the given Job ID are to be listed

##### --envId

specifies that only recipients with the given Envelope ID are to be listed.

##### --maxitems=n

specifies the maximum number of recipients to be displayed in the output. By default 100 recipients are displayed.

##### --pause

if recipient is being actively delivered, pause the delivery to enable showing the recipient in the output

##### --priority

display the priority of each recipient in the output of the command

##### --schedule

display the per-recipient schedule in the output of the command

##### --older-than=<time interval>

displays recipients that have been in the queue for longer than the specified time interval. For example, --older-than=24h will list all recipients that have been in the queue for longer than 24 hours. Other examples of valid time intervals are 6h, 2h15m, 4d12h.

#### Example:

```
$ pmta list --queue=yahoo.com/vmta1
```

### 6.3.28 show license

Allows listing license information from the command line.

#### Syntax:

```
pmta show license
```

#### Example:

```
$ pmta show license
product: PowerMTA
version: 5.0
platform: linux-intel
units: 0
instances: 1
licensee: Foo Corp
serial: 238478
issued: 2016-12-10
expires: 2100-03-20
copyright: Port25 Solutions, Inc. All Rights Reserved
```

### 6.3.29 deregister

Deregisters a node for PowerMTA Management Console. This command will delete all data that has not yet been loaded into PowerMTA Management Console. This does not effect the user configured accounting files.

#### Syntax:

```
pmta deregister [--user=name] [--local-only] [--retain-unloaded]
```

#### Options:

##### --user

specifies the PowerMTA Management Console user account to use when deregistering

##### --local-only

only deregisters locally, registration will still exist in PowerMTA Management Console.

##### --retain-unloaded

prevents deleting accounting files that have not yet been loaded from PowerMTA into PMC. Useful when needing to deregister and then reregister without losing any data.

**Example:**

```
$ pmta deregister --retain-unloaded
Deregistration successful.
You may still need to remove this PowerMTA node manually from PowerMTA Management Console.
```

**6.3.30 digest**

Calculates the hash of the given string using the specified algorithm, either HMAC-SHA256 or Siphash, and a key, provided either as a regular string using the `--key` option, or as a hexadecimal value using the `--hex-key` option.

**Syntax:**

```
pmta digest [--hmac-sha256] [--siphash] [--key=string] [--hexkey=string] string
```

**Options:****--hmac-sha256**

specifies the usage of the HMAC-SHA256 hashing algorithm. Cannot be combined with `--siphash`.

**--siphash**

specifies the usage of the Siphash hashing algorithm. A key of size 0 - 16 bytes must be provided. Keys less than 16 bytes will be end-padded with 0x00 (null-bytes) to reach 16 bytes. A key larger than 16 bytes will not be accepted. Cannot be combined with `--hmac-sha256`.

**--key=string**

specifies a regular string to be used as the key for the hashing algorithm. Cannot be combined with `--hexkey=string`.

**--hexkey=string**

specifies a hexadecimal string to be used as the key for the hashing algorithm. Cannot be combined with `--key=string`.

**Examples:**

```
$ pmta digest --hmac-sha256 --key=secretkey port25
Result: 614648A140A88192D375C22E606BEC922E8CFA71A5063C067E6BE7CB49C506AA

$ pmta digest --siphash --key=secretkey port25
Result: EB938A2508145899

$ pmta digest --hmac-sha256 --hexkey=1234abcd port25
Result: 3BB5D751EB1C3ECCD12C10B315DADEBA35C3A6DF822E44A347FC01B38001477A

$ pmta digest --siphash --hexkey=1234abcd port25
Result: D343D887797D2CFB
```

**6.3.31 register**

Registers a node with PowerMTA Management Console.

**Syntax:**

```
pmta register [--user=name] [--label=name] [--webmon-ip=ip] [--webmon-port=number]
[--pmtamc-port=number] [--reuse-label] pmtamc-hostname
```

**Options:****--user**

PowerMTA Management Console user to be used for registration

**--label**

Name of PowerMTA instance that will be shown in PowerMTA Management Console

**--webmon-ip**

IP address for PowerMTA Management Console to connect to PowerMTA instance

**--webmon-port**

Port for PowerMTA Management Console to connect to PowerMTA instance

**--pmtamc-port**

PowerMTA Management Console port to be used for registration

**--reuse-label**

If label was previously used, but deregistered, this allows label to be used again

**--pmtamc-hostname**

PowerMTA Management Console server to be used for registration

**Example:**

```
$ pmta register --label=pmta1server central.yourdomain.com
PowerMTA Management Console "admin" password:
Registration successful.
PowerMTA Management Console certificate fingerprint:
0B:1D:EC:9C:8F:20:4E:06:55:2C:A8:9B:A1:A6:05:C7:DD:07:93:01

For best security, please verify that the above value is the same as the one displayed on PowerMTA
Management Console's PowerMTA Management page.
```

**6.3.32 show registration**

Shows current PowerMTA Management Console registration information.

**Syntax:**

```
pmta show registration
```

**Example:**

```
$ pmta show registration
Currently registered with PowerMTA Management Console cental.yourdomain.com:8181 as "pmta1server"
PowerMTA Management Console certificate fingerprint:
0B:1D:EC:9C:8F:20:4E:06:55:2C:A8:9B:A1:A6:05:C7:DD:07:93:01
```

### 6.3.33 disable source

Disables the given source IP from being used for delivering email from the given queue.

**Syntax:**

```
pmta disable source [--reenable-after=interval] ip domain[/vmta]
```

**Example:**

```
$ pmta disable source --reenable-after=1h 1.2.3.4 yahoo.com/vmta1
```

### 6.3.34 enable source

Enables the given source IP to be used for delivering email from the given queue.

**Syntax:**

```
pmta enable source ip domain[/vmta]
```

**Example:**

```
$ pmta enable source 1.2.3.4 yahoo.com/vmta1
```

### 6.3.35 show disabled sources

Shows currently disabled source IPs for the given queue.

**Syntax:**

```
pmta show disabled sources [domain[/vmta]]
```

**Example:**

```
$ pmta show disabled sources yahoo.com/vmta1
```

### 6.3.36 show precache

Shows currently precached domains.

**Syntax:**

```
pmta show precache
```

**Example:**

```
$ pmta show precache
```

### 6.3.37 set priority

Sets priority for individual messages that match the given criteria.

**Syntax:**

```
pmta set priority [--queue=domain[/vmta]] [--orig=addr] [--rcpt=addr] [--jobId=id] [--envId=id] number
```

**Example:**

```
$ pmta set priority --rcpt=ceo@example.com 100
```

### 6.3.38 test config

Can be used to verify whether the syntax of the configuration file is correct without reloading PMTA's running configuration.

**Syntax:**

```
pmta test config
```

**Example:**

```
$ pmta test config
Error in line 11 of /etc/pmta/config: argument missing
```

## 7 Application Programming Interfaces

PowerMTA includes various programming interfaces (APIs) intended to facilitate the creation of email enabled applications and to optimize the way in which messages are submitted to PowerMTA. The following APIs are available:

- submission APIs, for submitting email from Perl, Java, .NET, C++, and C programs.
- a "pickup" directory, for submitting pre-formatted email by copying it into a directory.
- "pipe" delivery, for delivering email to a local program.

Use of the APIs may require the use of the [always-allow-api-submission](#) directive.

On Windows Server, any applicable API materials can be found in the api subdirectory within the folder into which PowerMTA was installed. On Linux, these can be found on `/opt/pmta/api`.

## 7.1 Submission APIs

The submission APIs allow you to programmatically submit messages for delivery by PowerMTA. Since most users already have pre-formatted messages to send, the APIs mostly concern themselves with the transport of such messages. However, they do include a Date header generator and a Base64 encoder to facilitate the jobs of those creating messages from scratch.

The submission APIs are object-oriented and comprise three objects:

- a `Connection` which represents a connection to a PowerMTA server.
- a `Message` which represents the message being submitted.
- a `Recipient` which represents the message's recipients.

The general procedure for submitting a message is pretty straightforward:

- connect to a PowerMTA server by creating a `Connection`.
- create a `Message`.
- create one or more `Recipients` and add them to the `Message`.
- add headers and body to the `Message` by invoking the `addData` method one or more times.
- submit the message by invoking the `Connection`'s `submit` method, passing the `Message` to it.

Other methods are available to set various delivery options, select automatic data encoding, etc. See the language-specific sections below for more information.

**NOTE:** The submission API doesn't have any clear advantage over the SMTP connection in terms of speed except in the cases where mailmerge is used and multiple recipients are sent per message.

### 7.1.1 Requirements

The submission APIs can be run remotely or locally, as they use a modified version of SMTP.

Applications using the submission APIs must also have the proper authorization. On Windows Server, programs must run under either the SYSTEM account or under an account belonging to the Administrators group.

On Unix, programs must either run as root or from an username who belongs to the pmta group. The pmta group is created automatically during PowerMTA's installation.

The following are required when using the Perl, .NET, or Java APIs:

- Perl 5.12 or newer when using the Perl API
- .NET 2.0 or better when using the .NET API
- Java 6 or better (for JavaMail: JavaMail and Java Activation Framework from Sun's download site as detailed in the JavaDoc from the archive) when using the Java API
- PowerMTA needs to be configured with `always-allow-relaying yes` and `allow-mailmerge yes` for the appropriate <source>. `always-allow-api-submission yes` can be used as well, as this will set `allow-mailmerge yes`. If you want to use authentication you need to define users and passwords and allow auth for the source.
- It is recommend to add no more than 5000 recipients per batch submission. In the event that the batch fails, this will affect fewer recipients. Also, this helps keep memory usage to a minimum.

### 7.1.2 Perl Submission API

To install on Linux, the Submitter package is distributed in the standard CPAN format, so you will need to unpack, compile and install as follows:

```
# cd /opt/pmta/api
# tar -xzf Port25-Submitter-1.5.tar.gz
# cd Port25-Submitter-1.5
# perl Makefile.PL
# make test
# make install
```

To install on Windows, use the following commands (possibly changing the path first). Currently only Perl 5.6 is supported:

```
ppm install --location=C:\pmta\api Port25-Submitter
ppm install --location=C:\pmta\api Port25-Accounter
```

Documentation for the perl APIs is provided in perldoc format. The APIs are available from the download site.

```
perldoc Port25::Submitter
perldoc Port25::Submitter::Connection
perldoc Port25::Submitter::Message
perldoc Port25::Submitter::Recipient
```

### 7.1.3 Java Submission API

To use the Java API, include `pmta.jar` in your classpath. If you want to use PowerMTA with Oracle's JavaMail API, install the provided `javamail.providers` file according to your system configuration. See the JavaMail documentation for details.

Documentation for the Java APIs is provided in JavaDoc format.

On Windows, it can be found in the `pmtajavadoc.zip` file in the `api\java` subdirectory of where PowerMTA was installed.

On Linux, it can be found in the `/opt/pmta/api/java/doc/javadoc` directory as `/opt/pmta/api/java/doc/javadoc/index.html`.

### 7.1.4 C++ Submission API

In C++, the three objects described in [Submission APIs](#) are available in the include files `submitter/Connection.hxx`, `submitter/Message.hxx` and `submitter/Recipient.hxx`.

On Windows, you can find these files in the `api\include` subdirectory of where PowerMTA was installed. On Linux, that is `/opt/pmta/api/include`.

All the methods in the submission API throw an exception in case of an error. Any errors render the object unusable, after which you should discard it.

---

#### 7.1.4.1 Connection::Connection

Creates a new connection.

```
Connection(const char* server, int port, const char* name = "", const char* password = "")
```

- `server` server to which to connect, either as a numeric IP address or a host name.
- `port` port number to connect to; use 25 for the default SMTP port.
- `name` user name for authentication with the remote server. Leave name and password empty if you don't want to use authentication.
- `password` password to use for authentication. Authentication is done with CRAM-MD5, so no password is ever sent to the remote server.

---

#### 7.1.4.2 Connection::submit

Submits a message for delivery.

```
void Connection::submit(const Message& message)
```

---

#### 7.1.4.3 Message::Message

Creates a new message.

```
Message::Message(const char* originator)
```

- `originator` the new message's originator.

---

#### 7.1.4.4 Message::setVerp

Sets whether this message should use the VERP extension.

```
void Message::setVerp(bool isVerp)
```

---

#### 7.1.4.5 Message::setReturnType

Sets the type of message return requested in delivery reports.

```
void Message::setReturnType(PmtaMsgRETURN type)
```

- `type` should be one of:
  - `PmtaMsgRETURN_HEADERS` (default) return only this message's headers
  - `PmtaMsgRETURN_FULL` return the full message headers and body

---

#### 7.1.4.6 Message::setEnvelopeId

Sets this message's envelope ID.

```
void Message::setEnvelopeId(const char* envelopeId)
```

---

#### 7.1.4.7 Message::setVirtualMta

Selects the VirtualMTA to use for this message.

```
void Message::setVirtualMta(const char* virtualMta)
```

---

#### 7.1.4.8 Message::setJobId

Sets the job id for this message. This tags the message as belonging to the specified job.

```
void Message::setJobId(const char* jobId)
```

- `jobId` id of the job (printable, non-white space characters only).

---

#### 7.1.4.9 Message::addRecipient

Adds a recipient to this message. Do not make any changes to a recipient after it has been added to the message.

```
void Message::addRecipient(const Recipient& r)
```

---

#### 7.1.4.10 Message::setEncoding

Sets the kind of encoding to perform on data. The encoding applies to all thereafter added data. The actual processing performed depends on the encoding selected: `PmtaMsgENCODING_7BIT` and `PmtaMsgENCODING_8BIT` both select the identity encoding, i.e., no transformation is performed on the data, except that any lines terminated with LF alone are converted to CRLF. In the case of 7-bit

encoding, it is your responsibility to ensure that all bytes have their high bit off. `PmtaMsgENCODING_BASE64` specifies that any data added shall be encoded on the fly using base-64 encoding. No transformation is performed on the data prior to encoding, making it possible to transmit binary content. Note, however, that to effectively transmit binary data, you must also provide the appropriate MIME headers. In fact, the selected encoding should, in general, agree with the Content-Transfer-Encoding header you specify. The default (if this method is never invoked) is `PmtaMsgENCODING_7BIT`.

```
void Message::setEncoding(PmtaMsgENCODING encoding)
```

- encoding Data encoding desired. This should be one of:
  - `PmtaMsgENCODING_7BIT`
  - `PmtaMsgENCODING_8BIT`
  - `PmtaMsgENCODING_BASE64`

---

#### 7.1.4.11 Message::beginPart

Starts the next mailmerge part which will have the given number. Note that part 1 is started automatically, so the first settable number is 2.

```
void Message::beginPart(int partNum)
```

- `partNum` positive integer for the part number. No part number may be specified more than once.

---

#### 7.1.4.12 Message::addData

Appends data to this message. Use this method to give this message both its headers and a body. How the data you add is handled depends on the kind of encoding selected (see `setEncoding`). All the data is entered as-is, with no mailmerge variable substitution being performed. To add mailmerge data, use `addMergeData`.

```
void Message::addData(const char* data, int length)
```

- `data` data to append (as-is).
- `length` length of the data block.

---

#### 7.1.4.13 Message::addData

Appends null-terminated data to this message. This is a convenience method to facilitate adding C strings directly.

```
void Message::addData(const char* data)
```

- `data` null-terminated data to append (as-is).

---

#### 7.1.4.14 Message::addMergeData

Appends merge data to this message. Use this method to give this message both its headers and a body. How the data you add is handled depends on the kind of encoding selected (see `setEncoding`). Currently only 7-bit and 8-bit encodings are supported. If any mailmerge variables are included in the data, they will be substituted by their respective values.

```
void Message::addMergeData(const char* data, int length)
```

- `data` data to append.
- `length` length of the data block.

---

#### 7.1.4.15 Message::addMergeData

Appends null-terminated merge data to this message. This is a convenience method to facilitate adding C strings directly.

```
void Message::addMergeData(const char* data)
```

- `data` null-terminated merge data to append.

---

#### 7.1.4.16 Message::addDateHeader

Adds (appends) a Date: header to this message. This method appends a date header line to the message. It is provided as a convenience function.

```
void Message::addDateHeader()
```

---

#### 7.1.4.17 Recipient::Recipient

Creates a new recipient with the given email address.

```
Recipient::Recipient(const char* address)
```

---

#### 7.1.4.18 Recipient::setNotify

Sets the kind of notification (report) desired for this recipient. By default, notification is only requested upon delivery failure.

```
void Recipient::setNotify(int notifyWhen)
```

- `notifyWhen` should be either `PmtaRcptNOTIFY_NEVER` to indicate that no notification is desired, or one or more of the following flags (joined by a bitwise OR):
  - `PmtaRcptNOTIFY_SUCCESS` notify in case the delivery is successful
  - `PmtaRcptNOTIFY_FAILURE` notify in case of a delivery failure
  - `PmtaRcptNOTIFY_DELAY` notify in case of a delay in the delivery

---

#### 7.1.4.19 Recipient::defineVariable

Defines a new mailmerge variable for this recipient. PowerMTA will substitute the variable for its value during delivery of a mailmerge

message. Carriage returns and line feeds may be added to a variable with \r or \n respectively.

```
void Recipient::defineVariable(char* name, char* value)
```

- name the variable's name.
- value the variable's value.

#### Examples:

```
Recipient rcpt("me@here.com");
rcpt.defineVariable("cr", "A\rB");           // one linebreak
rcpt.defineVariable("lf", "A\nB");           // one linebreak
rcpt.defineVariable("CrLf", "A\r\nB");       // one linebreak
rcpt.defineVariable("literalCrLf", "A\\r\\nB"); // no linebreaks
```

### 7.1.5 C Submission API

In C, the three objects described in [Submission APIs](#) are available as sets of functions which share a common prefix. Connection's functions all begin with `PmtaConn` and are defined in the file `submitter/PmtaConn.h`. Message's functions all begin with `PmtaMsg` and are defined in the file `submitter/PmtaMsg.h`; and Recipient's functions all begin with `PmtaRcpt` and are defined in the file `submitter/PmtaRcpt.h`.

On Windows, you can find these files in the `api\include` subdirectory of where PowerMTA was installed. On Linux, that is `/opt/pmta/api/include`.

In order to differentiate between "out of memory" conditions and other errors, object creation is split in two calls: allocation and initialization. To create an object, you first call its `...Alloc` function. If memory cannot be allocated, a null pointer is returned; otherwise, a pointer to the object is returned, which you then pass to the object's initialization function: `PmtaConnConnect`, `PmtaMsgInit` and `PmtaRcptInit`, for a Connection, Message or Recipient object, respectively.

When done with an object, you should call the object's `...Free` method to discard its resources.

Most of the functions in the submission API return a `BOOL` (boolean) value. You should check every return value to make sure that no errors occurred. If an error occurs, you can find out what went wrong by calling either `PmtaConnGetLastError`, `PmtaMsgGetLastError` or `PmtaRcptGetLastError` (depending on which object the error occurred). Any errors render the object unusable, after which you should discard it with the appropriate `...Free` function.

#### 7.1.5.1 PmtaConnAlloc

Allocates a new connection. After allocating, invoke `PmtaConnConnectRemote` to establish the connection to the PowerMTA server. It returns the new connection, or 0 if not enough memory was available.

```
PmtaConn PmtaConnAlloc()
```

#### 7.1.5.2 PmtaConnConnect

**Deprecated** Use one of `...Connect` functions below.

#### 7.1.5.3 PmtaConnConnectRemote

Connects to a PowerMTA server. No login / authentication is done. Returns `TRUE` in case of success, `FALSE` in case of failure. The cause of the error can be obtained with `PmtaConnGetLastError`.

```
PMTACALL(BOOL) PmtaConnConnectRemote(PmtaConn connection, const char* server, int port);
```

- connection connection to establish.
- server string containing the server to which to connect. This can be a host name or an IP address. Use "127.0.0.1" for the local machine.
- port which port to connect to. Use 25 for the default SMTP port.

#### 7.1.5.4 PmtaConnConnectRemoteAuth

Connects to a PowerMTA server using the given credentials. Returns `TRUE` in case of success, `FALSE` in case of failure. The cause of the error can be obtained with `PmtaConnGetLastError`.

```
PMTACALL(BOOL) PmtaConnConnectRemoteAuth(PmtaConn connection,
                                          const char* server, int port,
                                          const char* username,
                                          const char* password);
```

- connection connection to establish.
- server string containing the server to which to connect. This can be a host name or an IP address.
- port to which port to connect. Use 25 for the default SMTP port.
- username local user name for logging in to PowerMTA. PowerMTA must be configured to allow authentication.
- password password for the given user name.

#### 7.1.5.5 PmtaConnSubmit

Submits a message for delivery. Returns `TRUE` in case of success, `FALSE` in case of failure. The cause of the error can be obtained with `PmtaConnGetLastError`.

```
BOOL PmtaConnSubmit(PmtaConn connection, PmtaMsg message)
```

- connection connection through which to submit the message.
- message message to submit.

#### 7.1.5.6 PmtaConnFree

Discards this connection object, freeing any resources used by it. This automatically closes the connection to the server, if any.

```
void PmtaConnFree(PmtaConn connection)
```

- connection connection to discard.

---

#### 7.1.5.7 PmtaConnGetLastError

Returns the last error that occurred on this connection. This function should only be called immediately after an error occurred.

```
const char* PmtaConnGetLastError(PmtaConn connection)
```

- connection connection to query for an error.

---

#### 7.1.5.8 PmtaConnGetLastErrorType

Returns the type of the last error occurred (one of the PmtaApiERROR codes defined in PmtaApi.h.). This function should only be called immediately after an error occurred.

```
int PmtaConnGetLastErrorType(PmtaConn connection)
```

- connection connection to query for an error.

---

#### 7.1.5.9 PmtaMsgAlloc

Allocates a new message. Once allocated, the new message must be initialized with PmtaMsgInit. Once done with the message, invoke PmtaMsgFree to free its resources. Returns 0 if not enough memory was available.

```
PmtaMsg PmtaMsgAlloc()
```

---

#### 7.1.5.10 PmtaMsgInit

Initializes a new message. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with PmtaMsgGetLastError.

```
BOOL PmtaMsgInit(PmtaMsg message, const char* originator)
```

- message Newly allocated message to initialize.
- originator Address for the new message's originator.

---

#### 7.1.5.11 PmtaMsgFree

Frees memory allocated for this message.

```
void PmtaMsgFree(PmtaMsg message)
```

---

#### 7.1.5.12 PmtaMsgSetVerp

Sets whether this is a VERP message. If you enable VERP, PowerMTA will encode both the originator and recipient addresses in the originator address used for delivering the message. This may make tracking any bounces easier, since you can always derive the original addresses from the address where the bounce is sent. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with PmtaMsgGetLastError.

```
BOOL PmtaMsgSetVerp(PmtaMsg message, BOOL isVerp)
```

- message Message to set.
- isVerp TRUE if the message should be VERP.

---

#### 7.1.5.13 PmtaMsgSetReturnType

Sets the type of message return requested in delivery reports. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with PmtaMsgGetLastError.

```
BOOL PmtaMsgSetReturnType(PmtaMsg message, PmtaMsgRETURN type)
```

- message Message to set.
- type This should be one of:
  - PmtaMsgRETURN\_HEADERS (default) return only this message's headers
  - PmtaMsgRETURN\_FULL return the full message headers and body

---

#### 7.1.5.14 PmtaMsgSetEnvelopeId

Sets this message's envelope ID. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with PmtaMsgGetLastError.

```
BOOL PmtaMsgSetEnvelopeId(PmtaMsg message, const char* envelopeId)
```

- message Message to set.
- envelopeId Envelope ID for this message.

---

#### 7.1.5.15 PmtaMsgSetVirtualMta

Selects the VirtualMTA to use for this message. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with PmtaMsgGetLastError.

```
BOOL PmtaMsgSetVirtualMta(PmtaMsg message, const char* virtualMta)
```

- message Message to set.
- virtualMta name of the VirtualMTA to use.

---

#### 7.1.5.16 PmtaMsgSetJobId

Sets the job id for this message. This tags the message as belonging to the specified job. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaMsgSetJobId(PmtaMsg message, const char* jobId)
```

- `message` Message to set.
- `jobId` id of the job, printable, non-white space characters only.

---

#### 7.1.5.17 PmtaMsgAddRecipient

Adds a recipient to this message. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

Do not make any changes to a recipient after it has been added to the message. Once added, a recipient should be freed with `PmtaRcptFree`.

```
BOOL PmtaMsgAddRecipient(PmtaMsg message, PmtaRcpt recipient)
```

- `message` Message to set.
- `recipient` Recipient to add.

---

#### 7.1.5.18 PmtaMsgSetEncoding

Sets the kind of encoding to perform on data. The encoding applies to all thereafter added data. The actual processing performed depends on the encoding selected: `PmtaMsgENCODING_7BIT` and `PmtaMsgENCODING_8BIT` both select the identity encoding, i.e., no transformation is performed on the data, except that any lines terminated with LF alone are converted to CRLF. In the case of 7-bit encoding, it is your responsibility to ensure that all bytes have their high bit off. `PmtaMsgENCODING_BASE64` specifies that any data added shall be encoded on the fly using base-64 encoding. No transformation is performed on the data prior to encoding, making it possible to transmit binary content. Note, however, that to effectively transmit binary data, you must also provide the appropriate MIME headers. In fact, the selected encoding should, in general, agree with the Content-Transfer-Encoding header you specify. The default (if this function is never invoked) is `PmtaMsgENCODING_7BIT`. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaMsgSetEncoding(PmtaMsg message, PmtaMsgENCODING encoding)
```

- `message` Message to set.
- `encoding` Data encoding desired. This should be one of:
  - `PmtaMsgENCODING_7BIT`
  - `PmtaMsgENCODING_8BIT`
  - `PmtaMsgENCODING_BASE64`

---

#### 7.1.5.19 PmtaMsgBeginPart

Starts the next mailmerge part which will have the given number. Note that part 1 is started automatically, so the first settable number is 2. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaMsgBeginPart(PmtaMsg message, int partNum)
```

- `message` Message to set.
- `partNum` positive integer for the part number. No part number may be specified more than once.

---

#### 7.1.5.20 PmtaMsgAddData

Adds (appends) data to this message. Use this function to give this message both its headers and a body. How the data you add is handled depends on the kind of encoding selected (see `PmtaMsgSetEncoding`). All the data is entered as-is, with no mailmerge variable substitution being performed. To add mailmerge data, use `PmtaMsgAddMergeData`. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaMsgAddData(PmtaMsg message, const char* data, int length)
```

- `message` Message to which to add.
- `data` Data to add.
- `length` Length of the data block.

---

#### 7.1.5.21 PmtaMsgAddMergeData

Adds (appends) mailmerge data to this message. Use this function to give this message both its headers and a body. How the data you add is handled depends on the kind of encoding selected (see `PmtaMsgSetEncoding`). Currently only 7-bit and 8-bit encodings are supported. If any mailmerge variables are included in the data, they will be substituted by their respective values. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaMsgAddMergeData(PmtaMsg message, const char* data, int length)
```

- `message` Message to which to add.
- `data` Merge data to add.
- `length` Length of the data block.

---

#### 7.1.5.22 PmtaMsgAddString

Adds (appends) a string to this message. This function is identical to `PmtaMsgAddData`, except that it expects a '0'-terminated string and computes the length of the data automatically. It is provided as a convenience function only. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaMsgAddString(PmtaMsg message, const char* str)
```

- `message` Message to which to add.
- `str` Null-terminated string to add.

---

#### 7.1.5.23 PmtaMsgAddMergeString

Adds (appends) a mailmerge string to this message. This function is identical to `PmtaMsgAddMergeData`, except that it expects a '0'-

terminated string and computes the length of the data automatically. It is provided as a convenience function only. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaMsgAddMergeString(PmtaMsg message, const char* str)
```

- `message` Message to which to add.
- `str` Null-terminated string to add.

#### 7.1.5.24 PmtaMsgAddDateHeader

Adds (appends) a Date: header to this message. This function appends a date header line to the message. It is provided as a convenience function only.

```
BOOL PmtaMsgAddDateHeader(PmtaMsg message)
```

#### 7.1.5.25 PmtaMsgGetLastError

Returns the last error that occurred on this message. This function should only be called immediately after an error occurred.

```
const char* PmtaMsgGetLastError(PmtaMsg message)
```

- `message` Message to query for an error.

#### 7.1.5.26 PmtaMsgGetLastErrorType

Returns the type of the last error occurred (one of the `PmtaApiERROR` codes defined in `PmtaApi.h`). This function should only be called immediately after an error occurred.

```
int PmtaMsgGetLastErrorType(PmtaMsg message)
```

- `message` Message to query for an error.

#### 7.1.5.27 PmtaRcptAlloc

Allocates a new recipient. Once allocated, the new recipient must be initialized with `PmtaRcptInit`. When no longer needed (e.g., after adding the recipient to a message), its resources should be freed by calling `PmtaRcptFree`. Returns 0 if not enough memory was available.

```
PmtaRcpt PmtaRcptAlloc()
```

#### 7.1.5.28 PmtaRcptInit

Initializes a new recipient. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaRcptInit(PmtaRcpt recipient, const char* address)
```

- `recipient` Newly allocated recipient to initialize.
- `address` email address for the recipient.

#### 7.1.5.29 PmtaRcptFree

Frees storage allocated by this recipient. To avoid memory leaks, this function should be called on every recipient allocated, whether it has been added to a message or not.

```
void PmtaRcptFree(PmtaRcpt recipient)
```

#### 7.1.5.30 PmtaRcptSetNotify

Sets the kind of notification (report) desired for this recipient. By default, notification is only requested upon delivery failure. Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaRcptSetNotify(PmtaRcpt recipient, int notifyWhen)
```

- `recipient` Recipient to set
- `notifyWhen` This should be either `PmtaRcptNOTIFY_NEVER` to indicate that no notification is desired, or one or more of the following flags (joined by a bitwise OR):
  - `PmtaRcptNOTIFY_SUCCESS` notify in case the delivery is successful
  - `PmtaRcptNOTIFY_FAILURE` notify in case of a delivery failure
  - `PmtaRcptNOTIFY_DELAY` notify in case of a delay in the delivery

#### 7.1.5.31 PmtaRcptDefineVariable

Defines a new mailmerge variable for this recipient. PowerMTA will substitute the variable for its value during delivery of a mailmerge message. Carriage returns and line feeds may be added to a variable with `\r` or `\n` respectively (Assuming that `rcpt` is of type `PmtaRcpt` and is properly initialized). Returns TRUE in case of success, FALSE in case of failure. The cause of the error can be obtained with `PmtaMsgGetLastError`.

```
BOOL PmtaRcptDefineVariable(PmtaRcpt recipient, const char* name, const char* value)
```

- `recipient` recipient whose variable is being defined.
- `name` the variable's name.
- `value` the variable's value.

#### Examples:

```
if (!PmtaRcptDefineVariable(rcpt, "cr", "A\rB")) { // one linebreak
    fprintf(stderr, "Error defining cr variable: %s\n",
            PmtaRcptGetLastError(rcpt));
    return 1;
}
```

```

}
if (!PmtaRcptDefineVariable(rcpt, "lf", "A\nB")) { // one linebreak
    fprintf(stderr, "Error defining lf variable: %s\n",
            PmtaRcptGetLastError(rcpt));
    return 1;
}
if (!PmtaRcptDefineVariable(rcpt, "crlf", "A\r\nB")) { // one linebreak
    fprintf(stderr, "Error defining crlf variable: %s\n",
            PmtaRcptGetLastError(rcpt));
    return 1;
}
if (!PmtaRcptDefineVariable(rcpt, "literalCrlf", "A\\r\\nB")) { // no linebreaks
    fprintf(stderr, "Error defining literalCrlf variable: %s\n",
            PmtaRcptGetLastError(rcpt));
    return 1;
}
}

```

#### 7.1.5.32 PmtaRcptGetLastError

Returns the last error occurred on this recipient. This function should only be called immediately after an error occurred.

```
const char* PmtaRcptGetLastError(PmtaRcpt recipient)
```

- recipient Recipient to query for an error.

#### 7.1.5.33 PmtaRcptGetLastErrorType

Returns the type of the last error occurred (one of the PmtaApiERROR codes defined in PmtaApi.h). This function should only be called immediately after an error occurred.

```
int PmtaRcptGetLastErrorType(PmtaRcpt recipient)
```

- recipient Recipient to query for an error.

### 7.1.6 .NET Submission API

Documentation for the .NET APIs is provided in doc folder.

On Windows, it can be found in the api\dotnet subdirectory of where PowerMTA was installed.

## 7.2 Pickup Directory

PowerMTA can be instructed to pick up message files from a directory. To submit a message, you write it to a file, formatting it according to Internet email standards but preceding it with envelope information in special headers: x-sender, x-receiver and, optionally, x-envid and x-job. You then move this file to the pickup directory, from where PowerMTA reads it into its message spool and performs delivery. It is recommend to add no more than 5000 recipients per pickup file.

The x-sender header should be in the first line in the file, followed by one or more x-receiver headers, each in its own line. You can also optionally set the DSN envelope ID and the job ID by adding the x-envid and x-job headers respectively. The x-sender and x-receiver headers must contain the email addresses only, with *no* free form name, no <>s, etc. For example:

```

x-sender: me@over.here
x-receiver: you@over.there
x-receiver: someone.else@some.other.place
x-envid: me-0001
x-job: love-2003-01
From: Myself Personally <me@over.here>
To: You <you@over.there>
Date: Thu, 6 Jul 2000 07:45:43 -0400
Subject: I love it

Hey, did you know that ...

```

Note that in the example above, someone.else@some.other.place is BCC'ed, i.e., blind copied: he or she receives a copy of the message but is not included in the recipient headers.

PowerMTA's pickup directory also supports the x-sender-override header, which allows you to override the contents of x-sender. If x-sender-override is present, its contents override the sender specified in x-sender, if any. This is useful when you wish to customize the message sender despite the software you're using to create the messages not allowing for that. Typically that software will still allow you to pass additional headers, such as x-sender-override.

Other than reading the sender and receiver email address headers and removing them from the message, no other header or body processing takes place. You are responsible for properly formatting the message.

Instead of creating the file elsewhere and then moving to the pickup directory, it is also possible to create the file directly in it. However, since PowerMTA must repeatedly attempt to lock the file for exclusive access, it is less efficient to do so.

To configure PowerMTA for a pickup directory, specify the [pickup](#) directory. You must also specify a directory as the "bad mail" directory, to which messages are moved if there is a permanent error processing the message.

#### 7.2.1 BSMTP files

The pickup directory also supports BSMTP (batched SMTP) files. A BSMTP file is simply a file that contains all of the SMTP commands necessary for submitting a message batched together in one text file. For example:

```

MAIL FROM:<me@over.here>
RCPT TO:<you@over.there>
DATA
From: Myself Personally <me@over.here>
To: You <you@over.there>
Date: Thu, 6 Jul 2000 07:45:43 -0400
Subject: I love it

Hey, did you know that ...
.

```

The pickup directory supports the following SMTP commands: HELO, EHLO, MAIL, RCPT and DATA. Additionally, it also supports PowerMTA's extended SMTP mailmerge commands (XMRG, XDFN, XPRT and XACK), allowing for both mailmerge and non-mailmerge messages to be created.

To be recognized as such, BSMTMP files need not be marked in any special way, other than the first command be one of: HELO, EHLO, MAIL, XACK or XACK.

Like in traditional pickup files, only one message may be submitted per file. However, since mailmerge is supported, the message submitted may actually be a template used for many recipients. See [Pickup Directory Submission](#) for examples on using the pickup directory for submitting mailmerge messages.

If any commands cannot be successfully processed, the entire file is rejected and moved to the "bad file" directory. For certain recipient address related errors, however, you have the option of requesting that PowerMTA bounce the recipient rather than rejecting the file during submission. This is activated by disabling recipient acknowledgements, using PowerMTA's XACK SMTP extension. For example:

```
XACK OFF
MAIL FROM:<me@over.here>
RCPT TO:<you@over.there>
RCPT TO:<notyou@not.here>
DATA
From: Me Personally <me@over.here>
To: (recipients omitted)
Date: Thu, 6 Jul 2000 07:45:43 -0400
Subject: I love it

Hey, did you know that ...
.
```

In the example above, despite the error in the second recipient's domain, the file would not be rejected, but a bounce would be sent to the address in the MAIL FROM.

[Mailmerge SMTP extensions](#) describes PowerMTA's mailmerge extensions; look at [The XACK SMTP extension](#) for the XACK extension.

## 7.3 Pipe Delivery

PowerMTA can be configured to deliver email to another program on the same machine. It does so by starting the program for each new recipient and writing the message to the program's standard input. The mechanism used to pass on the message is called "pipe", so this feature is named "pipe delivery".

To use pipe delivery, you add domain definitions for one or more domains, specifying a type of pipe and defining the command to run for starting the pipe receiver program. For example:

```
<domain bounces.company.com>
  type pipe
  command "c:\\path\\bounceproc.exe --envid $envid"
</domain>
```

Note that since the backslash character is also used for escaping other characters (such as spaces, quotes and the dollar sign), you need to specify it twice where you want just one to appear. On Windows, you can also use the (forward) slash character to specify paths, so in the example above it could have been `c:/path/bounceproc.exe`.

Before starting the program, PowerMTA substitutes any macros found in the command line (such as `$envid` in the example above), allowing you to pass on additional message-related parameters to the pipe receiver program. The complete list of macros supported and other configuration directives for pipe domains are described in [Pipe Delivery Directives](#).

### 7.3.1 Delivering mail to local users

PowerMTA's pipe delivery can be used to deliver email to local users. The special `local` domain name makes this simple to configure by concentrating all email destined to recipients on any of the local domains. PowerMTA will use the hostname of the OS and any host-names configured in PowerMTA, to determine if an email is local. On Unix systems, it is often sufficient to specify `procmail` as the command to execute for pipe delivery:

```
<domain local>
  type pipe
  command "/usr/bin/procmail -t -Y -d $user"
</domain>
```

Note, however, that PowerMTA does not currently process the `/etc/aliases` or `.forward` files.

Alternatively, the same could be done by explicitly specifying the domain name as in this example:

```
<domain bounces.yourdomain.com>
  type pipe
  command "/usr/bin/procmail -t -Y -d $user"
</domain>
```

The same is also available on Windows, but you will need to provide a pipe delivery processor that can write the messages in a way your user agents (i.e., email clients) understand.

### 7.3.2 Pipe Delivery Programs Included With PowerMTA

PowerMTA comes with two (fully functional) sample programs to be used with the new pipe delivery functionality; `appendtofile`, which appends messages to a file, and `newfile`, which writes each message to a separate (new) file.

**appendtofile** appends the message and / or arguments passed to it to the file specified. The syntax for `appendtofile` is:

```
appendtofile [--nomsg] filename [text]
```

- where *filename* is the name of the file to which to append the data and *text* is an optional text or a supported macro to write in front of each message. If the first command line argument given to `appendtofile` is `--nomsg` (or `/nomsg`), it will just append the text or the macro's value to the file. You can thus choose to either save the complete message to the file or just the summary data passed on the command line, such as only the recipient address of the message (when using the `$to` macro).

**newfile** writes each message to a separate (new) file. The syntax for `newfile` is simply:

```
newfile dirname
```

- where *dirname* is the directory where the message file should be created. If the given directory does not yet exist, `newfile` automatically creates it. Each message file name is automatically generated by `newfile`, and will have a `.msg` extension.

For greatest flexibility, you can use the supported macro substitutions described in [Pipe Delivery Directives](#) to automatically pass

message-dependent file names and directory names. For example, `appendtofile c:/pipe/$domain` in the command directive will append messages to the file named after the domain of the recipient address, given the presence of the `$domain` macro. A message sent to `test@port25.com` would be appended to a file named `port25.com` in the directory `c:\pipe`, while a message sent to `test@powermta.com` would be appended to a file named `powermta.com` in the same directory.

In the typical scenario where the pipe delivery applications are used to store and process bounces, using the `$domain` macro results in files named after the originator / SMTP MAIL FROM domain you used in your original messages, since the recipient address of a bounce report is the original MAIL FROM address used.

For example, if you used a MAIL FROM domain of `bounces5.port25.com` for all of your messages for customer or campaign ID #5, the file `bounces5.port25.com` would contain the bounce reports for the messages of this customer or campaign. The configuration for this would be:

```
<domain bounces5.port25.com>
  type pipe
  command "c:/pmta/bin/appendtofile.exe c:/pipe/$domain"
</domain>
```

Alternatively, with the `--nomsg` and passing the `$to` macro to `appendtofile` as well, like in:

```
<domain bounces5.port25.com>
  type pipe
  command "c:/pmta/bin/appendtofile.exe --nomsg c:/pipe/$domain $to"
</domain>
```

only the recipients would be written to a file (given the `$to`), which is named after the domain of the recipients (given the `$domain`), `port25.com`.

With `newfile`, if you had a configuration like:

```
<domain bounces5.port25.com>
  type pipe
  command "c:/pmta/bin/newfile.exe c:/pipe/$domain"
</domain>
```

`newfile` will create subdirectories named after the recipients' domains, and only create message files within the appropriate subdirectories. Using the example above with the domain `bounces5.port25.com` as the domain of your originator address, when bounces come in, the directory created would be `c:\pipe\bounces5.port25.com`, and which would contain only files / messages received for the recipients of that domain. If you had PowerMTA handling inbound mail for 50 separate domains, `newfile` would create 50 separate subdirectories, each named for a different domain, and containing only messages for that domain.

7.3.3 Accessing Files Created By The Sample Applications

Since, in principle, new messages may be delivered at any time through the sample applications, special care must be taken when accessing the files these applications create to avoid loss of data

`newfile` solves concurrency conflicts by using a naming convention: it initially creates files with a `.tmp` suffix ("extension") and when done writing to them, renames them to `.msg`. This makes it very easy to access the files: you can freely access all the `.msg` files, while leaving any `.tmp` files alone.

Doing the same for `appendtofile` is a little more challenging, since it writes data to the same file and due to the differences in file lock handling across the PowerMTA platforms. To allow access to `appendtofile`-generated files in a simple and uniform way, SparkPost provides a program named `pmtagetfile`, which moves the contents of a given file.

To use `pmtagetfile`, you simply run it passing the name of the file to which `appendtofile` writes and the new destination file name on the command line. For example,

```
pmtagetfile c:/pmta-pipe/port25.com.txt c:/tmp/port25.com.txt
```

moves the file `c:\pmta-pipe\port25.com.txt` to `c:\tmp\port25.com.txt`.

Once in the new directory, the file is out of `appendtofile`'s reach, and you can safely process it as required. Note that the `pmtagetfile` program will fail if the first file listed does not exist (nothing to copy from) or if the second file already exists (preventing accidental overwriting of previous data). Also, `pmtagetfile` does not work across file system boundaries, i.e., the new file must be on the same file system or drive (on Unix and Windows systems, respectively). To move a file to a different file system, use `pmtagetfile` to first move it away from `appendtofile` to a safe place on the same file system, and then move that file with your usual tools (Explorer, shell commands, etc.) to the desired location.

Please note that `pmtagetfile` is only intended for use with files written by `appendtofile` or with files written to by File Delivery.

7.3.4 Writing Your Own Pipe Delivery Programs

For convenience and flexibility, the sample pipe delivery programs detailed above are also provided in source code, allowing you to modify or customize as needed.

The program used for pipe delivery should terminate with certain specific exit codes. On Unix systems, these are defined in `/usr/include/sys/exits.h`. On Windows, a corresponding file is included in the `api\include` directory. Depending on the exit code, PowerMTA will consider the delivery to have succeeded or to have temporarily or permanently failed and generate any DSN delivery reports as appropriate:

| exit code      | action | DSN status                              |
|----------------|--------|-----------------------------------------|
| EX_OK          | delete | 2.0.0 (success)                         |
| EX_USAGE       | bounce | 5.5.4 (invalid command arguments)       |
| EX_NOUSER      | bounce | 5.1.1 (bad destination mailbox address) |
| EX_NOHOST      | bounce | 5.1.2 (bad destination system address)  |
| EX_CANTCREAT   | bounce | 5.2.0 (mailbox-related failure)         |
| EX_UNAVAILABLE | bounce | 5.3.0 (mail system-related failure)     |
| EX_SOFTWARE    | bounce | 5.3.0 (mail system-related failure)     |
| EX_OSFILE      | bounce | 5.3.0 (mail system-related failure)     |
| EX_NOINPUT     | bounce | 5.3.0 (mail system-related failure)     |
| EX_OSERR       | retry  | 4.3.0 (mail system-related failure)     |
| EX_IOERR       | retry  | 4.3.0 (mail system-related failure)     |
| EX_TEMPFAIL    | retry  | 4.0.0 (transient failure)               |
| EX_DATAERR     | bounce | 5.5.2 (syntax error)                    |
| EX_PROTOCOL    | bounce | 5.5.0 (protocol-related failure)        |
| EX_CONFIG      | bounce | 5.3.5 (system incorrectly configured)   |
| EX_NOPERM      | bounce | 5.7.1 (delivery not authorized)         |

If you use `appendtofile` but for some reason would like to replace `pmtagetfile`, please look into `pmtagetfile`'s sources first to better

understand how it works to avoid file locking conflicts.

## 8 VirtualMTA Support

### 8.1 Overview

Increasing volumes of email have prompted many ISPs and email administrators to proactively filter or block incoming email. While their intentions are honest, this also results in the blocking of legitimate mail for sites running on or delivering through a shared mail gateway. Traditionally, all mail sent from a shared MTA or gateway is sent from same host name and IP addresses. Since ISPs and email administrators tend to initially key on the source IP address for blocking incoming mail, more aggressive senders, or simply higher volume customers delivering through a shared MTA will almost certainly put all others at risk of being blocked.

To maximize deliverability and minimize collateral blocking of legitimate email on shared gateways, SparkPost has developed *VirtualMTA* technology within PowerMTA. Simply put, this functionality allows sites to define and control both the source IP address and host name for each message. This allows for creating VirtualMTAs for separate and specific campaigns, customers or departments, while still running just one instance of PowerMTA.

### 8.2 VirtualMTA Definitions

VirtualMTAs are defined via the new `<virtual-mta>` directive. Defining a VirtualMTA entails creating a `<virtual-mta>` tag with the desired name and listing its host name and SMTP source IP(s). For example, if you wanted to create a VirtualMTA named `mta1`, you would simply add in the configuration file:

```
<virtual-mta mta1>
  smtp-source-host 1.2.3.4 mta1.port25.com
</virtual-mta>
```

Changes to the VirtualMTA definitions are reloadable via the `pmta reload` command and do not require a restart.

Note that if you leave out the `smtp-source-host` directive in the definition, PowerMTA will use the corresponding global settings for that VirtualMTA. Since that may or may not be the desired result, SparkPost always recommends that you specify at least one `smtp-source-host` within each VirtualMTA definition. For example, you could use the desired host name as the VirtualMTA name, but you would still need to specify the host-name directive to ensure that the right host name is used:

```
<virtual-mta mta2.port25.com>
  smtp-source-host 2.3.4.5 mta2.port25.com
</virtual-mta>
```

Lastly, if more than one `smtp-source-host` address is listed in a definition, PowerMTA will use the various source IPs in round robin fashion for connections from that VirtualMTA:

```
<virtual-mta mta3>
  smtp-source-host 1.2.3.4 mta4.port25.com
  smtp-source-host 2.3.4.5 mta5.port25.com
  smtp-source-host 3.4.5.6 mta6.port25.com
</virtual-mta>
```

VirtualMTA directives only apply when the VirtualMTA has been selected as described in [Selecting a VirtualMTA or VirtualMTA Pool](#). Otherwise, they are simply ignored. More information about [smtp-source-host](#) directive can be found in [PowerMTA Configuration](#) section.

### 8.3 VirtualMTA Pools

PowerMTA supports the defining and selecting of VirtualMTA pools per message or campaign as well, which are basically multiple VirtualMTAs grouped in a pool. When a pool is defined and selected for a campaign, PowerMTA will use the VirtualMTAs listed in round robin fashion for connections from that pool.

VirtualMTA pools are defined via the new `<virtual-mta-pool>` directive. Defining a pool entails creating a `<virtual-mta-pool>` group with the desired name and listing the VirtualMTAs that you want belonging to that pool. For example, if you wanted to create a VirtualMTA pool named `pond`, you would first define the specific VirtualMTAs to be part of `pond`, as described in [VirtualMTA Definitions](#), and then subsequently list each of the VirtualMTAs that are to be included using the `virtual-mta` directive:

```
<virtual-mta mta1>
  smtp-source-host 1.2.3.4 mta1.port25.com
</virtual-mta>

<virtual-mta mta2>
  smtp-source-host 2.3.4.5 mta2.port25.com
</virtual-mta>

<virtual-mta-pool pond>
  virtual-mta mta1
  virtual-mta mta2
</virtual-mta-pool>
```

Note that PowerMTA will not accept a VirtualMTA pool definition if the pool name equals a previously defined VirtualMTA name, given that you select a VirtualMTA pool and/or VirtualMTA via the same headers and function calls, as detailed in [Selecting a VirtualMTA or VirtualMTA Pool](#).

A `<domain>` may be defined inside of a `<virtual-mta-pool>`. If this is done, the defined directives will be inherited by any VirtualMTA that is part of the pool. If a conflicting directive is defined directly in the given VirtualMTA, the directive in the VirtualMTA overrides the one defined in the `<virtual-mta-pool>`.

The following directives are also supported:

```
cold-virtual-mta
domain-key
max-smtp-out
max-smtp-msg-rate
include-headers-from
```

### 8.4 Selecting a VirtualMTA or VirtualMTA Pool

PowerMTA supports various methods of selecting a VirtualMTA (or pool), depending on your submission method:

- Submission APIs:
  - by defining the `*vmta` mailmerge variable in mailmerge messages.

- by using the `setVirtualMta` or equivalent method (see [trace](#) for details).
- Pickup directory submission:
  - by defining the `*vmta` mailmerge variable in BSMTP messages.
  - by pattern matching based on the MAIL FROM, RCPT TO, or a header.
- SMTP submission:
  - by defining the `*vmta` mailmerge variable in mailmerge messages.
  - by including the X-virtual-MTA header in the message (see [Selecting a VirtualMTA with a X-virtual-MTA Header](#) below for details).
  - by regular expression matching of the SMTP MAIL FROM or RCPT TO addresses, or a message header; see [<pattern-list> Directives](#) and the [pattern-list](#) directive as well as [Selecting a VirtualMTA by Regular Expression Matching](#) below for details).
  - by configuring a default VirtualMTA on a `<source>` directive selected by the IP address and port in the PowerMTA machine that you connect to (see the [smtp-listener](#) directive as well as [VirtualMTA Pools](#) below for details).
  - by configuring a default VirtualMTA on a `<source>` directive selected by the source IP address (see the [default-virtual-mta](#) directive as well as [Selecting a VirtualMTA by the Source IP Address](#) below for details).

The selection methods above are listed in the order of precedence within each submission methods.

Using mailmerge may effect how VirtualMTA selection is done. When using mailmerge and the API or SMTP, the `x-virtual-mta` will apply to the entire mailmerge job, not individual recipients (in most cases using the first defined `x-virtual-mta`).

When using mailmerge and pattern lists, VirtualMTA selection will be for each recipient when matching the RCPT TO, and will only be for each recipient for the MAIL FROM when using the `*from` mailmerge variable. When not using the `*from`, or when using header matching, *all* recipients in the mailmerge job will use the same VirtualMTA (in most cases the first matched pattern).

Note that one selects a VirtualMTA pool using the same exact header, directives or function call, but using the pool name instead of the VirtualMTA name.

#### 8.4.1 Selecting a VirtualMTA with a X-virtual-MTA Header

To select a VirtualMTA with the X-virtual-MTA header, you need to:

1. define the VirtualMTA(s) or VirtualMTA pool(s) in the configuration file using the `<virtual-mta>` directive.
2. define the specific IP address or IP address range for the source of the messages that it is applicable using the expanded `<source>` directive.
3. define the ability for PowerMTA to process the `x-virtual-mta` header via the `process-x-virtual-mta` directive, within the `<source>` directive defined in step 2.

After setting up the configuration file properly, you simply need to then add the `x-virtual-mta` header to your message specifying the VirtualMTA name, like in

```
x-virtual-mta: yyyy
```

where `yyyy` is the VirtualMTA name in the configuration file. PowerMTA will remove the `x-virtual-mta` header by default before ultimate delivery, so it will not be seen by the recipient's mail gateway or the recipient.

For example, assuming that you had the following VirtualMTA definition in your configuration file:

```
<virtual-mta mta1>
  smtp-source-host 1.2.3.4 mta1.port25.com
</virtual-mta>
```

and you wanted PowerMTA to process the `x-virtual-mta` header for mail arriving from the local IP address `127.0.0.1` for selecting a VirtualMTA, you would need to have the following in your configuration file:

```
<source 127.0.0.1>
  process-x-virtual-mta yes # allow selection of a VirtualMTA using x-virtual-mta header
  always-allow-relaying yes # allow feeding from 127.0.0.1
</source>
```

With this, any messages submitted from the local IP address `127.0.0.1` that contained the header `x-virtual-mta: mta1` would have PowerMTA using VirtualMTA `mta1` and its parameters when making connections to deliver these messages. One can also select a VirtualMTA pool to be used for messages in the same manner.

The entire message (in this example, submitted through the pickup directory) could thus read:

```
x-sender: me@over.here
x-envid: me-0001
x-receiver: you@over.there
x-virtual-mta: mta1
From: Me <me@over.here>
To: You <you@over.there>
Date: Thu, 13 Mar 2018 07:45:43 -0400
Subject: PowerMTA is the best

Hey, this new VirtualMTA technology really improves delivery rates!
```

#### 8.4.2 Selecting a VirtualMTA by Regular Expression Matching

Sites can also have PowerMTA use a VirtualMTA or VirtualMTA pool for specific messages by way of regular expression pattern matching of the SMTP MAIL FROM address, the SMTP RCPT TO address, or a header, based on the IP address of the source of the message.

To select a VirtualMTA or VirtualMTA pool based on regular expression pattern matching, you

1. define the VirtualMTAs or VirtualMTA pools in the configuration file using the `<virtual-mta>` directive.
2. define the name for the regular expression pattern list using the `<pattern-list>` directive.
3. define the actual regular expression pattern to use for matching the SMTP MAIL FROM, the RCPT TO address, or a header as well as subsequent VirtualMTA or VirtualMTA pool to use when a match occurs, via the `mail-from`, `rcpt-to`, and header directives. These directives are defined within the `<pattern-list>` definition created in step 2.

4. define the specific IP address or IP address range for the source of the messages that is applicable within the expanded <source> directive.
5. define the pattern-list directive using the corresponding name defined in step 2 within the <source> directive defined in step 4. Conditional pattern matching as defined in [<pattern-list> Directives](#) may also be used.

For example, assuming that you had the following VirtualMTA definitions in your configuration file:

```
<virtual-mta mta1>
  smtp-source-host 1.2.3.4 mta1.port25.com
</virtual-mta>

<virtual-mta mta2>
  smtp-source-host 2.3.4.5 mta2.port25.com
</virtual-mta>
```

and you wanted PowerMTA to match the patterns "abc" and "xyz" at the beginning of the MAIL FROM address in order to select the appropriate VirtualMTA to use for each, you would need to have at a minimum the following in your configuration file:

```
<pattern-list group2>
  mail-from /^abc/ virtual-mta=mta1
  mail-from /^xyz/ virtual-mta=mta2
</pattern-list>

<source 127.0.0.1>
  pattern-list group2 # this selects the pattern list for messages from this IP
  always-allow-relaying yes # allow feeding from 127.0.0.1
</source>
```

With this, any messages submitted from the local IP address 127.0.0.1 would have PowerMTA using regular expression matching on the MAIL FROM address, and if PowerMTA found "abc" at the beginning of the address (for example [abc-2890u8@bounces.port25.com](#)), VirtualMTA mta1 and its parameters would be used when making connections to deliver these messages. For messages submitted that had a MAIL FROM starting with "xyz", for example xyz-1jwer@bounces.port25.com, VirtualMTA mta2 and its parameters would instead be used when making connections to deliver these messages.

To additionally select some other VirtualMTA for recipients at yahoo.com, you could add a rcpt-to directive as below:

```
<pattern-list group2>
  mail-from /^abc/ virtual-mta=mta1
  mail-from /^xyz/ virtual-mta=mta2
  rcpt-to /@yahoo.com$/ virtual-mta=mta3
</pattern-list>

...
```

Per the examples above, you can see that PowerMTA can use different matching patterns for different campaigns, customers and/or mailings simultaneously, in order to use different VirtualMTAs or VirtualMTA pools for each.

More than one mail-from, rcpt-to and header directives are allowed within a <pattern-list> definition, and since the first match is used, you will want to list the most likely matches first. Only one pattern-list is allowed within a <source> group.

Conditional matching is allowed as well. For example, if you want wanted all emails that have a MAIL FROM of X, and a RCPT TO of Y, that is allowed in a pattern similar to the following:

```
<pattern-list patterns>
  mail-from /X/ <pattern-list>
    rcpt-to /Y/ virtual-mta=vmta2
    rcpt-to /Z/ virtual-mta=vmta1
    * virtual-mta=vmta2
  </pattern-list>
</pattern-list>
```

For more information on conditional pattern matching, see [<pattern-list> Directives](#).

Note that simple to advanced regular expression pattern matching is supported, and which is based on Perl Compatible Regular Expressions (PCRE).

### 8.4.3 Selecting a VirtualMTA by the IP Address/Port Receiving Connections

For added flexibility, PowerMTA supports the ability to select a VirtualMTA based on the IP address (and TCP port number) on the PowerMTA machine to which you connect when submitting your messages.

For example, if you have 10 VirtualMTAs defined, each with a different IP address, you could select the VirtualMTA to use for each message simply by making your SMTP connection to the specific IP address defined on your PowerMTA machine for the specific VirtualMTA.

This is accomplished through named <source> tags, in conjunction with the smtp-listener directive. Named <source> tags allow you to specify directives that override those normally selected by source IP address; they apply only when referenced in the smtp-listener directive (and for authenticated users — see [Implementation for Inbound Connections](#)).

For example, to select VirtualMTA vmta1 by its assigned smtp-source-host address 1.2.3.4 and the standard SMTP port (25), you would configure:

```
<virtual-mta vmta1>
  smtp-source-host 1.2.3.4 vmta1.yourdomain.com
  ...
</virtual-mta>

<source vmta1>
  default-virtual-mta vmta1
</source>

smtp-listener 1.2.3.4:25 source=vmta1
```

Multiple smtp-listener entries can be specified in a configuration file, each of which specifying a different IP address and/or port.

Since the named source's directives apply to anyone who can connect to the listener's IP address and port, one should be careful about what is entered there. For example, always-allow-relaying should most probably **not** be enabled in the named source. Instead, permission for relaying, for accessing the mailmerge extensions, etc. should be based on authenticated sources (see [Implementation for Inbound Connections](#)) or on the source IP address, as in the example below:

```
<virtual-mta vmta1>
```

```

    smtp-source-host 1.2.3.4 vmta1.yourdomain.com
    ...
</virtual-mta>

smtp-listener 1.2.3.4:25 source=vmta1
smtp-listener 0.0.0.0:25

<source vmta1>
default-virtual-mta vmta1
</source>

<source 10/8>
allow-mailmerge yes
always-allow-relaying yes
</source>

<source 0/0>
allow-mailmerge no
always-allow-relaying no
</source>

```

Above, VirtualMTA vmta1 is selected by default for any messages received through 1.2.3.4 on the PowerMTA machine. The default listener on 0.0.0.0:25 is retained through its own `smtp-listener` directive. Permission for relaying and for using the mailmerge extensions is granted to connections coming from `<source 10/8>`, whether they connect to 1.2.3.4 or not.

#### 8.4.4 Selecting a VirtualMTA by the Source IP Address

For sites that are using applications that do not allow them to create custom X-headers within their messages, or for sites feeding PowerMTA messages from a variety of different SMTP applications and machines, PowerMTA supports the ability to select and use VirtualMTAs based only on the IP address of the source of the message.

To select a VirtualMTA or VirtualMTA pool based on the IP address of the original source of the message, you need to:

1. define the VirtualMTAs or VirtualMTA pools in the configuration file using the `<virtual-mta>` directive.
2. define the specific IP address or IP address range for the source of the messages that is applicable using the expanded `<source>` directive.
3. define the VirtualMTA to use for messages via the new `default-virtual-mta` directive within the `<source>` directive defined in step 2.

For example, assuming that you had the following VirtualMTA definition in your configuration file:

```

<virtual-mta mta1>
    smtp-source-host 1.2.3.4 mta1.port25.com
</virtual-mta>

```

and you wanted only mail coming from the local IP address 127.0.0.1 to use this VirtualMTA by default for each message (without having to specify an `x-virtual-mta` header in the messages), you would need to have at a minimum the following in your configuration file:

```

<source 127.0.0.1>
    default-virtual-mta mta1 # selects a VirtualMTA/pool by source of the message
    always-allow-relaying yes # allow feeding from 127.0.0.1
</source>

```

With this, any messages submitted from the local IP address 127.0.0.1 would have PowerMTA using VirtualMTA mta1 and its parameters when making connections to deliver these messages. One can also select a VirtualMTA pool to be used for messages coming from the specific IP address by instead specifying a VirtualMTA pool name via the `default-virtual-mta` directive.

You can basically define as many different source IP addresses or IP address ranges for messages in the configuration file as you wish, with each using a different VirtualMTA or VirtualMTA pool.

#### 8.4.5 Selecting a VirtualMTA by a VirtualMTA IP address

In addition to the above, a VirtualMTA can be selected based on which PowerMTA IP is used for the connection, and the corresponding VirtualMTA that is configured with that IP. To use this selection method, the source directive `default-virtual-mta` will need to be used with a setting of `by-smtp-source-ip` in place of a VirtualMTA name.

When configured in this way, on an inbound SMTP connection PowerMTA will use as the default VirtualMTA that whose `smtp-source-host` setting equals the destination IP address on the inbound connection.

**Example:**

```

<virtual-mta mta1>
    smtp-source-host 1.2.3.4 vmta1.port25.com
</virtual-mta>

<source 0/0>
    default-virtual-mta by-smtp-source-ip
</source>

```

In the above example, connecting to IP address 1.2.3.4 would automatically route the message through VirtualMTA “mta1”.

### 8.5 Changing VirtualMTAs on the Fly

There may be times when you need to change or even cancel a VirtualMTA definition after mail has already been queued for it, so PowerMTA has been designed to handle these scenarios.

#### 8.5.1 Changing a VirtualMTA

If you change the host name and/or source IP addresses of a VirtualMTA after email has already been queued, PowerMTA will use the new information for all new connections once the configuration file has been reloaded. However, if connections were already established when the changes are loaded, the old VirtualMTA information will be used for the duration of those connections.

#### 8.5.2 Canceling a VirtualMTA

If you would like to cancel a VirtualMTA after messages have already been queued, you can simply delete or comment its definition out of the configuration file (and run the `pmta reload` command). PowerMTA will use the new information for all new connections once the configuration file has been reloaded. However, if connections were already established when the changes are loaded, the old VirtualMTA information will be used for the duration of those connections.

If a message selects a VirtualMTA which does not exist when PowerMTA attempts to deliver it, that message is bounced with the DSN status

#### 5.3.5 (specified VirtualMTA does not exist)

## 9 Mailmerge Support

### 9.1 Overview

PowerMTA includes internal mailmerge support to maximize performance and efficiency in the creation, queueing, and delivery of customized email messages. PowerMTA's mailmerge implementation supports both simple keyword or phrase merging as well as merging of larger blocks of data (such as paragraphs or body parts) for respective recipients. Merging does not depend on the actual message contents and can thus be used for both text and HTML formatted messages.

The process is basically similar to general mailmerge implementations. Your application generates a pre-formatted merge file or *recipe* that includes all of the recipient data (recipient addresses, names, order information, etc.) to be merged for each recipient, along with a message template. The message template includes the respective pointers to the variable content and data blocks to be merged in. Based on this, PowerMTA then creates the final merged (i.e., customized) messages for each recipient at actual delivery time.

PowerMTA mailmerge is supported with all of the current mail submission methods. Mailmerge submissions through SMTP and the pickup directory (BSMTP) are supported through special protocol extensions.

### 9.2 Benefits

While PowerMTA is designed to handle the queueing and delivery of very large volumes of customized messages, inline mailmerge support via PowerMTA Mailmerge offers additional performance and efficiency advantages to all components of your solution. Some of the obvious benefits are:

- decreased responsibility and load on your CRM or email marketing application by shifting the merging to PowerMTA.
- increased feeding performance by application to PowerMTA, given that you would be feeding far fewer messages to PowerMTA for the same number of recipients.
- increased efficiency and delivery throughput given the decrease of the I/O impact on your PowerMTA machine. There will be a huge decrease in the number of queued mail files in the spool for the same number of recipients, allowing for maximum scalability. For example, instead of one spool file for each recipient in the queue, with PowerMTA Mailmerge you could have 10,000 recipients in a single spool file.

### 9.3 Configuration Requirements

No special configuration is needed to submit mailmerge messages through the submission APIs or the pickup directory. However, to use the mailmerge SMTP extensions, you must enable them with the `allow-mailmerge` directive for the IP address(es) from which you will be submitting the messages:

```
<source 127.0.0.1>
    allow-mailmerge yes
</source>
```

### 9.4 Creating Mailmerge Messages

#### 9.4.1 Message Template and Mailmerge Variables

Creating mailmerge messages is easy. The first task is to create a message template, marking and labeling the portions that you would like customized for each recipient, like in the example below:

```
From: shipping-confirmation@example.port25.com
Subject: Your order #[OrderNumber] shipped!

[First_Name],

Your order #[OrderNumber] was shipped on [ShipTime]. Thank you for your
business!
```

The labels for the customized portions are names of the variables which define what the content for each portion.

Once you have created your message template, you define the content for of these variables for the various recipients. For example, for one recipient you may want to have `First_Name` set to Bob, `OrderNumber` set to 1234-56 and `ShipTime` to 7/14 10:00. In the example above, this would yield:

```
From: shipping-confirmation@example.port25.com
Subject: Your order #1234-56 shipped!

Bob,

Your order #1234-56 was shipped on 7/14 10:00. Thank you for your
business!
```

Mailmerge variables are defined per recipient, and you cannot use the same variable name to define more than one set of data in the same message. Variable names are case-insensitive, e.g. `name` and `Name` are the same variable. Variable names may be composed of the letters (a-z and A-Z), the digits (0-9), and the underscore (`_`). The first character in a variable name must be a letter.

As noted above, variable substitutions are enclosed in square brackets in the message template. To include a square bracket itself, simply write it twice. For example,

```
[First_Name],

Please fill out the fields below marked with square brackets [[]].
...
```

would become (for a `First_Name` of "George")

```
George,

Please fill out the fields below marked with square brackets []
...
```

## 9.4.2 Reserved Variables

Variable names beginning with an asterisk are "reserved" and have a special meaning to PowerMTA:

### **\*from**

overrides the SMTP MAIL FROM (originator) address for the recipient's message.

### **\*envid**

overrides the DSN envelope ID for the recipient's message, for better tracking in bounces and in the accounting file. This setting overrides any other method of defining the envelope ID.

### **\*vmta**

selects the VirtualMTA (or VirtualMTA pool) through which the recipient's message should be delivered. This setting overrides any of the other VirtualMTA selection methods described in [Selecting a VirtualMTA or VirtualMTA Pool](#).

### **\*jobid**

defines the job (or campaign) ID for the recipient's message, for tracking in the WWW based monitoring facility. If set, the content of this variable overrides any job ID already set for the recipient's message. x-job header can be used with Mailmerge, but using \*jobid is faster.

### **\*parts**

defines the list of merge parts to include in the recipient's message. For example, if your message has 3 parts, numbered 1, 2 and 3 and you want to include all of them in the recipient message, you should set \*parts=1,2,3 or shorter, \*parts=1-3. If you wanted part 3 to go before part 2 in the recipient's message, you would set \*parts=1,3,2.

### **\*to**

merges in the defined RCPT TO/recipient email address. This variable is automatically set from the recipient address and cannot be overridden.

### **\*date**

merges in the message's reception date & time stamp. This variable is automatically set and cannot be overridden.

### **\*schedule**

sets the delivery schedule for the recipient in a message. This overrides any schedule that may be defined in the message body.

### **\*priority**

sets the priority of a recipient to the specified value. The value must be a number in the range 1-100, with higher values signifying higher priority for the recipient. The default priority for a recipient, if not set, is 50. If "\*priority" is used, it overrides the priority value, if any, obtained from <pattern-list> for the same recipient.

### **\*subaccountid**

sets the numeric subaccount ID to be included in SparkPost Signals events. If this variable is set, it overrides a subaccount determined by any configuration file directives.

### **\*custommessageid**

sets the custom message ID for SparkPost Signals events.

---

## 9.4.3 Merge Parts

Each message template is made up of one or more sections called *merge parts*. A merge part can be a sentence, a paragraph, a block of paragraphs, a body part, or even the message headers. Within each merge part there can be many mailmerge variable substitutions.

The parts or sections of the message are labelled with a part number, and then assembled for each recipient based upon the content of the special \*parts variable for the recipient.

For example, a travel alerts service could send messages in three parts: the first containing the headers and a common preamble,

```
From: "Customer Service" [*from]
To: "[FName]" [*to]
Date: [*date]
Subject: Travel Plans
```

Dear [FName],

You are flying out of [Airport] given that it is closest to your location in [City], [State].

the second containing a notice that is only sent to some of the recipients,

```
With this flight, you are only [MilesToBonus] frequent flyer miles away from
your next SuperBonus(TM)!
```

the third with a common final block:

```
Have a great trip!
```

Some recipients could then be sent all parts (\*parts=1-3) while others would only be sent parts #1 and #3 (\*parts=1,3). Note that the variable MilesToBonus would only need to be defined for those recipients receiving part #2.

## 9.5 Submitting Mailmerge Messages

You can feed mailmerge messages to PowerMTA using any of the standard supported submission APIs, pickup directory, or through PowerMTA's mailmerge SMTP extensions. It is recommend to add no more than 5000 recipients per mail merge file when feeding via pickup directory.

### 9.5.1 API Submission

All PowerMTA submission APIs include support for mailmerge. The basic process involves defining per-recipient variables by invoking the `defineVariable` for each Recipient object, and then passing the message template by invoking the `addMergeData` method for the `Message` object. Please refer to the corresponding API reference for more details.

Sample programs for using mailmerge in the APIs are provided with PowerMTA.

### 9.5.2 Mailmerge SMTP extensions

SparkPost has extended the SMTP protocol for sites that would prefer to feed PowerMTA mailmerge messages via a modified SMTP interface.

The mailmerge extensions define three new verbs:

#### 9.5.2.1 XMRG

Replaces the MAIL FROM command of a standard SMTP submission. All extensions (such as DSN, VERP, etc.) that are supported with MAIL are also supported with XMRG.

**Example:**

```
XMRG FROM:<travelAlerts@example.port25.com> VERP
```

#### 9.5.2.2 XDFN

Defines variables for the recipient following it. One or more XDFN commands should be issued prior to each RCPT command in a mailmerge submission. Variables are defined in the format `name=value`, where `name` is the variable name and `value` is the value to be associated with the variable. If the value contains any white space, it should be enclosed in double quotes. You can specify multiple variables in a single XDFN command by separating them with blanks.

**Example:**

```
XDFN name="Bob Example" Airport=BWI
XDFN *jobid="TravelAlerts"
RCPT TO:<bob@example.port25.com>
```

#### 9.5.2.3 XPRT

Replaces the DATA command of a standard SMTP submission. One XPRT command should be issued per merge part in the mailmerge submission, passing the merge part number, like in the example below:

```
XPRT 1
```

Like for DATA, PowerMTA responds to the XPRT command with a 3xx reply code. Also, like data, the message template must be passed dot-escaped. A dot on a line by itself terminates the merge part. The last XPRT is indicated with the additional LAST parameter and terminates the submission:

```
XPRT 2 LAST
```

The following example shows all the three verbs used to submit a mailmerge to two recipients. In addition to parts 1 and 3, Alice is also sent part #2 with a reminder that she's close to obtaining a bonus:

```
XMRG FROM:<travelAlerts@example.port25.com> VERP
XDFN FName="Alice" Airport="IAD" MilesToBonus="120"
XDFN City="Herndon" State="VA" *parts=1-3 *vmta="mta4"
XDFN *jobid="TravelAlerts"
RCPT TO:<alice@example.port25.com>
XDFN FName="Bob" Airport="BWI"
XDFN City="Ellicott City" State="MD" *parts=1,3 *vmta="mta5"
XDFN *jobid="TravelAlerts"
RCPT TO:<bob@example.port25.com>
XPRT 1
From: "Customer Service" [*from]
To: "[FName]" [*to]
Date: [*date]
Subject: Travel Plans

Dear [FName],

You are flying out of [Airport] given that it is closest to your location
in [City], [State].
.
XPRT 2

With this flight, you are only [MilesToBonus] frequent flyer miles away from
your next SuperBonus(TM)!
.
XPRT 3 LAST

Have a great trip!
.
```

### 9.5.3 Pickup Directory Submission

The pickup directory submission method includes support for mailmerge, which greatly minimizes the disk I/O impact usually associated with this submission method. Instead of creating and writing 100,000 messages, one per recipient, mailmerge support allows you to write say only 25-50 files for the same number of recipients, greatly minimizing the resources utilized and increasing throughput. See [Pickup Directory](#) for more details.

The pickup directory submission method is less efficient than submitting messages over SMTP.

Support for mailmerge in the pickup directory is accomplished through its support for BSMTP (batched SMTP) files. See [BSMTP files](#) for more information.

Here is a simple example of a properly formatted BSMTP mailmerge file for PowerMTA's pickup directory:

```
XMRG FROM:<bounce@port25.com>
XDFN *from="blahblah@port25.com"
XDFN subject="BSMTP mailmerge format"
RCPT TO:<bob@port25.com>
XPRT 1 LAST
From: [*from]
To: [*to]
Subject: [subject]
```

```
Hello [to]!  
.
```

Here is another that contains information for 4 recipients:

```
XACK ON  
XMRG FROM:<bounce@port25.com> verp  
XDFN FName="Jim" FavFood="pizza" Car="minivan" CID="1234"  
XDFN *vmta="mta1" *parts="1-3" *jobID="1234"  
RCPT TO:<jim@test.port25.com>  
XDFN FName="Tim" FavFood="sushi" Car="porsche" CID="2345"  
XDFN *vmta="mta2" *parts="1,3,2" *jobID="5678"  
RCPT TO:<tim@test.port25.com>  
XDFN FName="Cosmo" Car="Gremlin" CID="3456"  
XDFN *vmta="mta3" *parts="1,3" *jobID="1234"  
RCPT TO:<cosmo@test.port25.com>  
XDFN FName="Frank" FavFood="pizza" CID="4567"  
XDFN *vmta="mta5" *parts="1,2" *jobID="5678"  
RCPT TO:<frank@test.port25.com>  
XPRT 1  
X-CID: [CID]-abc  
From: "Customer Service" [*from]  
To: "[Fname]" [*to]  
Subject: Travel Plans  
Date: [*date]  
  
Hey [Fname],  
.  
XPRT 2  
Your favorite food is [favfood].  
.  
XPRT 3 LAST  
You drive a [car].  
.
```

[Mailmerge SMTP extensions](#) also provides another good example for reference.

Here is an example of a multi-part mailmerge file for a pickup directory:

```
XMRG FROM:<me@yourdomain.com> verp  
XDFN *from="me@yourdomain.com"  
XDFN subject="test \ 1"  
RCPT TO:<you@yourdomain.com>  
XDFN *from="me@yourdomain.com" subject="test 2"  
RCPT TO:<you@yourdomain.com>  
XPRT 1 LAST  
Mime-Version: 1.0  
Content-Type: multipart/alternative; boundary="12345"  
From: [*from]  
To: [*to]  
Subject: [subject]  
  
--12345  
Content-Type: text/plain;  
  
Hello. This is the text version.  
--12345  
Content-Type: text/html;  
  
<b>This is the HTML version.</b>  
  
--12345--  
.
```

### 9.5.4 The XACK SMTP extension

To facilitate and improve the feeding performance while using the mailmerge extensions, SparkPost has also defined the XACK protocol extension.

The XACK command disables acknowledgements for the RCPT and XDFN commands. It accepts a single parameter, either ON or OFF. To disable acknowledgements, you issue XACK like in the example below:

```
XACK OFF
```

Once acknowledgements are disabled, they stay disabled until the connection is closed or until they are re-enabled with XACK ON.

While acknowledgements are disabled, any errors occurring as a result of RCPT or XDFN command during a submission (i.e., after MAIL or XMRG and before the final DATA/BDAT/XPRT command) are reported as a response to the next DATA or BDAT (for non-mailmerge messages) or XPRT (for mailmerge messages). As a result of the error, the entire submission is canceled.

Some errors (e.g., address errors in the RCPT) may optionally be reported by sending a failure Delivery Status Notification, thus allowing the submission to continue. Any RCPT or XDFN errors (e.g., out of sequence errors) occurring not during a submission are ignored entirely.

The following example shows the connection trace (with commands and replies, but with the message content omitted) while submitting a mailmerge message:

```
>>> 220 mailhost.port25.com (PowerMTA v2.x) ESMTP service ready  
<<< EHLO test.port25.com  
>>> 250-mailhost.port25.com says hello  
>>> 250-ENHANCEDSTATUSCODES  
>>> 250-PIPELINING  
>>> 250-CHUNKING  
>>> 250-8BITMIME  
>>> 250-XACK  
>>> 250-XMRG  
>>> 250-SIZE 0  
>>> 250-VERP  
>>> 250 DSN  
<<< XACK OFF
```

```
>>> 250 2.0.0 ok
<<< XMRG FROM:<bounce@example.port25.com> verp
>>> 250 2.1.0 XMRG ok
<<< XDFN FName="Jim" FavFood="pizza" Car="minivan" CID="1234" *vmta="vmtapool" *parts="1-3"
<<< RCPT TO:<jim@example.port25.com>
<<< XDFN FName="Tim" FavFood="sushi" Car="porsche" CID="2345" *vmta="vmta2" *parts="1,3,2"
<<< RCPT TO:<tim@example.port25.com>
<<< XDFN FName="Cosmo" Car="Gremlin" CID="3456" *vmta="vmta3" *parts="1,3"
<<< RCPT TO:<cosmo@example.port25.com>
<<< XDFN FName="Frank" FavFood="pizza" CID="4567" *vmta="vmta1" *parts="1,2"
<<< RCPT TO:<frank@example.port25.com>
<<< XPRT 1
>>> 354 send part
<<< .
>>> 250 2.6.0 part ok
<<< XPRT 2
>>> 354 send part
<<< .
>>> 250 2.6.0 part ok
<<< XPRT 3 LAST
>>> 354 send part
<<< .
>>> 250 2.6.0 message received
<<< QUIT
>>> 221 2.0.0 mailhost.port25.com says goodbye
```

## 10 Advanced Features

### 10.1 SparkPost Signals

[SparkPost Signals](#) is an email analytics suite that helps you improve email performance by diagnosing health issues before they impact your business. It analyzes your email sending and benchmark data from across our industry-leading email network to provide data-driven, predictive, and actionable insights to help you increase open rates, prevent blacklisting, and boost inbox placement. See <https://www.sparkpost.com/docs/signals/overview/> for more information.

To utilize Signals, you'll need to enroll in the service and have the feature enabled. To begin your enrollment process please visit [Signals enrollment](#).

Once the feature is enabled you will need to create a SparkPost API key and configure your PowerMTA to send accounting data to SparkPost. To create a SparkPost API key with the Incoming Events: Write permission go to <https://app.sparkpost.com/account/api-keys>.

Note: For initial testing, the Signals servers need several thousand emails worth of data and then must batch-process the data, so there may be up to a day's delay before data is reflected in the reports.

Note: A restart of PowerMTA is recommended after the <signals> tag has been added to the [configuration](#) or any signals-related settings have been changed. While a reload is all that is required, a restart will ensure that all internal processes start working with the new settings immediately.

Note: For messages that are larger than 2.1GB in size, Sparkpost Signals will truncate and report the size to be 2.1GB. This may in turn affect the accuracy of the reports about message size in Signals (such as Avg Delivery Message Size on the Summary page etc.) if PowerMTA transmits messages larger than 2.1GB in size.

#### Note:

If PowerMTA connects to the external world through a firewall, you need to enable TCP connections originating from the machine that PowerMTA runs on to the hostname `api.sparkpost.com` port 443 through the firewall in order for PowerMTA to upload SparkPost Signals data. Please note that all communication from PowerMTA nodes to Sparkpost Signals is uni-directional only (i.e. PowerMTA is limited to pushing data and does not pull data from the Sparkpost Signals endpoint). More information about Sparkpost Signals API endpoint can be found online at <https://developers.sparkpost.com/api/>.

#### 10.1.1 Optional Data Loading Support

You can optionally designate the virtual MTAs that are allowed to send Signals data to SparkPost. You can also optionally designate the sending domain (this would be the domain portion of the Sender or From header within the body of the message - see [Sample Use Cases](#) below for more details) whose events will be sent over to SparkPost.

PowerMTA offers the use of [enable-signals](#) at these scopes:

1. Global scope: used to enable/disable generation of Signals events globally.
2. VMTA pool scope: used to enable/disable generation of Signals events for all the virtual MTAs within the same pool.
3. VMTA scope: used to enable/disable generation of Signals events for a single VMTA.

#### 10.1.2 SparkPost Subaccounts

You can configure PowerMTA to send Signals data to a specific SparkPost subaccount. In order to use a subaccount ID in PowerMTA it must first be created on SparkPost. To manage subaccounts on SparkPost, go to <https://app.sparkpost.com/account/subaccounts>

If you configure PowerMTA with a subaccount ID that does not exist on SparkPost, you may see subaccount names such as these in the Signals reports: "Deleted (ID x)", where x is the configured subaccount ID. In such a case:

- if the data should not be in the subaccount then, update the subaccount ID in PowerMTA configuration
- if the data should be in the stated subaccount, then create the subaccount ID on SparkPost

See the [enable-signals](#), [sender-or-from](#), and [process-x-sp-subaccount-id](#) directives for information on how to specify the subaccount ID for a message.

#### 10.1.3 Sample Use Cases for Optional Loading of Signals Data

##### 10.1.3.1 Use Case #1. Disable Signals completely.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals false
```

**Result** No signals data is sent at all from this PowerMTA.

**Explanation** The global directive [enable-signals](#) is set to false, and is inherited by all VMTAs in the system.

#### 10.1.3.2 Use Case #2. Use Signals to track every email delivered out of PowerMTA.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals true

<virtual-mta mktg>
</virtual-mta>

<virtual-mta transact>
</virtual-mta>
```

**Result** Signals data for all VMTAs, including {default}, mktg and transact, is sent from this PowerMTA to SparkPost.

Explanation: The global directive [enable-signals](#) is set to true and is inherited by all VMTAs in the system. So, PowerMTA sends event data for all VMTAs to SparkPost.

#### 10.1.3.3 Use Case #3. Dedicated VMTAs per message stream.

I am an ESP or an enterprise business with multiple customers/message streams. I have mapped my customers or message streams to individual VMTAs. I don't mix different message streams within a single VMTA. I want to selectively enable Signals for only some of my message streams but I want to disable Signals for all other customers.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals false

<virtual-mta mktg>
  enable-signals true
</virtual-mta>

<virtual-mta transact>
</virtual-mta>
```

**Result** Signals data for all events in mktg virtual MTA is sent from this PowerMTA to SparkPost. No data is sent for any other virtual MTA in the system (all other VMTAs, including transact VMTA, inherit enable-signals false from global settings)

#### 10.1.3.4 Use Case #4. Dedicated VMTAs per message stream, with subaccounts.

I am an ESP or an enterprise business with multiple customers/message streams. I have mapped my customers or message streams to individual VMTAs. I don't mix different message streams within a single VMTA. I want to selectively enable Signals for only some of my message streams (and assign them a subaccount ID) but I want to disable Signals for all other customers.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals false

<virtual-mta mktg>
  enable-signals true,subaccount=100
</virtual-mta>

<virtual-mta sales>
  enable-signals true,subaccount=200
</virtual-mta>

<virtual-mta transact>
</virtual-mta>
```

**Result** Signals data for all events in mktg and sales virtual MTAs are sent from this PowerMTA to SparkPost. Each event will have the subaccount ID configured for that VMTA, (mktg events will have subaccount ID 100, while sales VMTA events will have subaccount ID set to 999). No data is sent for any other virtual MTA in the system (all other VMTAs, including transact VMTA, inherit enable-signals false from global settings)

#### 10.1.3.5 Use Case #5. Use of VMTA pools.

I am an ESP or an enterprise business with multiple customers/message streams. I have mapped my customers or message streams to individual VMTAs. Some of my large message streams have been assigned multiple VMTAs, all of which are in a pool. I want to enable Signals for a large message stream using a pool, but keep Signals disabled for all other message streams.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals false

<virtual-mta v1>
</virtual-mta>

<virtual-mta v2>
</virtual-mta>

<virtual-mta transact>
</virtual-mta>

<virtual-mta-pool pond>
  virtual-mta v1
  virtual-mta v2

  enable-signals true
</virtual-mta-pool>
```

**Result** Signals data for all events in v1 and v2 virtual MTAs (because they inherit `enable-signals true` from pond virtual MTA pool) are sent from this PowerMTA to SparkPost. No data is sent for any other virtual MTA in the system (all other VMTAs, including `transact VMTA`, inherit `enable-signals false` from global settings). This data can also be configured to have a subaccount ID.

#### 10.1.3.6 Use Case #6. Use of shared VMTAs.

I am an ESP or an enterprise business with multiple customers/message streams that mixes multiple message streams within a single shared VMTA. I need to enable Signals for only some of the mail streams that go out from the shared VMTA but not all. Each mail stream uses a unique `sender-or-from` domain.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals false

<signals-filter ListOne>

  sender-or-from mktg.customer1.com          enable-signals,subaccount=100

  sender-or-from sales.customer1.com          enable-signals,subaccount=100

  sender-or-from sales.customer2.com          enable-signals,subaccount=999

</signals-filter>

<virtual-mta v1>
  enable-signals ListOne
</virtual-mta>

<virtual-mta transact>
</virtual-mta>
```

**Result** Signals data for all events with `mktg.customer1.com` or `sales.customer1.com` as their `sender-or-from` domain are sent from this PowerMTA to SparkPost with subaccount ID set to 100. Similarly, data for all events with `sales.customer2.com` as their `sender-or-from` domain are sent from this PowerMTA to SparkPost with subaccount ID set to 999. No data is sent for any other `sender-or-from` domain that is used for sending within this virtual MTA. No data is sent for any other `sender-or-from` domain within v1 virtual MTA, or any other virtual MTAs in the system (all other VMTAs, including `transact VMTA`, inherit `enable-signals false` from global settings)

#### 10.1.3.7 Use Case #7: Use of shared VMTAs (wildcard support)

I am an ESP or an enterprise business with customers/multiple message streams that mixes multiple message streams within a single shared VMTA. I need to enable Signals for all mail streams that go out from the shared VMTA with a common subdomain (`*.customer1.com`) as their `sender-or-from` address. I also want to disable Signals for some mail streams that use a specific `sender-or-from` domain from within that shared VMTA.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals false

<signals-filter ListOne>

  sender-or-from *.customer1.com          enable-signals,subaccount=100

  sender-or-from transact.customer1.com    disable-signals

</signals-filter>

<virtual-mta v1>
  enable-signals ListOne
</virtual-mta>

<virtual-mta transact>
</virtual-mta>
```

**Result** Signals data for all events with `.customer1.com` as their `sender-or-from` domain suffix are sent from this PowerMTA to SparkPost with subaccount ID set to 100. The only exception is that events with `sender-or-from` domain set to `transact.customer1.com` within this virtual MTA are not sent to SparkPost. No data is sent for any other virtual MTA in the system (all other VMTAs, including `transact VMTA`, inherit `enable-signals false` from global settings)

#### 10.1.3.8 Use Case #8: Global level configuration.

I want a single, global location where I can enable/disable Signals for all the `sender-or-from` domains I use to mail from, irrespective of the virtual MTA that the email goes out of.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals ListOne

<signals-filter ListOne>

  sender-or-from *.foo.com          enable-signals,subaccount=100

  sender-or-from transact.foo.com    disable-signals

</signals-filter>

<virtual-mta v1>
</virtual-mta>

<virtual-mta v2>
</virtual-mta>
```

**Result** Signals data for all events in the system, irrespective of the VMTA they were generated from, with `.foo.com` as their `sender-or-from` domain suffix are sent from this PowerMTA to SparkPost with subaccount ID set to 100. The only exception is that events with `sender-or-from` domain set to `transact.foo.com` are not sent to SparkPost.

### 10.1.3.9 Use Case #9: Global level configuration, example #2.

I want to enable Signals for all Virtual MTAs in the system, but I don't want Signals enabled for the VMTAs used for transactional and personal email streams. I also want to be able to configure the specific sender-or-from domains that Signals is enabled for, and keep Signals disabled for the rest of the domains.

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>

enable-signals ListOne

<signals-filter ListOne>

  sender-or-from mail.customer1.com          enable-signals,subaccount=100
  sender-or-from sales.customer1.com         enable-signals,subaccount=200
  sender-or-from mktg.customer1.com          enable-signals,subaccount=300
  sender-or-from foo.customer1.com           enable-signals,subaccount=400
  sender-or-from bar.customer1.com           enable-signals,subaccount=500
  sender-or-from baz.customer1.com           enable-signals,subaccount=600

  sender-or-from *                           disable-signals
</signals-filter>

<virtual-mta v1>
</virtual-mta>

<virtual-mta v2>
</virtual-mta>

<virtual-mta transact>
  enable-signals false
</virtual-mta>

<virtual-mta personal>
  enable-signals false
</virtual-mta>
```

**Result** Signals data for all events in the system with any of {mail|sales|mktg|foo|bar|baz}.customer1.com as their sender-or-from domain suffix are sent from this PowerMTA to SparkPost along with their respective subaccount IDs.

Events belonging to personal or transact VMTAs, or having any other sender-or-from domain other than the ones listed above are not sent to SparkPost.

### 10.1.4 Configuration Directives

Signals configuration is done within the <signals> tag. Note that there is no need to designate which records and record fields to send to SparkPost - all the record types and fields needed for Signals are automatically configured by PowerMTA.

Note: It's recommended to restart PowerMTA after the <signals> tag has been added to the configuration or any signals-related settings have been changed. While a reload is all that is required, a restart will ensure that all internal processes start working with the new settings immediately.

You also have the option of configuring PowerMTA to only transmit data for designated virtual MTAs, virtual MTA pools or sending domains to Signals. See [enable-signals](#) directive and [Optional Data Loading Support](#) section for more details.

In prior versions of PowerMTA, <signals> tag was called <sparkpost-analytics>, which is deprecated. Please use signals instead.

The available internal directives are as follows.

#### 10.1.4.1 Global Directives

##### 10.1.4.1.1 enable-signals

This directive can be used at global, VMTA pool or VMTA scopes to enable/disable generation of Signals events at that scope level. When this directive is set to true, it can also be optionally suffixed with the [subaccount](#) information in Signals for the events generated within that scope.

If this directive is unspecified at global level, then it defaults to true if a <signals> tag has been added to PowerMTA configuration. This is in order to maintain compatibility with how the feature was enabled in previous versions.

Normal rules for inheritance apply to this directive. A virtual MTA inherits this setting from both pool and global scopes, with the pool scope taking precedence over the global scope.

#### Syntax:

```
enable-signals {true[,subaccount=N] | false | <signals-filter-name>}
```

where N is a [subaccount](#) ID obtained from SparkPost.

Note that subaccount ID can only be specified when enable-signals is set to true. It cannot be specified when enable-signals is false or set alongside a <signals-filter> list.

**Default:** true at global level if <signals> has been configured; false otherwise

#### Example #1:

```
# at global scope, inherited by all VMTAs in the system.
enable-signals true,subaccount=555

# at pool scope, inherited by all VMTAs in the pool
<virtual-mta-pool pond>
  enable-signals true,subaccount=999

  virtual-mta v1
  virtual-mta v2
  virtual-mta v3
  virtual-mta v4
</virtual-mta-pool>
```

```
# inherits pool setting
<virtual-mta v1>
</virtual-mta>

# overrides all inheritance
<virtual-mta v2>
  enable-signals true,subaccount=888
</virtual-mta>

# inherits pool setting
<virtual-mta v3>
</virtual-mta>

# overrides all inheritance
<virtual-mta v4>
  enable-signals false
</virtual-mta>

# inherits global setting
<virtual-mta v5>
</virtual-mta>
```

#### Example #2:

```
<signals>
  api-key 0123456789012345678901234567890123456789
</signals>
```

In the above example, enable-signals defaults to true since its not explicitly set at global level but a signals tag is present in the configuration.

#### 10.1.4.2 <signals> Directives

##### 10.1.4.2.1 api-key

PowerMTA authenticates itself to the Signals Endpoint using an API key when uploading accounting data. This directive can be used to specify the API key (obtained from SparkPost) that authorizes this PowerMTA to the Signals server.

This directive is required, and PowerMTA will not start without an api-key directive defined inside <signals> tag.

See [Sparkpost API Overview](#) for details about authentication when using Sparkpost API. Please contact Sparkpost Signals support for help on creating an API key.

Must be composed of exactly 40 hexadecimal digits (0-9 A-F a-f).

#### Example:

```
<signals>
  api-key 0123456789abcdef012345678901234567890123
</signals>
```

##### 10.1.4.2.2 upload-url

Specifies the SparkPost Signals server URL where the accounting data is sent.

#### Default:

```
https://api.sparkpost.com/api/v1/ingest/events
```

#### Example:

```
<signals>
  api-key 0123456789012345678901234567890123456789
  upload-url https://api.sparkpost.com/api/v1/ingest/events
</signals>
```

##### 10.1.4.2.3 engagement-tracking

This directive must be set to true for Signals Health Score to be calculated. Note that transmission of engagement events such as open and click events is done outside of PowerMTA. Contact SparkPost Support for more help in getting Signals Health Score when using PowerMTA.

Note that this directive needs to be set to true only if you plan to send engagement events such as open and click events to SparkPost Signals. If you don't intend to use SparkPost Signals to track engagement events, this directive should be set to false.

#### Default: false

#### Example:

```
<signals>
  api-key 0123456789012345678901234567890123456789
  engagement-tracking true
</signals>
```

##### 10.1.4.2.4 custom-message-id-header

Allows setting the name of a header that will be used to set the custom message ID in Signals events. This field is used by Signals to correlate open and click events with other Signals events for a message. This correlation is important for getting the full benefits of the Signals analytics suite. The ID specified by the header can be up to 512 ASCII characters. If using Mailmerge, you must use the \*custommessageid variable, not the header.

Sample header:

```
x-my-msgid: 01234.abc@mysite
```

#### Default: none

#### Example:

```
<signals>
  api-key 0123456789012345678901234567890123456789
  custom-message-id-header x-my-msgid
</signals>
```

#### 10.1.4.2.5 process-x-sp-subaccount-id

Specifies whether PowerMTA should process an `x-sp-subaccount-id` header. If set to true and one such header is included, PowerMTA will use the value to set the subaccount ID in Signals events. If set to false, PowerMTA will ignore this header, leaving it in the message if present. The subaccount ID must be an unsigned integer greater than zero. If using Mailmerge, you must use the `*subaccountid` variable, not this header. If the `*subaccountid` variable is set, or this header is enabled and present in the message, a subaccount determined by a configuration file directive will be ignored.

Sample header:

```
x-sp-subaccount-id: 123
```

**Default:** false

**Example:**

```
<signals>
  api-key 0123456789012345678901234567890123456789
  process-x-sp-subaccount-id true
</signals>
```

#### 10.1.4.2.6 retain-x-sp-subaccount-id

Specifies whether PowerMTA should retain a processed `x-sp-subaccount-id` header. If set to true or `process-x-sp-subaccount-id` is false, PowerMTA will keep the header in the message when sending it on.

If `retain-x-sp-subaccount-id` is set to true, then you have the option of writing the content of this header to accounting files as part of `[d|b|t]` records. This is especially useful if you are using Sparkpost Ingest API to manually submit engagement events to Sparkpost.

**Default:** false

**Example:**

```
<signals>
  api-key 0123456789012345678901234567890123456789
  process-x-sp-subaccount-id true
  retain-x-sp-subaccount-id true
</signals>

<acct-file /var/log/pmta/acct.csv>
  records d
  record-fields d *,header_x-sp-subaccount-id
</acct-file>
```

#### 10.1.4.2.7 log-verbose

Causes all upload activity to be logged. If not set, upload activity is summarized and logged at regular intervals. Enable this setting to help diagnose upload problems.

**Default:** false

**Example:**

```
<signals>
  api-key 0123456789012345678901234567890123456789
  log-verbose true
</signals>
```

#### 10.1.4.2.8 min-free-space

Instructs PowerMTA to start deleting SparkPost accounting data files if the space available on the file system falls below the amount specified. Note that these files are normally deleted once the contents have been sent to the SparkPost Signals server, so space should only run low if there is a problem such as the Signals server being down or too busy or an incorrect configuration. Once these files are deleted, that data is lost forever, and there will be "holes" in the Signals reports.

On Unix, this refers to the space available to non-root users in `/var/lib/pmta/sparkpost`.

**Syntax:**

```
min-free-space { N{b|k|m|g|t} | 0 }
```

where N is a positive integer followed by a size multiplier (B for byte, K for Kilobyte, M for Megabyte, G for Gigabyte or T for Terabyte).

**Default:** 20M

**Example:**

```
<signals>
  api-key 0123456789012345678901234567890123456789
  min-free-space 1G
</signals>
```

#### 10.1.4.2.9 privacy-policy

This directive can be used to tell PowerMTA to hash (with the one-way SHA-1 hash) the lower-cased local parts of recipient address in the data sent to the Signals server.

**Syntax:**

```
privacy-policy { retain | sha1 }
```

**Default:** retain

**Example:**

```
<signals>
  api-key 0123456789012345678901234567890123456789
  privacy-policy sha1
</signals>
```

### 10.1.4.3 <signals-filter> Directives

#### 10.1.4.3.1 sender-or-from

Another way to designate the specific events for Signals is to use the domain portion of Sender or From headers within the content of the email to filter. This permits filtering independent of the virtual MTA that the email was received or delivered out of.

**Syntax:**

```
<signals-filter NAME>
  sender-or-from <domain-pattern> { enable-signals[,subaccount=N] | disable-signals}
</signals-filter>
```

where N is a [subaccount](#) ID obtained from SparkPost.

Note that subaccount ID can only be specified when enable-signals is set. It cannot be specified when disable-signals is set.

**Example:**

```
# at global scope, inherited by all VMTAs in the system.
<signals-filter ListOne>

  sender-or-from marketing.customer1.com enable-signals,subaccount=555
  sender-or-from transact.customer1.com disable-signals
  sender-or-from *.customer1.com enable-signals,subaccount=333

  sender-or-from * enable-signals,subaccount=222
</signals-filter>

<signals-filter ListTwo>

  sender-or-from sales.customer2.com enable-signals,subaccount=999
</signals-filter>

<signals-filter ListThree>

  sender-or-from * enable-signals,subaccount=777
</signals-filter>

enable-signals ListOne

# at pool scope, inherited by all VMTAs in the pool
<virtual-mta-pool pond>
  enable-signals ListTwo

  virtual-mta v1
  virtual-mta v2
  virtual-mta v3
  virtual-mta v4
</virtual-mta-pool>

# inherits pool setting
<virtual-mta v1>
</virtual-mta>

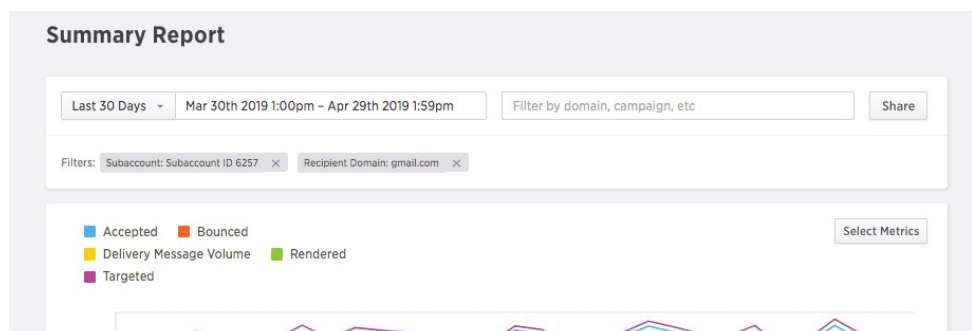
# overrides all inheritance
<virtual-mta v2>
  enable-signals ListThree
</virtual-mta>

# inherits pool setting
<virtual-mta v3>
</virtual-mta>

# overrides all inheritance
<virtual-mta v4>
  enable-signals false
</virtual-mta>

# inherits global setting
<virtual-mta v5>
</virtual-mta>
```

### 10.1.5 Summary Report Display





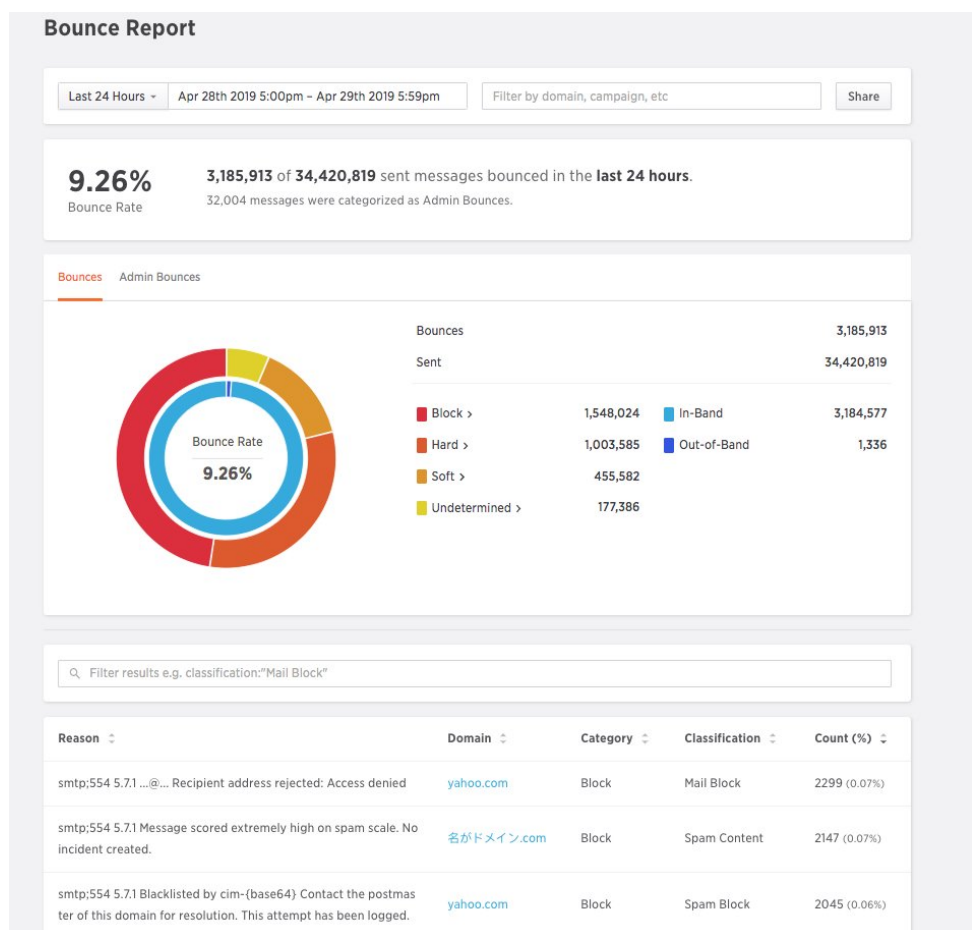
The Summary report provides you the ability to view a variety of [metrics](#), adjust reporting date range, and filter by campaign (jobID), recipient domain, or other metrics by clicking on the desired value in the "Group By" listing.

The following "Group By" options are available for viewing detailed data:

- **Recipient Domain**
- **Sending Domain** - the domain portion of the "Sender" or "From" header
- **Campaign** - the jobID
- **Sending IP**
- **IP Pool** - the VirtualMTA pool

For more information, see <https://www.sparkpost.com/docs/reporting/reporting-and-analytics/>.

### 10.1.6 Bounce Report Display

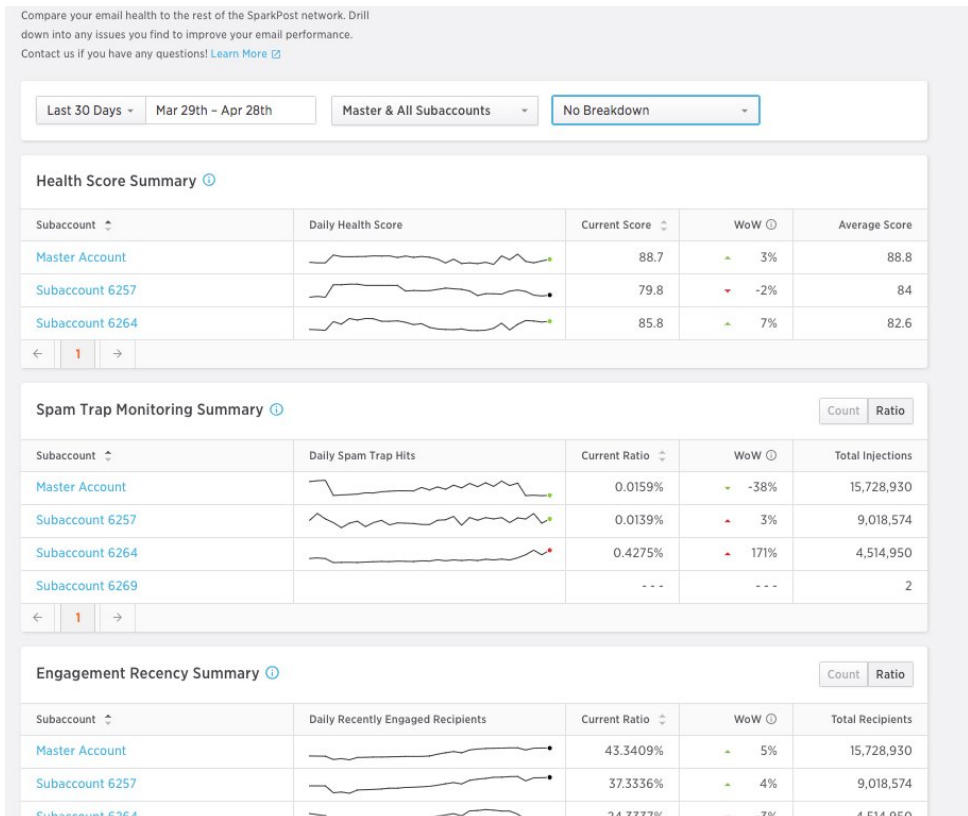


The Bounce report provides detailed information about the bounces during the selected time period. Note that the bounces have been classified according to [SparkPost bounce classification codes](#) and will not correspond directly to PMTA bounce categories.

See <https://www.sparkpost.com/docs/reporting/bounces/> for more information.

### 10.1.7 Signals Overview Display

#### Signals Overview

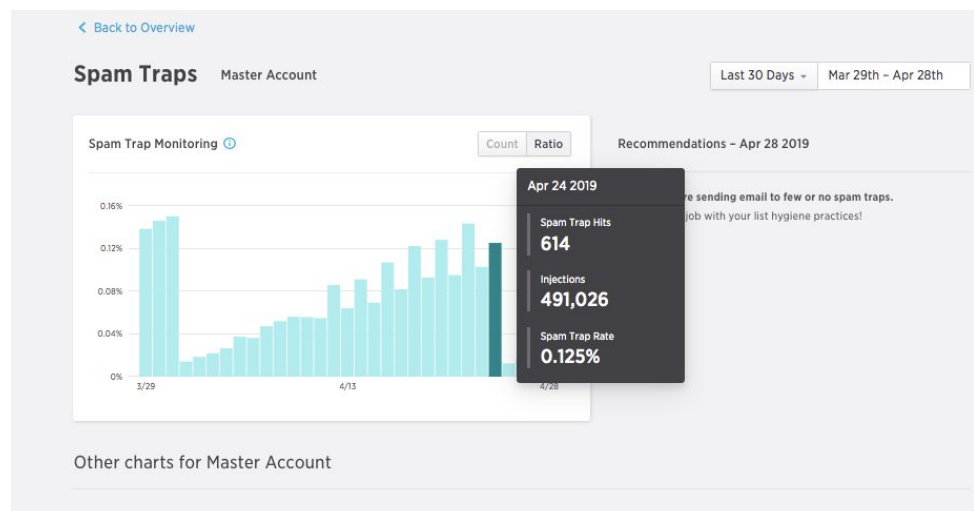


The Signals Overview contains leaderboards for the three signals provided: Health Score, Spam Trap Monitoring, and Engagement Recency. The Engagement Recency Summary is not yet supported in PMTA.

Each row in these tables contains the summary for a different subaccount, and can be further sliced by campaign (jobID), IP pool (VirtualMTA pool), or sending domain to help catch issues across any of these facets. Additionally, you can view the data for up to 90 days to analyze trends over time.

See <https://www.sparkpost.com/docs/user-guide/signals/> for more information.

### 10.1.8 Spam Traps Display



With visibility across multiple commercial and non-commercial spam trap networks, SparkPost Signals is a crucial resource for maintaining the health of your email. Spam Trap Monitoring reports the share over time of your total email volume that was sent to known spam traps for each given day. This indicator is important to take note of since an excessive rate of spam trap hits indicates poor list procurement and hygiene practices. You can view the spam trap hits as a raw number or a ratio over the number of injections for that day.

See <https://www.sparkpost.com/docs/user-guide/signals/#spam-trap-monitoring> for more information.

## 10.2 HAProxy Protocol for outbound traffic

For customers that own multiple source IPs distributed across many PowerMTA installations, we offer support for proxy protocol that helps make administration easier. With this feature, you can now deploy any number of PowerMTAs within your network to use internal IPs. All external facing source IPs can be deployed on a single proxy machine. The internal PowerMTA machines will not connect to the remote MTAs (like gmail.com, yahoo.com) directly, but instead connect through the proxy. The proxy protocol support in PowerMTA will let the proxy server know which external facing source IP to use when connecting to the remote IPs.

PowerMTA can be configured to use a proxy server that supports HAProxy protocol version 1 (see <https://www.haproxy.org/download/1.8/doc/proxy-protocol.txt> for more details about the protocol) for delivering email to the outside world. When PowerMTA is configured to use a proxy for outbound traffic, PowerMTA will perform DNS resolution of the destination domain as usual, but when delivering email, it will connect to the configured proxy server instead of directly connecting to the remote destination MTA. PowerMTA will transmit the client and destination IPs that it wants to connect to as part of the proxy header. It is the responsibility of the proxy server to interpret the protocol header to connect to the destination IP address using the requested client IP

address.

When a proxy is configured, for deliveries and bounces, PowerMTA will report the client IP as the source IP address("divSourceIp" in CSV accounting file) and the destination MTA's IP as the destination IP ("divDestinationIp" in the CSV accounting file). The fact that it used a proxy to deliver the email is transparent when looking at the data in accounting files.

Note that proxy protocol support is only for outbound traffic from PowerMTA to remote MTAs. We will be extending PowerMTA to support proxy protocol for traffic inbound into PowerMTA in subsequent releases.

See [<proxy> Directives](#) for more information on directives.

### Example

In the configuration below, PowerMTA is configured to use 2 proxy servers:

1. **mta-proxy-a** This proxy server listens on IP 10.1.2.3, port 5000 for incoming connections. When making outgoing connections, this proxy will use the external facing IP 16.70.55.233 (used here for illustration purpose only) and the hostname `client.hostname.com`.
2. **mta-proxy-b** This proxy server listens on hostname `mta-proxy-b.internal.hostname.com`, port 5000 for incoming connections. When making outgoing connections, this proxy will use the external facing IP 16.80.77.192 (used here for illustration purpose only) and the hostname `client2.hostname.com`.

There is one virtual MTA on PowerMTA.

1. **vmta** This virtual MTA is configured to use an internal IP 10.8.9.10 (with the hostname `vmta.hostname.com`) to connect to the proxy nodes. The virtual MTA `vmta` is configured to use the 2 proxy nodes defined above. The second proxy node is used only when the connection to the first proxy node fails for any reason.

```
<proxy mta-proxy-a>
# the proxy server listens on IP 10.1.2.3, port 5000 for incoming connections
server 10.1.2.3:5000
# the client IP tells the proxy what source IP to use when connecting to the destination host
client 16.70.55.233:0 client.hostname.com
</proxy>

# You can configure multiple proxies, each with its own name, proxy server IP and client IP configuration
<proxy mta-proxy-b>
# the proxy server listens on mta-proxy-b.internal.hostname.com, port 5000 for incoming connections
server mta-proxy-b.internal.hostname.com:5000

# the client IP tells the proxy what source IP to use when connecting to the destination host
# PowerMTA only checks for syntactic validity of the client IP address, and it cannot verify that
# the client IP address has been configured on the proxy server.
client 16.80.77.192:0 client2.hostname.com
</proxy>

<virtual-mta vmta>
# The "smtp-source-host" is used to configure the source IP that PowerMTA will connect to the proxy
server from
# The IP address must be configured on the machine on which PowerMTA runs
# The IP address is completely independent of the client IP address that the proxy will use to connect
to the destination
smtp-source-host 10.8.9.10 vmta.hostname.com

# "use-proxy" directive is used to configure the proxy server(s) that PowerMTA will use when connecting
to destination MTAs.
# This directive takes a comma separated list of proxy server names.
# PowerMTA will connect to proxies strictly in the order defined - PowerMTA will only use the second
proxy if the connection
# to the first proxy fails, and similarly for the third.
# PowerMTA does NOT roundrobin or load balance its outgoing connections between the configured proxy
servers.
# Configuring multiple proxies is strictly for helping with failover.

use-proxy mta-proxy-a,mta-proxy-b # "mta-proxy-a" is always used unless connections to it fail. "mta-
proxy-b" is used only when connecting to "mta-proxy-a" fails

</virtual-mta>
```

## 10.3 Sender Reputation Monitoring

### 10.3.1 Overview

One of the keys to maximizing deliverability to valid domains is to know as soon as possible when specific delivery problems occur. The sooner one realizes that there are certain problems, or more specifically, that a remote site is rejecting messages for one reason or another, the sooner one can attempt to get the problems resolved. There are many different kinds of errors that remote sites can respond with when connected, both positive and negative, and these responses allow you to see your current sender reputation in the eyes of that recipient domain or mail gateway.

Given the importance of having this information as soon as possible, PowerMTA includes automated sender reputation monitoring, via an advanced, real time SMTP command monitoring implementation. Not only does it give you an early warning alert system for specific types of errors, but it can also be configured to try and possibly resolve some problems as well, via support for automated configuration changes for future delivery attempts.

### 10.3.2 Implementation

PowerMTA groups messages for a particular domain for a particular VirtualMTA in its own queue. Sender reputation monitoring provides real time monitoring of defined SMTP responses (e.g., "551 Sender IP rejected") from remote hosts for each queue, with the ability to define certain actions (e.g., send an email alert while throttling deliveries, lower the rate at which messages are delivered etc.) upon a match.

This is accomplished through the introduction of queue "modes". The two modes initially supported are "normal" and "backoff". You can configure PowerMTA so that when a queue enters backoff mode, it

- sends an email alert to one or more addresses (`backoff-notify` directive).
- dynamically changes the retry schedule for that queue (`backoff-retry-after` directive).
- dynamically applies per-queue delivery throttling (`backoff-max-msg-rate` directive).
- dynamically reroutes the messages from one VirtualMTA to another VirtualMTA (`backoff-reroute-to-virtual-mta` directive).

directive).

The dynamic reroute, combined with the ability to have separate per domain configurations per VirtualMTA, allows one to use the full range of domain directives beyond the new `backoff-...` directives (e.g., using `bounce-after`), as well as VirtualMTA attributes, for the fallback queue. A chain of backoff VirtualMTAs is also supported (`vmta -> vmta1, vmta1 -> vmta2, vmta2 -> vmta3`, etc.).

SMTP response matching is performed based on a list of regular expressions, entered in an `smtp-pattern-list` directive in the configuration file.

For example, you could define a list of errors that might occur during deliveries to AOL in an `smtp-pattern-list` named `aol-errors`, and specify that the queue be put in backoff mode if any such match:

```
<smtp-pattern-list aol-errors>
  reply /generating high volumes of.* complaints from AOL/ mode=backoff
  reply /Excessive unknown recipients - possible Open Relay/ mode=backoff
  reply /^421 .* too many errors/ mode=backoff
  reply /blocked.*spamhaus/ mode=backoff
  reply /451 Rejected/ mode=backoff
</smtp-pattern-list>
```

You would then indicate that PowerMTA should match based on that pattern list while delivering to `aol.com`:

```
<domain aol.com>
...
  smtp-pattern-list aol-errors
...
</domain>
```

Finally, also within that domain definition, you would specify certain backoff-... actions to be taken when the queue enters backoff mode:

```
<domain aol.com>
...
  smtp-pattern-list aol-errors # SMTP response pattern list to use
  backoff-max-msg-rate 120/h # 10 messages per 5 min interval
  backoff-retry-after 1h # retry every hour
  backoff-notify postmaster@yourdomain.com
...
</domain>
```

Upon the first match of the SMTP response, the queue for `aol.com` for this particular VirtualMTA will enter backoff mode, dynamically applying the specific backoff-... directives:

- send an email alert to the defined PowerMTA postmaster address.
- change the retry interval to `aol.com` for this particular VirtualMTA to 1 hour.
- throttle the delivery rates to `aol.com` for this particular VirtualMTA to at most 120 attempted deliveries per hour.

The backoff directives apply until the mode for the queue is returned to normal, which can be accomplished either within an `smtp-pattern-list` (by specifying a `mode=normal` for a pattern), or with the `pmta set queue` command, or by restarting PowerMTA. In the example above, the command to return that queue to normal mode would be

```
pmta set queue --mode=normal aol.com
```

Defining backoff VirtualMTAs is also relatively straightforward, using the `backoff-reroute-to-virtual-mta` directive. It simply specifies that PowerMTA should reroute messages to the given VirtualMTA in case the queue enters backoff mode.

For example, if messages are queued for example `port25.com/vmta1` and that queue enters backoff mode while `backoff-reroute-to-virtual-mta` is set to `vmta2`, all of its messages would be rerouted to the queue `example.port25.com/vmta2`. Since the messages would then be queued within `vmta2`, that VirtualMTA's settings would apply. New messages entering the queue are also immediately rerouted to the new VirtualMTA, as long as the `vmta1` queue remains in backoff mode.

For example, one could define a backoff VirtualMTA as follows:

```
<virtual-mta mta1>
  smtp-source-host 1.1.1.1 mta1.port25.com

  <domain *>
    backoff-reroute-to-virtual-mta backoff-mta1
    smtp-pattern-list block-errors
  </domain>
</virtual-mta>

<virtual-mta backoff-mta1>
  smtp-source-host 1.1.1.2 mta2.port25.com

  <domain *>
    retry-after 30m
    bounce-after 1h
  </domain>
</virtual-mta>
```

With this, if there is an SMTP pattern match for any domain for VirtualMTA `mta1`, any messages in that specific queue will be delivered using the domain directives and VirtualMTA parameters defined in VirtualMTA `backoff-mta1`. As desired, the backoff parameters will *only* apply to messages for the specific queue, that is, for messages in the queue whose delivery attempts received the matched SMTP response.

Note that since the messages are really transferred from one VirtualMTA to another, the `backoff-reroute-to-virtual-mta` directive effectively takes precedence over `backoff-max-msg-rate` and `backoff-retry-after`.

### 10.3.3 Supporting commands

There are two commands that can be used for sender reputation monitoring and queue modes:

- `pmta set queue --mode={normal|backoff} domain/vmta`
- `pmta show queues`

The `set queue` command sets the given queue's mode of operation to either normal or backoff. See [set queue](#) for more information.

The `show queues` command allows you to view current information on the specified queues, including the current mode of the queue. See [show queues](#) for more information.

## 10.4 SMTP AUTH support

### 10.4.1 Overview

To facilitate creating a secure email delivery channel, PowerMTA supports authentication in both inbound and outbound SMTP connections.

### 10.4.2 Implementation for Inbound Connections

PowerMTA supports authentication with either LOGIN, PLAIN or the CRAM-MD5 mechanism in inbound connections. Support for CRAM-MD5 is available whenever AUTH itself is enabled. The password is never sent when using CRAM-MD5 as it's a challenge/response authentication.

To use AUTH you must also create an `<smtp-user>` tag for each valid username, including their passwords, as well as the named source to use when that user authenticates, like in the example below:

```
<smtp-user jsmith>
  password qwerty123
  source auth2
</smtp-user>

<smtp-user jdoe>
  password abc123
  source auth1
</smtp-user>
```

For maximum flexibility in supporting the various SASL authentication methods, the passwords must be entered in plaintext in the configuration file. Because of this, it is crucial that the PowerMTA config file permissions are set up as restrictively as possible.

The named source in the configuration file (that is a `<source>` entry that has a name rather than an IP address or CIDR specification), allows you to specify which `<source>` settings should apply on a per-user basis.

In the example above, and with the configuration file below, user `jsmith` would be allowed access to PowerMTA's mail merge extensions, while `jdoe` would not:

```
<source auth2>
  allow-mailmerge yes
  always-allow-relaying yes # allow feeding for auth2
  process-x-virtual-mta yes # allow selection of a VirtualMTA
  max-message-size unlimited
  smtp-service yes          # allow SMTP service
  require-auth true
</source>

<source auth1>
  allow-mailmerge no
  always-allow-relaying yes # allow feeding for auth1
  process-x-virtual-mta yes # allow selection of a VirtualMTA
  max-message-size unlimited
  smtp-service yes          # allow SMTP service
  require-auth true
</source>
```

When users `jsmith` and `jdoe` authenticate, any settings in their named sources would override those settings normally obtained based on each connections' source IP address. This way, you can combine the usual IP-based settings (which apply to all connections) with specific settings based on the authenticated user.

Certificate based authentication is also supported with a configuration like the following:

```
<smtp-user jsmith>
  authentication-method certificate
  tls-certificate /etc/pmta/jsmith.pem
  source auth3
</smtp-user>

<source auth3>
  always-allow-relaying yes # allow feeding for auth2
  process-x-virtual-mta yes # allow selection of a VirtualMTA
  max-message-size unlimited
  smtp-service yes          # allow SMTP service
  require-auth true
</source>
```

In the above example, `jsmith` can authenticate with the certificate given in `jsmith.pem` file, and relay through PowerMTA. `<source {auth}>` would apply to any users that don't have `<source>` set.

### 10.4.3 Implementation for Outbound Connections

PowerMTA supports authentication with the CRAM-MD5, LOGIN, or PLAIN mechanisms in outbound connections, with additional mechanisms being considered for a future version.

To have PowerMTA authenticate to a remote mailhost when delivering messages to a domain, define the domain in the configuration file and specify the `auth-username` and `auth-password` directives. Authentication will be attempted only if the remote mailer supports CRAM-MD5, LOGIN, or PLAIN (LOGIN or PLAIN if using the `<domain>` directive `use-unencrypted-plain-auth yes`).

To illustrate how this feature could be used, consider the following scenario: an email service provider using PowerMTA is delivering email to `secure.example.org`, but that domain's corporate mail system will only accept mail from an authenticated source.

To satisfy the corporate mail system's requirement, one could configure PowerMTA to select a VirtualMTA based on pattern matching of the From: address (using the support for pattern matching of any RFC 2822 header), and within that VirtualMTA, configure the required username and password for all mail delivered to `secure.example.org`.

The VirtualMTA definition could look like:

```
<virtual-mta auth-mta>
  smtp-source-host 1.2.3.4 vmta.esp.com
  ...

  <domain secure.example.org>
    auth-username costanza
    auth-password bosco
```

```
</domain>
</virtual-mta>
```

With this, all mail delivered by this VirtualMTA to the domain of secure.example.org would attempt authentication, whereas all other mail delivered to this domain by other VirtualMTAs would not.

Note that authentication is not attempted if either `auth-username` or `auth-password` are not defined for a domain.

## 10.5 Automated Authentication

To help maintain a strong sender reputation, PowerMTA can be configured to ensure SPF compliance by only delivering mail that passes SPF tests. Compliance checking is done based on the sender address (SMTP MAIL FROM and HELO domain), the source IP address to be used in the connection, and the SPF records published in the DNS. If the test results in anything other than a "Pass" or a "TempError", the message bounces.

To enable Automated Authentication for a domain, set the `check-spf-outbound` directive to `yes` in its definition in the configuration file (or in `<domain *>` for all domains).

When enabling this feature, it is advisable to disable it for the domains to which your bounces are delivered, if they are delivered via SMTP. This way, if the check fails, there is no risk of these bounces being lost ("double-bouncing") due to some problem in your bounce domain's SPF records.

For example, if your MAIL FROM domain was yourbouncedomain.com, you could configure:

```
<domain yourbouncedomain.com>
  check-spf-outbound no
</domain>

<domain *>
  check-spf-outbound yes
</domain>
```

Note that since a specific IP address is needed for the Automated Authentication check, you will need to make sure that the IP address(es) used by PowerMTA for outbound connections are defined in the configuration file (via the `smtp-source-host` directive). If no specific source IPs are defined, PowerMTA cannot perform the SPF check. It then generates an error message in the log file and treats this like a transient ("4xx") error, keeping the messages in the queue and retrying until the bounce-after interval passes.

## 10.6 DKIM support

### 10.6.1 Overview

DomainKeys Identified Mail (DKIM) is a sender authentication method based on cryptographically signing messages. Essentially, the steps to signing an email are:

1. A public / private key pair is generated.
2. The public key is published in DNS, while the private key is made available (normally as part of the configuration) to the email server.
3. When a message is submitted for delivery, the email server uses the private key to cryptographically sign the message, adding a DKIM-Signature: header.

More information about DKIM can be found at

<http://www.dkim.org/>

SparkPost highly recommends this site to anyone looking to use this functionality.

### 10.6.2 PowerMTA DKIM Implementation

Domain keys are added to PowerMTA's configuration using the `domain-key` directive, which can be defined globally, making it available for all mail, or within a VirtualMTA definition, making it specific to that VirtualMTA. They can also be defined within a `<domain-key-list>` section which can be selected by name in various ways. The directive takes three comma-separated arguments:

- The key's selector, which identifies that key among others on the same domain.
- The DNS domain name for the key, and
- The name of the file containing the key.

For example:

```
<virtual-mta domainkey>
  host-name port25.com
  ...
  domain-key test, port25.com, c:\pmta\test.port25.pem
</virtual-mta>
```

or

```
domain-key test, port25.com, c:\pmta\test.port25.pem
```

You can specify multiple `domain-key` entries. PowerMTA determines the key to use by going through the keys sequentially and picking the first key whose domain matches the identity on behalf of which the email is being signed. That identity is extracted by default from the email's Sender header address, or, if Sender is not present, the From header, but can also be specified with the `dkim-identity` directive.

A `domain-key` may also specify a wildcard "\*" domain, in which case PowerMTA considers it to match any identity's domain. This is useful when the same public key is entered in the DNS under multiple domain names.

To set up DKIM signing for a domain,

1. Determine the domain in which the key will be registered. That should be either the DKIM identity's domain itself or a parent of that domain. In the examples below, we will assume example.org.
2. Decide what to use for your selector. The selector will uniquely identify the key within the domain determined above. We will use `k1` in the examples below.
3. Generate your private key using either PowerMTA's `pmtakeytool` utility or SparkPost's DomainKey wizard at

<https://port25.com/dkim-wizard/>

(If using the wizard, skip step 8):

pmakeytool newdk k1 example.org 2048

1. This creates a 2048-bit private key in the file k1.example.org.pem and a corresponding public key in BIND format in the file k1.example.org.dns.
2. Move the PEM file to c:\pmta\ on Windows or to /etc/pmta on Unix and ensure its permissions are set properly (ideally only PowerMTA process should have access to it).
3. Add the DNS TXT record written to k1.example.org.dns to your DNS configuration. You may have to adjust the syntax depending on the software used. Also, please refer to the DKIM specification (from <http://www.dkim.org>) for more information on the record syntax:

```
k1._domainkey IN TXT ("k=rsa\; p=<br />
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAXeRmb1kAUC0ZwBL1tS37"
"ubg0k3biFNP8ubH5wc01mkAskRJcSDnXhxr-bNseIBJye2lwVYrSDmkhCo5lreGbt"
"pWdo3kqIEj0k2+ZC2MSNREovoyFXuJHbNDB2VUv1h0ROJMNQx3VuBsYr884K1dsE"
"tzb+9+8JjYwKr6khNu4GeueBT4XdTwcqvtYkqw/Mw4H0RxPJ8/w7pUHy1o08W1Hk"
"N/wAQkhcxygJTaehtkApDFY3N5kVk8XAYdzbJaSe/7Uw0ImqFgxy60Mrw1UaUyW0"
"uTBwIPQScDgX3yI11GHgV25D1EKo8sfTNGFJhi11ju3iZ25KoiPeFyWN+7AohF1Y"
"SQIDAQAB")
```

1. Add the domain-key directive to the configuration file:

```
<virtual-mta example>
...
domain-key k1, example.org, c:\pmta\k1.example.org.pem
</virtual-mta>
```

1. or (at global level in the configuration):

domain-key k1, example.org, c:\pmta\k1.example.org.pem

1. Enable signing for the domain(s) desired, or in <domain \*> to enable signing for all domains:

```
<domain yahoo.com>
...
dkim-sign yes
</domain>
```

1. If the domain-key directive is configured for a VirtualMTA instead of globally, configure PowerMTA so that email originating in the domain(s) to be signed are submitted to that VirtualMTA, for example, by using a pattern list:

```
<pattern-list L>
  mail-from /[@\.]example\.com$/ virtual-mta=example
</pattern-list>

<source 10.1.0.0/16> # feeding sources
  pattern-list L
</source>
```

1. See [VirtualMTA Support](#) for more information on selecting VirtualMTAs. [<pattern-list> Directives](#) describes pattern lists in greater detail.

### 10.6.3 Passing DKIM signing options in a header

PowerMTA supports passing DKIM signing options in a header, overriding the configuration on a per message basis. To do so, enable the <source> directive process-X-DKIM-Options and add a X-DKIM-Options header before submitting the messages. The header uses a syntax similar to the semicolon-separated tags in DKIM-Signature and may be used to pass multiple options:

- s - selector – specifies the value of the s= tag in the signature, and requires that the domain-key have the given selector
- d - domain – specifies the value of the d= tag in the signature, and requires that the domain-key have that exact domain (or the “\*” wildcard)
- i - identity – specifies the value for the i= tag in the signature. Also, if d isn’t specified, requires that the domain-key have the same domain as or a parent domain of i (or the “\*” wildcard)
- q - query method – specifies the value for the q= tag in the signature
- key-list – specifies that the domain key is to be picked from the given [<domain-key-list>](#) section

For example, with the following configuration:

```
domain-key key1, example.com, c:\pmta\key1.example.com.pem
domain-key key2, example.com, c:\pmta\key2.example.com.pem
```

the X-DKIM-Options could be used to select the key to use:

X-DKIM-Options: s=key2

### 10.6.4 Second DKIM signing support

PowerMTA supports adding a second DKIM-Signature header with different options from the first one. All dkim-... directives are supported with the prefix "second-" added:

```
second-dkim-add-body-limit
second-dkim-add-timestamp
second-dkim-algorithm
second-dkim-body-canon
second-dkim-disallow-adding-headers
second-dkim-expire-after
second-dkim-headers
second-dkim-headers-canon
second-dkim-identity
second-dkim-identity-fallback
second-dkim-sign
second-dkim-query-method
```

In the example below, PowerMTA adds both a customer signature and an ESP signature to messages sent through VirtualMTA “12345”. The customer signature is based on the domain of the From header (customer.com) while the ESP signature is based on the second-

dkim-identity option defined (esp.com). PowerMTA would include gmail's required Feedback-ID header in both signatures for messages to gmail only.

```
<virtual-mta 12345>
  domain-key 2017k1, esp.com, c:\pmta\keys\esp.esp.com.pem
  domain-key 2017-01, customer.com, c:\pmta\keys\2017-01.customer.com.pem
  <domain gmail.com>
    dkim-headers Feedback-ID
    second-dkim-headers Feedback-ID
  </domain>
  <domain *>
    dkim-sign yes
    second-dkim-sign yes
    second-dkim-identity @esp.com
  </domain>
</virtual-mta>
```

Note: The X-DKIM-Options header method is not supported for the second signature, but still can be used for the first signature.

PowerMTA will still deliver the message if due to message attributes and configuration settings, there are no signatures to be applied, only one signature to be applied or both signatures are to be applied.

### 10.6.5 Ability to DKIM sign with the private key given in the message

PowerMTA supports the ability to extract a private key from a header in a message and use that private key for DKIM signing the message. This is an alternative to having the keys defined in the configuration file. To enable this feature, use the per-source process-x-dkim-key directive.

If set to true, PMTA will extract the private key specified in the header X-DKIM-Key, and use it for DKIM signing the message. For security reasons, the header will be removed from the message prior to transmission. Note that when using this feature, you have to specify the selector(the 's=' value) for the DKIM signing using the X-DKIM-Options header, otherwise the message cannot be signed. Processing of the X-DKIM-Options header should be enabled using the per-source process-x-dkim-options directive.

Also note that the content of the X-DKIM-Key will span multiple lines, so you need to fold the header in the body of the message as in the example below.

```
Date: Mon, 8 Mar 2017 13:29:22 -0500
From: Test User <test@port25.com>
To: test@port25.com
Subject: help!
X-Dkim-Options: s=ball; d=some-esp.com;
X-Dkim-Key: -----BEGIN PRIVATE KEY-----
MIICdQIBADANBgkqhkiG9w0BAQEFAASCA18wggJbAgEAAoGBAL2/eG1UEGzYGMoN
P8fdRRHF8BZ0/7+aniSIkZq5W84WDjSLtx80ryMYJNtt0tdcIy3vb5YCSqziZncE
a8ite5GiYct2SW+Such/KTWwNuJFEpyz7bjWjP46izuoktZ6JWErnmK+IiPCkc8/
iSVJZGafio5gBwr0mUi2YDGUX/RAgMBAAECgYBGYCSV1EL/S6ruDbnxNAUxEVv5
yE0u0HcIrYurekItjoqYPJWfpD+a4ZKR+9JOG/NET0Dwv3Fro9DjJbWgSY/BkSKs
ckypdEh2DEctAaqpZ18Bs7nmaHnsyhaFHaTvT/VKM/WVvBXfh+NlaHoJK05um9MS
jVwiDkZHVeKu6g+y0QJBA0392X05FziwQYF55Y6ldfyJfWQzRmGtsz8B1HGeS84N
wuMtP4zjL/c5nwOkN5S08inE7xZLy958WQQtZW9BW00CQQDMGxbwuK4HmBLEjLZw
R7g9SH8b/54bWwt+4BmiWjF1Dnbeq/5Co2ZHcXQ050fopk/OsAsN/F0VZb55781z
nsyVakAyhSieBHiBeCjvowFH/qfFLAqKMrhGDmyGe4T1eXzuCM30G9TZu4B7Iag2
xa1mw9F0PHHkccKdQBhFYr2dmGohAKAJi6bKAUKaT9FqGRibLa9/JjnTnGPPkaIG
YAKyShb0C4YhqdeNVEj/RF4MQ9TIG6cV2oPZ1pif3I9I8p7wSLf1AkBznyYhKnML
OdsR8QUfwVm8o5TtFz+kisDJ7fQc1kCruB5IyLXPnKxugUcetHBVnC4Y70996N4R
mkoEKsyfiQhk
-----END PRIVATE KEY-----

HELP! How do I port25?
Thanks abunch,
.. Test
```

With the email above, you have to add the following directives to PMTA config to DKIM sign the message.

```
<source 0/0>
  process-x-dkim-key yes
  process-x-dkim-options yes
</source>

<source {pickup}>
  process-x-dkim-key yes
  process-x-dkim-options yes
</source>
```

## 10.7 Warming of IPs with Cold VirtualMTA

### 10.7.1 Overview

**Note: This functionality is useful when you have at least one warm IP address. If you only have cold IP addresses then use rate limiting features of the VirtualMTAs to build up reputation for the cold IPs.**

When starting to send email from a new IP address, it is sometimes important to control the amount of traffic sent from that IP, so that reputation can be built properly. To facilitate this task, PowerMTA adds support for configuring separate queues and diverting part of the traffic to those queues.

The cold VirtualMTA functionality, at its core, is a rerouting feature. **The first x messages for each domain submitted to the warm VirtualMTA each day are auto-rerouted to the cold VirtualMTA.** The number that is rerouted for each domain per day is determined by the max-cold-virtual-mta-msg directive defined in the warm VirtualMTA for the domains. The rest of the messages submitted that day to the warm VirtualMTA stay in the warm VirtualMTA. The cold VirtualMTA rerouting counters are reset at midnight.

The messages routed to the cold VirtualMTA are handled/delivered based on the directives defined in the cold VirtualMTA. To this end it is important to configure the cold VirtualMTA as desired.

**Note: When MX rollout rules are also active in the configuration, PowerMTA will use the max-cold-virtual-mta-msg setting from the rollout queue instead of the individual queues that got rolled up.**

Note: PowerMTA may route less traffic through cold VirtualMTAs when an <mx-rollup-list> tag exists in the configuration, even for domains that aren't rolled up.

For example:

```

<virtual-mta vmta1>
  smtp-source-host 1.2.3.4 vmta1.example.com
  cold-virtual-mta vmta2
  <domain *>
    max-cold-virtual-mta-msg 1000/day
  </domain>
</virtual-mta>

<virtual-mta vmta2>
  smtp-source-host 5.6.7.8 vmta2.example.com
  <domain *>
    max-msg-rate 100/h
  </domain>
</virtual-mta>

```

With these settings, the first 1000 messages submitted into the warm VirtualMTA vmta1 per day, for each domain, would be auto-rerouted to the cold VirtualMTA vmta2. Every message submitted after the first 1000 per day per domain to the warm VirtualMTA would stay in the warm VirtualMTA, with the daily rerouting counters resetting at midnight. Because **max-cold-virtual-mta-msg setting is per domain**, the first 1000 submitted for yahoo.com to vmta1 per day would be rerouted to vmta2, the first 1000 submitted for hotmail.com to vmta1 per day would be rerouted to vmta2, etc.

This is the count for each domain separately given that max-cold-virtual-mta-msg 1000/day is defined in <domain \*> in the VirtualMTA in this example. If different counts per day for different domains are needed, define the different domain definitions in the VirtualMTA with the specific counts desired for each, along with a count for all other domains in <domain \*>. Messages that are routed to the cold VirtualMTA are then handled based on the settings defined in the cold VirtualMTA.

The default setting for the "max-cold-virtual-mta-msg" directive is 0, which means "don't re-select a cold VirtualMTA". If "max-cold-virtual-mta-msg" is set globally but the VirtualMTA selected doesn't have a "cold-virtual-mta" directive configured, no re-selection takes place.

Since the cold VirtualMTA is just another VirtualMTA, this feature can also be used to route traffic from an old server to a new server. This is useful when moving server locations and the new server is getting new IPs that are not yet established with a good reputation. For example:

```

<virtual-mta vmta1>
  smtp-source-host 1.2.3.4 vmta1.example.com
  cold-virtual-mta vmta2
  <domain *>
    max-cold-virtual-mta-msg 1000/day
  </domain>
</virtual-mta>

<virtual-mta vmta2>
  smtp-source-host 5.6.7.8 vmta2.example.com
  <domain *>
    queue-to route.queue
  </domain>
  <domain route.queue>
    smtp-hosts [9.8.7.6]:25
  </domain>
</virtual-mta>

```

In the above example, the first 1000 messages per domain per day injected into vmta1 are routed to vmta2. At this point they are grouped into one large queue with the queue-to directive to help better manage the amount of outbound connections the messages will make.

Both the limits and the selection are persisted on disk, so that you can safely restart PowerMTA without resetting the current rates.

## 10.8 Use of Bounce Category Patterns

### 10.8.1 Overview

This feature can be used to make the bounce categories used in CSV accounting files field customizable. A customer can specify the tag in the configuration file, in which case the patterns specified there take precedence over those defined in config-defaults file. The data in the <bounce-category-patterns> tag consist of two fields. The first field is the SMTP response on which to match, and the second field is the name of the category to be used. Custom category names are allowed. It should be entered in a <bounce-category-patterns> entry, like in:

```

<bounce-category-patterns>
  /spam/ spam-related
  /no longer (valid|available)/ bad-mailbox
  /mailbox +(is +)?full/ quota-issues
  /banned/ policy-related
</bounce-category-patterns>

```

<bounce-category-patterns> uses pattern matching in the same method as does <pattern-list>. [<smtp-pattern-list> Directives](#) describes SMTP pattern lists and pattern matching in greater detail.

## 10.9 Combining several domains into one queue

### 10.9.1 Overview

PowerMTA has the ability to group together different email domains into a single large queue. This can be useful for controlling the message stream across multiple domains. This includes any <domain> based controls such as max-msg-rate, max-smtp-out, and max-connect-rate.

**Note that you will need to manually define the smtp-hosts directive for the new queue if it is not resolvable through DNS.**

The directive to use is "queue-to", which is defined in the <domain> definitions that you want to group together. The parameter defined within the queue-to directive is the name of the "domain" that the other domains will be rolled up into. Use of the smtp-hosts directive is required.

For example:

```

<virtual-mta hotmail-queue>
  <domain hotmail.queue>
    max-smtp-out 10
    smtp-hosts hotmail.com # you must define this for the consolidated queue.
  </domain>
  <domain hotmail.com>
    queue-to "hotmail.queue"

```

```

</domain>
<domain hotmail.co.uk>
    queue-to "hotmail.queue"
</domain>
<domain live.com>
    queue-to "hotmail.queue"
</domain>
</virtual-mta>

```

With this, all messages queued for this VirtualMTA for any hotmail.com, hotmail.co.uk, or live.com domain will be put in one queue named "hotmail.queue", and treated as such.

Or:

```

<virtual-mta client-mta>
    smtp-source-host 1.2.3.4 mail1.yourdomain.com
    <domain *.psmtp.com>
        queue-to "psmtp.com"
        max-smtp-out 20
        bounce-after 2d
        max-msg-per-connection 10
    </domain>
</virtual-mta>

```

With this, all messages queued for this VirtualMTA for any \*.psmtp.com domain (e.g. mail9.psmtp.com, someone.psmtp.com, etc.) will be put in the one queue for psmtp.com, and treated as such. As a note, if you are using a cold VirtualMTA, the directive max-cold-virtual-mta is applied AFTER queue-to is applied.

## 10.10 Setting recipient priority

### 10.10.1 Overview

It may be necessary to set a given recipient of set of recipients with a higher or lower priority (e.g. password reset vs. weekly newsletter). This can be done in PowerMTA in different ways. The first is via the <domain> directive "queue-priority". The directive works in the following manner:

```

<domain someplace.com>
    queue-priority 75
</domain>

<virtual-mta newsletter>
    <domain someplaceelse.com>
        queue-priority 30
    </domain>
</virtual-mta>

```

With the above set, all messages going to someplace.com will get a queue-priority of 75 (the default is 50 with a range of 1-100, higher amounts being given higher priority). This will give this set of messages first priority in PowerMTA for connections and delivery. The messages for someplaceelse.com/newsletter will get a lower priority than the average mail, and will go out last.

Another method of setting the message priority is by using a <pattern-list>. The directive works in the following manner:

```

<pattern-list myList>
    rcpt-to /password-reset@customer.com/ recipient-priority=100
    rcpt-to /special@customer.com/ virtual-mta=high, recipient-priority=60
    mail-from /newsletter/ recipient-priority=25
</pattern-list>

```

In the above example, individual emails to [password-reset@customer.com](mailto:password-reset@customer.com) will get the highest priority, generic emails to [special@customer.com](mailto:special@customer.com) will get a slightly higher than normal priority, and any emails from anything containing newsletter will get a low priority.

If both queue-priority and recipient-priority are used, queue-priority takes precedence. recipient-priority is only used for determining which recipients in a queue are attempted first for delivery.

## 10.11 SNMP Support

### 10.11.1 Overview

PowerMTA supports SNMP for monitoring basic status information.

Currently all the information from the pmta show status command is what is available for monitoring via SNMP.

PowerMTA SNMP support is included with the Windows installer. There is a separate package available for Linux from the PowerMTA download page. PowerMTA's SNMP daemon can be used in master mode or as an AgentX subagent (not available on Windows). It automatically detects whether there is a AgentX master running (e.g. Net-SNMP's snmpd). If so it registers itself as a subagent, otherwise it runs in master mode.

As a note, to allow Net-SNMP's snmpd to act as a AgentX master it is required to activate the AgentX support in snmpd (by having a line "master agentx" in snmpd.conf).

The default configuration for PowerMTA's SNMP daemon is located in:

Linux:

```
/etc/pmta/pmtasnmplib.conf
```

Windows:

```
<PowerMTA install directory>\pmtasnmplib.conf
```

This will allow anybody to read data from it with the "public" community string, which may or may not be appropriate. Since the SNMP daemon is based on Net-SNMP the configuration file is a standard Net-SNMP configuration file and documentation is available from:

<http://www.net-snmp.org/docs>

The data available through SNMP is described by the MIB which can be found at the following location:

Linux:

```
/usr/share/snmp/mibs/PowerMTA-MIB.txt
```

Windows:

```
<PowerMTA install directory>\mibs\PowerMTA-MIB.txt
```

To access the data via SNMP a third party solution is required.

On Linux you have the Net-SNMP utils installed you can test that it works (and see the data available) with the following command:

```
snmpwalk -v2c -c public localhost PowerMTA-MIB::mta
```

## 10.12 High Availability Configuration

In "High Availability" deployment, two servers run PowerMTA service, and if one fails, new mail gets sent to the backup server by the feeding application. If the spool directory is recoverable, it is added as a second spool to the backup, and the service is restarted to pickup this spool, sending all new and old mail.

When the production server is backup and online, new mail is redirected back to the production server, and the hot backup is left running until all mail has been delivered or bounced. After the mail is delivered from the hot backup, it is put back into its original single spool state, and is kept idle until the next time it is needed.

## 10.13 Precached domains support

For senders that have DNS server issues when loading large amounts of domains into PowerMTA in a short period of time, or when maintaining large amounts of domains with short a short DNS TTL, PowerMTA offers the ability to precache a given subset of the DNS names to ensure the DNS name is always available. This feature is enabled by use of the new, global `<dns>` tag.

```
<dns>
  precached-domains-file C:\pmta\precached-domains.txt
  precached-max-domains 500
  precached-refresh-interval 5m
</dns>
```

The file `C:\pmta\precached-domains.txt` would contain a list of domains, each on a line by itself. For example,

```
hotmail.com
yahoo.com
aol.com
gmail.com
```

PowerMTA would only precache the first 500 domains in the file, refreshing the DNS for these domains every 5 minutes.

## 10.14 Scheduled Delivery Control

PowerMTA supports the ability to schedule deliveries via a header.

The format of the header is:

```
x-schedule: <start time>/<end time>, <start time>/<end time>
```

Here is an example:

```
x-schedule: 2015-05-29 17:01:11 / 2015-05-29 17:30:11, 2015-05-30 17:01:11 / 2015-05-30 17:30:11
```

You can add as many time intervals as needed.

The bounce-after does not apply to such recipients. The schedule overrides the bounce-after and the message is bounced when there are no more schedules to try. In addition, if the queue is in retry mode when the messages are injected, the message delivery will not start at the window start time, but rather message delivery attempts will start when the queue come out of retry mode.

## 10.15 Custom retry intervals

retry-after and backoff-retry-after can also take a time interval in addition to the static, defined time. This allows for adding up to 30 different retry periods. This may be useful for domains in which the likelihood of messages being delivered decreases over time. The last defined interval is used until the message bounces. The interval resets on a restart of the PowerMTA server. An example usage would be:

```
<domain example.com>
  retry-after 10m,10m,10m,10m,10m,30m,30m,30m,30m,1h,1h,2h,2h,4h,4h,10h
</domain>
```

In the above example the domain would be tried every 10 minutes for the 6 tries, every 30m for the next 4 tries, every hour for the 2 tries, every 4 hours for the next two tries, and finally, every 10 hours until the message is bounced.

For configuring retry interval specific to DNS resolution errors, see [retry-after-dns-error](#)

## 10.16 Address Suppression Lists

Lists can be defined with the new global `<address-list listName>` tag and one or more can be referenced within a `<source>` with the new "suppression-lists" directive. Addresses in the suppression list are rejected (or turned into bounces, depending on options) during submission. Multiple suppression lists may be used:

```
<address-list suppression1>
  address foo # local part
  address foo@bar.com # email address
  address /foo/ # pattern
</address-list>

<address-list suppression2>
  domain foo.com # domain
  address-file /etc/pmta/addresses # email addresses or local parts, 1/line
</address-list>

<source 0/0>
  suppression-lists suppression1,suppression2
</source>
```

NOTE: Defining more than 500 patterns within a single suppression list is not recommended.

If the `<source>` directive `accept-invalid-recipients` is set to "no", then the message is rejected at the time of submission (i.e., while processing the RCPT command).

If the <source> directive accept-invalid-recipients is set to "yes", then the message is queued and later bounced with generation of a bounce record with details such as:

```
dsnStatus: 5.7.1 (delivery not authorized)
dsnDiag: smtp;550 5.7.1 recipient is in suppression list "a1": <foo@bar.com>
```

For BSMTP pickup files, If XACK is set to "on", then the message is rejected while the file is picked up (i.e., while processing the RCPT command). This is logged in the PowerMTA log file just like any other pickup submission error. If XACK is set to "off", then the file is picked up and the message is later bounced with a bounce record that looks similar to SMTP submission case above.

For non-BSMTP pickup files, the message is rejected while the file is picked up (i.e., when processing the x-receiver header). This is logged in the PowerMTA log file just like any other pickup submission error.

## 10.17 MX Rollup List

This feature allows one to define "rollup" queues based on the MX records of the recipient domain, in order to consolidate separate but related recipient domains into one queue. Messages moved into the roll up queue are handled as if they were one recipient domain with regards to rate limiting, connection caps, etc.... For example, since msn.com, live.com and hotmail.com are all handled by the same exact MXs/gateway, it makes sense to roll these up into one queue vs. being handled separately by PowerMTA. This feature works very well for the large mailbox providers that provide corporate hosting, for filtering/antispam cloud providers, or for large hosting providers in general that manage corporate email for tens or hundreds of thousands of various domains.

**Note: When MX rollup rules are also active in the configuration, PowerMTA will use the max-cold-virtual-mta-msg setting from the rollup queue instead of the individual queues that got rolled up.**

To configure this feature, you need to define the MX record that maps to each rollup queue name along with the name of the rollup queue, which **must** end in ".rollup". More than one MX can be mapped to the same rollup queue, for example:

```
<mx-rollup-list>
  mx mx1.hotmail.com hotmail.rollup
  mx mx2.hotmail.com hotmail.rollup
  mx mx3.hotmail.com hotmail.rollup
  mx mx4.hotmail.com hotmail.rollup
  mx smtp.secureserver.net secureserver.net.rollup
</mx-rollup-list>
```

**NOTE** that only the lowest preference (highest priority) MX records should be defined with this feature.

Wildcards are also supported for the MX records, for example:

```
<mx-rollup-list>
  mx *.hotmail.com hotmail.rollup
  mx *.mail.protection.outlook.com outlook.rollup rollup-by-ip
</mx-rollup-list>
```

which would then automatically include all matching lowest preference MX records. Any recipient domain whose lowest preference MXs match the wildcard would be automatically placed in the hotmail.rollup queue, and which all messages would be treated as if to a single recipient domain.

**No match occurs when a rollup queue defined in the <mx-rollup-list> has more lowest preference MXs configured to it than what is returned by DNS.**

For rollup queues in which the MX records are explicitly defined (no wildcards) in the rollup list, PowerMTA will automatically use/look up the specific defined MXs in DNS for routing information for the domains in the rollup queue.

For rollup queues in which a wildcard is used, one **must** explicitly define the smtp-hosts directive within the domain definition for the rollup queue along with the appropriate MX or A records, since it is impossible for PowerMTA to infer the proper MXs to use for all domains in the rollup queue (in contrast to when they are explicitly defined).

For rollup queues in which a wildcard is used with the "rollup-by-ip" option, there is no need to configure "smtp-hosts" directive. PMTA will automatically set the "smtp-hosts" for such rollup queues to the hosts that the original domain resolved to. In order to perform the rollup action, PMTA will use the IP addresses that the MXes resolve to, not the MX name.

SparkPost recommends always explicitly defining the rollup queue within its own domain definition in order to explicitly set the delivery controls you want to achieve for the rollup queue.

For clarification, the table below provides various scenarios and end results based on the following MX rollup list definitions and defined DNS results.

**Configuration settings for MX Rollup list to be used for examples below:**

```
<mx-rollup-list>
  mx mx1.domain.com domain.rollup
  mx mx2.domain.com domain.rollup
  mx mx1.domain2.com domain2.rollup
</mx-rollup-list>
```

The below table lists various examples of how this would work. For all examples the recipient domain would be domain.com.

### Example 1

**DNS results for domain's MX lookup (MX name + preference)**

```
mx1.domain.com 10
mx2.domain.com 10
```

**Rollup Queue Used** domain.rollup

**Destination MXes for the Rollup Queue**

```
mx1.domain.com
mx2.domain.com
```

**Explanation** MXs defined for rollup queue are present in domain's DNS results.

### Example 2

**DNS results for domain's MX lookup (MX name + preference)**

```
mx1.domain.com 10
mx2.domain.com 20
```

**Rollup Queue Used** N/A – rollup would not take place

**Explanation** Rollup queue has more lowest-preference MXs defined than what DNS returns.

#### Example 3

**DNS results for domain's MX lookup (MX name + preference)**

```
mx1.domain.com 10
mx2.domain.com 10
mx3.domain.com 10
```

**Rollup Queue Used** domain.rollup

**Destination MXes for the Rollup Queue**

```
mx1.domain.com
mx2.domain.com
```

**Explanation** MXs defined for rollup queue are present in domain's DNS results, mx3.domain.com is not used by PowerMTA for delivery.

#### Example 4

**DNS results for domain's MX lookup (MX name + preference)**

```
mx1.domain.com 10
```

**Rollup Queue Used** N/A – rollup would not take place

**Explanation** Rollup queue has more lowest preference MXs

#### Example 5

**DNS results for domain's MX lookup (MX name + preference)**

```
mx1.domain2.com 10
mx2.domain2.com 10
```

**Rollup Queue Used** domain2.rollup

**Destination MXes for the Rollup Queue**

```
mx1.domain2.com
```

**Explanation** MXs defined for rollup queue are present in domain's DNS results, mx2.domain2.com is not used by PowerMTA for delivery.

#### Example 6

**DNS results for domain's MX lookup (MX name + preference)**

```
mx1.domain2.com 10
mx1.domain2.com 20
```

**Rollup Queue Used** domain2.rollup

**Destination MXes for the Rollup Queue**

```
mx1.domain2.com
```

**Explanation** MXs defined for rollup queue are present in domain's DNS results.

#### Example 7

**DNS results for domain's MX lookup (MX name + preference)**

```
mx1.mail.domain.com 10
```

**Rollup Queue Used** N/A – rollup would not take place

**Explanation** DNS results do not match any defined rollup queue.

## 10.18 Recipient Events

To aid in real-time queue diagnostics, PowerMTA shows the last 50 recipient delivery, bounce, and deferred events. This data is disabled by default, and can be enabled by setting:

```
<domain *>
  track-recipient-events true
</domain>
```

You can also enable this feature for only some of the domains instead of all.

The data is available in the output of the following commands in addition to being displayed on the individual queue detail page in the web monitor:

```
pmta show queues
pmta show topqueues
pmta show queue domain[/vmta]
```

The information is held in memory as long as the queue has any recipients in it.

## 10.19 HTTP Transmissions API

PowerMTA's HTTP Transmissions API allows you to submit email into PowerMTA through HTTP. Each API request can include single or several recipients. At present, there is no support for customizing the body of the message using mailmerge-like substitution syntax.

### 10.19.1 Setup

To use the HTTP Transmissions API, you'll need to configure in `<transmissions-api>` and one or more `<transmission-api-listener>`s. Since the communication from clients actually uses HTTPS, you will need to specify a certificate to each listener.

Within `<transmissions-api>` you will also need to configure one or more `api-keys`, alphanumeric strings that connecting clients will use to authenticate themselves before being allowed to submit email. Clients pass such an API key using the Authorization HTTP header.

You can also set `<source>`-level properties for emails submitted via the Transmissions API using the named source `{transmissions-api}`. For example, you could set `process-x-virtual-mta yes` within that `<source>` to enable PowerMTA to process the X-Virtual-Mta header for emails submitted over HTTP Transmissions API.

Here is a sample configuration:

```
# Listener specific configuration here (TLS certificate, CA certificates/CRL, trusted certificates etc.)
<transmissions-api-listener 1.2.3.4:8585>
  # tls-certificate is required, other directives in this scope are optional
  tls-certificate /etc/pmta/transmissions-api-certificate.pem

  # other directives with "tls-" prefix are also valid here, such as
  # "tls-trusted-certificate" etc.
</transmissions-api-listener>

# Transmissions API specific configuration
<transmissions-api>
  log-requests yes
  api-key 14ac5499cfdd2bb2859e4476d2e5b1d2bad079bf
  log-data no
  log-tls no
</transmissions-api>

# add any source settings that affect messages received via HTTP API
<source {transmissions-api}>
  process-x-dkim-options yes
  process-x-virtual-mta yes
  process-x-job yes
</source>
```

### 10.19.2 Creating a certificate

You can use a self-signed certificate or a certificate signed by a certification authority (CA). A self-signed certificate is easy to create, while a CA signed certificate is more secure, but has the overhead of having the certificate signed.

You can create a self signed certificate using the `pmtakeytool` utility that comes with PowerMTA.

```
pmtakeytool newcert > /etc/pmta/transmissions-api-certificate.pem
```

You can also provide your own certificate obtained from a CA to the `tls-certificate` directive. If you were provided with a list of CA trust anchors, and/or CRLs, configure them in a separate file using the `tls-ca-file` directive within `<transmissions-api>` tag.

### 10.19.3 API Keys

The syntax of the `api-key` directive is similar to the syntax of the `http-api-key` directive. The value for this directive must be a string of 32 to 128 characters, each character being alphanumeric ([a-zA-Z0-9]) or underscore ('\_'). The key is case-sensitive.

You can use the `api-key` directive multiple times, specifying multiple keys, e.g., for different users. Transmissions API request can use any of the `api-key` values configured in `<transmissions-api>`.

### 10.19.4 Submitting messages to PMTA using the Transmissions API

To submit messages, make an HTTP POST request with its HTTP Content-Type header set to `application/json`. Here's an example request sent using the Unix command line tool `curl`.

It assumes the JSON encoded Transmissions API request is in the file named `msg.json` in the current directory, that `0f5019df4641c37a65212564980f5c320d9a5d84` is the value of an `api-key` configured in `<transmissions-api>`, and `<transmissions-api-listener 127.0.0.1:8585>` is defined in the configuration file. The `--insecure` option is only needed for self-signed certificates.

```
curl -X POST --insecure \
  -H "Content-Type: application/json" \
  -H "Authorization: 0f5019df4641c37a65212564980f5c320d9a5d84" \
  --data @msg.json \
  https://127.0.0.1:8585/api/v1/transmissions
```

The `msg.json` file could look like this:

```
{
  "return_path": "xyz@bounce.port25.com",
  "campaign_id": "xmas2018",
  "options": {
    "smtpUTF8": true
  },
  "recipients": [
    {
      "address": {
        "email": "fred@flintstone.com"
      }
    },
    {
      "address": "wilma@flintstone.com",
      "return_path": "wilma@bounce.port25.com"
      "substitution_data": {
        "*custommessageid": "01234.abc@mysite"
      }
    }
  ],
  "content": {
```

```

"emailRFC822": "From: \"Xmas 2018\" <xmas2018@transmissions.port25.com>\n Content-Type:
text/plain\nTo: \"Fred Flintstone\" <fred@flintstone.com>\n \nHi Fred Flintstone,\nSave big this Christmas
in your Bedrock area!\n Click http://www.mysite.com and get huge discount\nHurry, this offer is only to
Platinum customers.\nBig Store Team\n"
}
}

```

The request body consists of exactly one JSON object (i.e., one email message). It can have multiple recipients and message metadata like the campaign ID (job ID).

Here are all the fields allowed within the JSON request object along with an explanation of the possible JSON fields. All field names are case-sensitive.

`return_path`

**Type:** string

**Description:** Email address to use for envelope FROM. This field can be used at message level to set the originator address for all recipients in the email. It could also be specified at recipient level in the `recipient` object, which would override the per-message `return_path`. A `return_path` field must be specified at either message-level, or at recipient-level, else the API rejects the request as **400 Bad Request**.

`campaign_id`

**Type:** string

**Description:** The campaign ID for this request. This field can be used at message level to set the campaign ID for all recipients in the email. In PowerMTA, the equivalent of the term `campaign` is `job`. The content of this field is treated as the job ID of the message within PowerMTA. If present in the request, this field overrides the `x-job` header contained within the body of the message in case the `{transmissions-api}` source also contains `process-x-job yes` directive. The syntax rules for Campaign ID field are the same as the rules for Job in PowerMTA.

`options.smtpUTF8`

**Type:** boolean

**Description:** If the email is a SMTPUTF8 message (any or both of recipient and originator addresses have non-ASCII local parts), set the "smtpUTF8" option to true. If missing or set to false and any or both of recipient and originator addresses have non-ASCII local parts, the message is rejected as a bad request.

**Note:** If any other values in `options` as described in Sparkpost API documentation (`start_time` etc.) are present, the request is rejected since those are unsupported in PowerMTA at present.

`recipients`

**Type:** array of JSON objects, each object representing a recipient object. See below for more description about recipient object.

`recipient`

**Type:** Object

**Description:** Represents a recipient object.

`recipient.address`

**Type:** Object or string

**Description:** If a string, it is interpreted as the email address. Otherwise, it must be an object with the following attributes.

`recipient.address.email`

**Type:** string, required

**Description:** The email address of the recipient.

`recipient.return_path`

**Type:** string, optional

**Description:** The optional, recipient-level FROM address for this recipient that overrides an per-message `return_path` in the request.

`recipient.substitution_data`

**Type:** Object, optional

**Description:** The optional, recipient-level substitution data for this recipient. The only key supported within this object is `*custommessageid`.

`recipient.substitution_data.*custommessageid`

**Type:** string, optional

**Description:** The optional, recipient-level custom message ID for SparkPost Signals events. If specified, this takes precedence over any value that is set with the [custom-message-id-header](#). If this is omitted and the [custom-message-id-header](#) is set, then the value from the header is sent to SparkPost. If this omitted and [custom-message-id-header](#) is not set, then no custom message ID is sent to SparkPost.

`content.emailRFC822`

**Type:** string

**Description:** Pre-built message with the format as described by the [message/rfc822](#) Content-Type.

## 10.19.5 Responses

If the HTTP request was accepted by PowerMTA, a **200 Ok** response is returned, along with the number of recipients in the message that were accepted.

```

HTTP/1.1 200 Ok
Content-Type: application/json
Content-Length: 73

{

```

```

    "results":
    {
      "total_accepted_recipients":1,
      "total_rejected_recipients":0
    }
  }

```

### 10.19.6 Errors

In case of errors seen during the transmission, PowerMTA will return an appropriate HTTP error, along with data like the following in the body of the response.

```

HTTP/1.1 400 bad request
Content-Type: application/json
Content-Length: 118

{
  "errors": [
    {
      "message": "some error",
      "description": "more detail about the error"
    }
  ]
}

```

### 10.19.7 List of HTTP errors returned by the API

The following HTTP errors could be returned by the API

#### 400 Bad Request

**Description:** In case of errors with the format of the request (if a required field was missing, a JSON field had invalid data type, if the request contained a Sparkpost Transmissions API field that is unsupported in PowerMTA Transmissions API etc.), PowerMTA will return 400 Bad Request with the following data:

```

HTTP/1.1 400 bad request
Content-Type: application/json
Content-Length: [0-9]+

{
  "errors": [
    {
      "message": "bad request",
      "description": "more detail about the request"
    }
  ]
}

```

#### 403 Forbidden

**Description:** In case of invalid or missing Authorization header in the request.

#### 404 Not Found

**Description:** In case of the URI in the request was invalid.

#### 405 Method Not Allowed

**Description:** In case of the request used a HTTP method other than POST.

#### 500 Internal Server Error

**Description:** In case of the internal errors when attempting to process a valid HTTP API request.

## 10.20 HTTP Delivery Webhooks.

PowerMTA can be configured to deliver email from a queue to an HTTP endpoint. This can be useful, for example, to relay messages (such as feedback, or remote bounce reports) from a queue within PowerMTA to an external/internal service endpoint over HTTP(S) for additional processing.

To use HTTP delivery, you will need to configure the queue's type to be `http-delivery`, and configure the address of the HTTP endpoint that PowerMTA will deliver to using the [http-delivery-url](#) directive. You can also optionally configure PowerMTA to additionally include extra HTTP headers (such as Authorization) as part of the HTTP request when delivering email to the configured endpoint.

PowerMTA will assume that the message was delivered successfully only if the destination returns HTTP response code 200. All other HTTP responses from the destination codes are treated as temporary failures to deliver and the message is held back in the queue. The queue's bounce-after setting will determine when a message that couldn't be delivered from this queue will get bounced as expired.

Only one message is included within the body of an HTTP request from PowerMTA.

If the content of the message contains any non-7bit ASCII characters, PowerMTA will Base64 encode the entire content of the message before including it in the body of the HTTP request. A JSON field `email_rfc822_is_base64` will be set to `true` to indicate that the content is in Base64.

### 10.20.1 Example Configuration

```

<domain foo.com>
  type http-delivery

  # Delivery URL. Must be absolute, have "http" or "https" scheme and
  # not include "userinfo", "query-string" or "fragment" components.
  http-delivery-url https://webhooks.endpoint.com:8181/api/v1/uri

  # (optional) HTTP headers to be included in the request during HTTP delivery,
  # for example, to add authentication headers.
  http-delivery-extra-headers /etc/pmta/http-delivery-headers.txt
</domain>

```

Using the example configuration above, PowerMTA will start delivering messages from the queue `foo.com` to the destination host `webhooks.endpoint.com`, port 8181 using HTTPS.

### 10.20.2 Delivery request format

PowerMTA will deliver the messages from an http-delivery queue as a HTTP POST request with Content-Type set to application/json. The body of the request will be a JSON object with the following fields:

email\_rfc822

**Type:** string

Raw MIME content for an email. If the Raw MIME content contains at least one non UTF-8 encoded character, the entire email\_rfc822 value will be base64 encoded and email\_rfc822\_is\_base64 will be set to true.

email\_rfc822\_is\_base64

**Type:** boolean

Whether the email\_rfc822 value is base64 encoded. Only included if it is true. If the field is missing, the email\_rfc822 can be assumed to be unencoded.

msg\_from

**Type:** string

SMTP envelope "MAIL FROM"

rcpt\_to

**Type:** string

SMTP envelope "RCPT TO"

headers

**Type:** array of objects, each object representing a header extracted from the SMTP message being delivered, consisting of a single key/value pair of header name and its value.

### 10.20.3 Sample HTTP Delivery Webhook

```
POST /api/v1/webhooks HTTP/1.0
Host: webhooks.endpoint.com
Content-Type: application/json
X-Header-One: Value-One
X-Header-Two: Value-Two
X-Header-Three: Value-Three
Content-Length: 523

{
  "msg_from": "me@bounce.port25.com",
  "rcpt_to": "a@port25.com",
  "content": {
    "email_rfc822": "Return-Path: <me@here.com>\r\nMIME-Version: 1.0\r\nFrom: me@here.com\r\nReceived: by 10.114.82.10 with HTTP; \r\n    Mon, 4 Jul 2016 07:53:14 -0700 \\\(PDT\\)\r\nDate: Mon, 4 Jul 2016 15:53:14 \\\(+0100\r\nMessage-ID: <484810298443-112311-xqxbby@mail.there.com>\r\nSubject: Relay webhooks rawk!\r\nTo: you@there.com\r\nContent-Type: multipart/alternative;\r\nboundary=deaddeaf00fall145dbhai1980dhypnot0ad\r\n\r\n--\r\ndeaddeaf00fall145dbhai1980dhypnot0ad\r\nContent-Type: text/plain; charset=UTF-8\r\nHi there\r\nPort25!\r\n\r\n--deaddeaf00fall145dbhai1980dhypnot0ad\r\nContent-Type: text/html; charset=UTF-8\r\n\r\n<p>Hi there <strong>Port25!</strong></p>\r\n\r\n--deaddeaf00fall145dbhai1980dhypnot0ad--\r\n",
    "headers": [
      {
        "Return-Path": "<me@here.com>"
      },
      {
        "MIME-Version": "1.0"
      },
      {
        "From": "me@here.com"
      },
      {
        "Received": "by 10.114.82.10 with HTTP; Mon, 4 Jul 2016 07:53:14 -0700 \\\(PDT\\)"
      },
      {
        "Date": "Mon, 4 Jul 2016 15:53:14 \\\(+0100"
      },
      {
        "Message-ID": "<484810298443-112311-xqxbby@mail.there.com>"
      },
      {
        "Subject": "Relay webhooks rawk!"
      },
      {
        "To": "you@there.com"
      },
      {
        "Content-Type": "multipart/alternative; boundary=deaddeaf00fall145dbhai1980dhypnot0ad"
      }
    ]
  }
}
```

## 10.21 DANE

DANE (DNS-Based Authentication of Named Entities) is a new standard for improving transport security. RFC 7672 describes the use of DANE for improving SMTP security between MTAs.

PowerMTA has supported STARTTLS for a long time. The use of STARTTLS for establishing channel security (encrypting the channel) can be achieved via the use of [use-starttls](#) directive. use-starttls works in an opportunistic way - if the remote side supports STARTTLS, PowerMTA will attempt to setup encryption using STARTTLS. If the remote side does not support STARTTLS, or if the attempt to setup STARTTLS fails for any reason, the SMTP STARTTLS protocol allows PowerMTA to downgrade to plain-text for the purpose of sending email. The downgrade happens silently without any notification to the user. PowerMTA also supports [require-starttls](#), which disallows downgrading to plain text, but it's up to customers to configure the use of this directive (it is set to false by default, meaning that PowerMTA will automatically downgrade to plain-text if STARTTLS fails or is unsupported).

Even if STARTTLS were to be issued, the current STARTTLS standard does not enforce the use of Public Key Infrastructure(PKI) to verify the authenticity of the certificates. The remote site is allowed to present any certificate (whether self-signed, or issued by a certificate authority (CA) that is unknown to PowerMTA). It is up to PowerMTA customers to configure the CA certificates in PowerMTA that they trust and configure PowerMTA to verify the certificate chain presented by the remote site (via the use of [smtp-tls-ca-file](#) directive).

DANE for SMTP security allows the remote side to enforce the use of STARTTLS whenever clients like PowerMTA connect to the site to transfer email. RFC 7672 describes various methods through which remote sites could advertise the availability of STARTTLS with them, and whether they require that connecting clients must always use STARTTLS when communicating with them. DANE requires that the DNS resolver support DNSSEC, a secure way of looking up DNS records. Google DNS (8.8.8.8) is an example of a DNS lookup service that supports DNSSEC.

PowerMTA 5.0 supports the use of DANE for enforcing SMTP security. It is strictly required that PowerMTA is to be configured with a DNS resolver that supports DNSSEC. If the new per-domain directive [use-dane](#) is set to true, PowerMTA will lookup the remote site's STARTTLS preferences (through DNS resolution) and use that information to influence whether STARTTLS is established or not with the remote site prior to transferring email to it. If the remote site requires the use of STARTTLS, PowerMTA will only transfer email to the remote site if STARTTLS was successful in establishing an encrypted channel to the remote site. And, in accordance with the DANE standard, PowerMTA will attempt to verify the certificate presented by the remote host during STARTTLS based on the fingerprints it obtained through DNS DANE records. If the remote site is strict about the use of STARTTLS, and PowerMTA could not perform STARTTLS successfully (for e.g. PowerMTA failed to verify that the fingerprint of the certificate presented by the remote site matches the fingerprint of the certificate specified in the DANE DNS records), PowerMTA will not transfer email to the remote server. Email is held in the queue until PowerMTA can successfully establish STARTTLS successfully in accordance with DANE protocol with the remote site.

See RFC 7672 for more descriptions on how certificate matching is specified via the use of DNS records. Please contact Port25 support if you have any further questions about DANE protocol.

## 11 Accounting Files and Statistics

### 11.1 Introduction

The accounting file can be written in CSV format to a file or streamed to pipe.

### 11.2 The Accounting File

For efficiency and ease of use, PowerMTA writes accounting information to a separate file rather than to its logging file.

For ease of use, the accounting file is a CSV (Comma Separated Value) file.

There are 8 types of records available:

- *Successful delivered* records which provide the data for all messages successfully transferred to the destination domain mail servers.
- *Bounced* records which provide the data for all reports PowerMTA generated and delivered, including bounces for messages that PowerMTA could not deliver.
- *Transient error* records which show each unsuccessful delivery attempt of an individual message. These errors are for individual recipients.
- *Transient queue error* records for queue-level errors.
- *Inbound recipients* records which show each successful submission of an individual message to PowerMTA.
- *Feedback loop* records can be logged if PowerMTA has been configured with a <feedback-loop-processor>
- *Remote bounce/status* records can be logged if PowerMTA has been configured with a <bounce-processor>

All records are written to the accounting file at the time the message or report is delivered, so they will be ordered by delivery time.

#### 11.2.1 Accounting records

PowerMTA writes an accounting record per recipient delivered. These various records can be stored in all in one file, or in separate files by record type. The CSV field type is used to determine one of four possible values for the file. The various types are "d" for delivered, "b" for bounced, or "t" for transient, "tq" for transient-queue, and "r" for received, "f" for feedback loop records and "rb"/"rs" for remote bounce or remote status records. By default, only "d" and "b" records are recorded.

#### 11.2.2 "Successful delivered" Records

The following are all the available fields for the CSV file that can be configured with "d" or "delivery" record type.

| CSV Accounting File (record-fields) | Description                                                                                                                                                                                                                            |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type                                | Type of record ("d" or "delivery") (Note: this field is always logged and cannot be disabled or changed).                                                                                                                              |
| timeLogged                          | Time the record was delivered and logged to the accounting file                                                                                                                                                                        |
| timeQueued                          | Time message was queued to disk                                                                                                                                                                                                        |
| timeFirstAttempt                    | Time message delivery was first attempted                                                                                                                                                                                              |
| totalSecondsQueued                  | the total amount of time in seconds, between reception to delivery                                                                                                                                                                     |
| orig                                | originator (from MAIL FROM:<x>)                                                                                                                                                                                                        |
| rcpt                                | recipient (RCPT TO:<x>) being reported                                                                                                                                                                                                 |
| orcpt                               | original recipient (from RCPT TO: . . . ORCPT=x), immediately following the recipient to which it refers                                                                                                                               |
| dsnAction                           | DSN action for the recipient to which it refers (Relayed means the message was passed on to another MTA, one that doesn't support the DSN extensions. Delivered means the MTA that issued the report actually did the final delivery.) |
| dsnStatus                           | DSN status for the recipient to which it refers                                                                                                                                                                                        |
| dsnDiag                             | DSN diagnostic string for the recipient to which it refers                                                                                                                                                                             |
| dsnMta                              | DSN remote MTA for the recipient to which it refers                                                                                                                                                                                    |
| bounceCat                           | likely category of the bounce (see <a href="#">Bounce Categories</a> ), following the recipient which it refers                                                                                                                        |
| srcType                             | source type from which the message was received, either <code>api</code> or <code>smtp</code>                                                                                                                                          |
| srcMta                              | source from which the message was received. the MTA name (from the HELO/EHLO command) for messages received through SMTP                                                                                                               |
| dlvType                             | delivery method, one of <code>smtp</code> , <code>pipe</code> , <code>discard</code> , or <code>file</code>                                                                                                                            |
| dlvSourceIp                         | local IP address PowerMTA used for delivery                                                                                                                                                                                            |
| dlvDestinationIp                    | IP address of the mailer to which the report/message was delivered                                                                                                                                                                     |
| dlvEsmtplibAvailable                | SMTP extensions supported by receiving mailer                                                                                                                                                                                          |

| CSV Accounting File (record-fields) | Description                                                                          |
|-------------------------------------|--------------------------------------------------------------------------------------|
| dlvSize                             | report/message size in bytes                                                         |
| vmta                                | VirtualMTA selected for this message, if any                                         |
| jobId                               | job ID for the message, if any                                                       |
| envId                               | envelope Id, from MAIL FROM: . . . ENVID=x (present only if specified)               |
| queue                               | domain/vmta used for delivery (remote status only)                                   |
| vmtaPool                            | VirtualMTA pool selected for this message, if any (remote status only)               |
| header_XXX                          | for each custom header included where XXX is the name of the header, if any          |
| userString                          | String added to accounting file as defined by the <acct-file> directive user-string. |
| dlvTlsProtocol                      | The TLS version that was negotiated with the remote end during STARTTLS              |
| dlvTlsCipher                        | The TLS cipher that was negotiated with the remote end during STARTTLS               |
| rcvSmtplibUser                      | The SMTP authenticated user that submitted the message                               |

Sample accounting entry for a delivery record:

```
d,2019-02-08 21:54:05+0000,2019-02-08 21:54:05+0000,AA-D=1181.com@bounce.bar.com,D@1181.com,,relayed,2.0.0
(success),smtp;250 DATA ok,foo.int.port25.com (10.1.2.3),,api,foo.int.port25.com
(127.0.0.1),smtp,10.1.2.3,10.1.2.3,"ENHANCEDSTATUSCODES,8BITIME,SMTPUTF8",41159,vmta48,3oBID29546016861868,,1181.com/vmta48
```

### 11.2.3 “Bounced” Records

The following are all the available fields for the CSV file that can be configured with “b” or “bounced” record type.

| CSV Accounting File (record-fields) | Description                                                                                                     |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| type                                | Type of record (“b” or “bounce”) (Note: this field is always logged and cannot be disabled or changed).         |
| timeLogged                          | Time the record was delivered and logged to the accounting file                                                 |
| timeQueued                          | Time message was queued to disk                                                                                 |
| timeFirstAttempt                    | Time message delivery was first attempted                                                                       |
| totalSecondsQueued                  | the total amount of time in seconds, between reception to delivery                                              |
| orig                                | originator (from MAIL FROM: <x>)                                                                                |
| rcpt                                | recipient (RCPT TO: <x>) being reported                                                                         |
| orcpt                               | original recipient (from RCPT TO: . . . ORCPT=x), immediately following the recipient to which it refers        |
| dsnAction                           | DSN action for the recipient to which it refers                                                                 |
| dsnStatus                           | DSN status for the recipient to which it refers                                                                 |
| dsnDiag                             | DSN diagnostic string for the recipient to which it refers                                                      |
| dsnMta                              | DSN remote MTA for the recipient to which it refers                                                             |
| bounceCat                           | likely category of the bounce (see <a href="#">Bounce Categories</a> ), following the recipient which it refers |
| srcType                             | source type from which the message was received, either 'api' or 'smtp'                                         |
| srcMta                              | No data present in Bounce Record                                                                                |
| dlvType                             | No data present in Bounce Record                                                                                |
| dlvSourceIp                         | No data present in Bounce Record                                                                                |
| dlvDestinationIp                    | No data present in Bounce Record                                                                                |
| dlvEsmtplibAvailable                | No data present in Bounce Record                                                                                |
| dlvSize                             | Deprecated. No data present in Bounce Record                                                                    |
| vmta                                | VirtualMTA selected for this message, if any                                                                    |
| jobId                               | job ID for the message, if any                                                                                  |
| envId                               | envelope Id, from MAIL FROM: . . . ENVID=x (present only if specified)                                          |
| queue                               | domain/vmta used for delivery (remote status only)                                                              |
| vmtaPool                            | VirtualMTA pool selected for this message, if any (remote status only)                                          |
| header_XXX                          | for each custom header included where XXX is the name of the header, if any                                     |
| userString                          | String added to accounting file as defined by the <acct-file> directive user-string.                            |
| dlvTlsProtocol                      | The TLS version that was negotiated with the remote end during STARTTLS                                         |
| dlvTlsCipher                        | The TLS cipher that was negotiated with the remote end during STARTTLS                                          |
| rcvSmtplibUser                      | The SMTP authenticated user that submitted the message                                                          |

Sample accounting entry for a bounce record:

```
b,2019-02-08 21:53:41+0000,2019-02-08 21:53:41+0000,CB-A=yahoo.com@bounce.bar.com,A@yahoo.com,,failed,5.7.1
(delivery not authorized),smtp;554 5.7.1 Denied - Spam Suspected -
http://www.w357sfhsjh873.net/services/spamwatch/blocked.php,foo.int.bar.com (10.1.2.3),spam-
related,api,foo.int.bar.com
(127.0.0.1),smtp,10.1.2.3,10.1.2.3,"ENHANCEDSTATUSCODES,8BITIME,SMTPUTF8",,vmta18,,yahoo.rollup/vmta18
```

### 11.2.4 “Transient error” Records

There are times when sending emails may fail. To help identify these issues, PowerMTA has the ability to show transient errors per message. This logging is handled with the record type “t”.

The following are all the available fields for the CSV file that can be configured with “t” or “transient” record type.

| CSV Accounting File (record-fields) | Description                                                                                                     |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| type                                | Type of record (“t” or “transient”) (Note: this field is always logged and cannot be disabled or changed).      |
| timeLogged                          | Time the record was delivered and logged to the accounting file                                                 |
| timeQueued                          | Time message was queued to disk                                                                                 |
| timeFirstAttempt                    | Time message delivery was first attempted                                                                       |
| totalSecondsQueued                  | the total amount of time in seconds, between reception to delivery                                              |
| orig                                | originator (from MAIL FROM: <x>)                                                                                |
| rcpt                                | recipient (RCPT TO: <x>) being reported                                                                         |
| orcpt                               | original recipient (from RCPT TO: . . . ORCPT=x), immediately following the recipient to which it refers        |
| dsnAction                           | DSN action for the recipient to which it refers                                                                 |
| dsnStatus                           | DSN status for the recipient to which it refers                                                                 |
| dsnDiag                             | DSN diagnostic string for the recipient to which it refers                                                      |
| dsnMta                              | DSN remote MTA for the recipient to which it refers                                                             |
| bounceCat                           | likely category of the bounce (see <a href="#">Bounce Categories</a> ), following the recipient which it refers |
| srcType                             | source type from which the message was received, either 'api' or 'smtp'                                         |

| CSV Accounting File (record-fields) | Description                                                                                                              |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| srcMta                              | source from which the message was received. the MTA name (from the HELO/EHLO command) for messages received through SMTP |
| dlvType                             | delivery method, one of "smtp", "pipe", "discard", or "file"                                                             |
| dlvSourceIp                         | local IP address PowerMTA used for delivery                                                                              |
| dlvDestinationIp                    | IP address of the mailer to which the report/message was delivered                                                       |
| dlvEsmtpAvailable                   | SMTP extensions supported by receiving mailer                                                                            |
| dlvSize                             | Deprecated. report/message size in bytes                                                                                 |
| vmta                                | VirtualMTA selected for this message, if any                                                                             |
| jobId                               | job ID for the message, if any                                                                                           |
| envId                               | envelope Id, from MAIL FROM: . . . ENVID=x (present only if specified)                                                   |
| queue                               | domain/vmta used for delivery (remote status only)                                                                       |
| vmtaPool                            | VirtualMTA pool selected for this message, if any (remote status only)                                                   |
| header_XXX                          | for each custom header included where XXX is the name of the header, if any                                              |
| userString                          | String added to accounting file as defined by the <acct-file> directive user-string.                                     |
| dlvTlsProtocol                      | The TLS version that was negotiated with the remote end during STARTTLS                                                  |
| dlvTlsCipher                        | The TLS cipher that was negotiated with the remote end during STARTTLS                                                   |
| rcvSmtplibUser                      | The SMTP authenticated user that submitted the message                                                                   |

Sample accounting entry for a transient record:

```
t,2019-02-08 21:53:54+0000,2019-02-08 21:53:54+0000,DB.AD.G.AA.AB-
DA=yahoo.fr@bounce.bar.com,DA@yahoo.fr,,delayed,4.7.0 (other or undefined security status),smtp;421 4.7.0
mx002.roc2.blueit.com Error: too many errors,foo.int.bar.com (10.1.2.165),policy-
related,api,foo.int.bar.com
(127.0.0.1),smtp,10.59.14.69,10.1.2.165,"ENHANCEDSTATUSCODES,8BITMIME,SMTPUTF8",,vmta21,,,yahoo.rollup/vmta21,
```

### 11.2.5 "Transient Queue" Records

There are times when sending emails that an entire given queue may fail. It may be problems with a particular outbound IP address, or maybe one domain is currently offline. To help identify these issues, PowerMTA has not only the ability to show transient recipient errors per message, but also the same information on a per queue basis. This logging is handled with the record type "tq". The records when using "tq" are slightly different than those mentioned above. The fields available are as follows:

| CSV Accounting File (record-fields) | Description                                                                                                       |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| type                                | Type of record ("tq" or "transient queue") (Note: this field is always logged and cannot be disabled or changed). |
| timeLogged                          | current time                                                                                                      |
| queue                               | queue name                                                                                                        |
| vmta                                | VirtualMTA name                                                                                                   |
| dsnStatus                           | DSN status code for failure                                                                                       |
| dsnDiag                             | DSN diagnostic string for failure                                                                                 |
| dsnMta                              | remote MTA, if applicable                                                                                         |
| dlvType                             | delivery type                                                                                                     |
| dlvSourceIp                         | IP address from which the connection was made/attempted                                                           |
| dlvDestinationIp                    | IP address to which the connection was made/attempted                                                             |
| userString                          | String added to accounting file as defined by the <acct-file> directive user-string.                              |

Sample accounting entry for a transient queue record:

```
tq,2019-02-08 21:53:39+0000,4.4.1 (no answer from host),,mx.xn--bcher-kva.de
(87.98.150.101),smtp,0.0.0.0,87.98.150.101,vmta29,bucher.de/vmta29
```

### 11.2.6 "Received" Records

PowerMTA can log messages in the accounting file during the injection process. This logging is handled with the record type "r". The fields available are as follows:

| CSV Accounting File (record-fields) | Description                                                                                                              |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| type                                | Type of record ("r" or "received") (Note: this field is always logged and cannot be disabled or changed).                |
| timeLogged                          | current time                                                                                                             |
| orig                                | originator (from MAIL FROM: <x>)                                                                                         |
| rcpt                                | recipient (RCPT TO: <x>) being reported                                                                                  |
| orcpt                               | original recipient (from RCPT TO: . . . ORCPT=x), immediately following the recipient to which it refers                 |
| srcType                             | source type from which the message was received, either api or smtp                                                      |
| srcMta                              | source from which the message was received. the MTA name (from the HELO/EHLO command) for messages received through SMTP |
| rcvSourceIp                         | IP address of the server connecting to PowerMTA                                                                          |
| rcvDestinationIp                    | IP address PowerMTA used to handle the connection                                                                        |
| vmta                                | VirtualMTA selected for this message, if any.                                                                            |
| vmtaPool                            | VirtualMTA pool selected for this message, if any.                                                                       |
| jobId                               | job ID for the message, if any                                                                                           |
| envId                               | envelope Id, from MAIL FROM: . . . ENVID=x (present only if specified)                                                   |
| header_XXX                          | for each custom header included where XXX is the name of the header, if any                                              |
| userString                          | String added to accounting file as defined by the <acct-file> directive user-string.                                     |
| rcvSmtplibUser                      | The SMTP authenticated user that submitted the message                                                                   |

Note:

In PowerMTA version 4.5r18 and below, if the message was submitted to a pool, the vmta field will contain the pool name instead of the eventual virtual MTA (of that pool) that the message was placed into.

Starting from PowerMTA 4.5r19, if the message was submitted to a pool, the vmta field will contain the actual VMTA name that the message was entered into. The pool that the message was submitted to can be found in the vmtaPool field.

Sample accounting entry for a received record:

```
r,2010-04-17 16:19:57-0400,test@port25.com,test@port25.com,,smtp,tesFeeder
(192.168.25.56),192.168.25.56,192.168.25.11,vmta1,job2,envId1234
```

### 11.2.7 “Remote Bounce” & “Remote Status” Records

PowerMTA can log remote (asynchronous) bounces in the accounting file and remote status reports. This logging is handled with the record type “rb” for remote bounce and “rs” for reporting other kinds of remote status (used for DSN positive and “delayed” status notifications). The fields available are as follows:

| CSV Accounting File (record-fields) | Description                                                                                                                                      |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| type                                | Type of record “rb” or “rs” (Note: this field is always logged and cannot be disabled or changed).                                               |
| timeLogged                          | current time                                                                                                                                     |
| orig                                | originator address (of original email)                                                                                                           |
| rcpt                                | recipient address (of original email)                                                                                                            |
| orcpt                               | original recipient (from RCPT TO: . . . ORCPT=x), immediately following the recipient to which it refers                                         |
| dsnAction                           | DSN action (remote status only)                                                                                                                  |
| dsnStatus                           | DSN status                                                                                                                                       |
| dsnDiag                             | DSN diagnostic                                                                                                                                   |
| dsnMta                              | DSN remote MTA                                                                                                                                   |
| dsnReportingMta                     | DSN reporting MTA (remote bounce only)                                                                                                           |
| bounceCat                           | bounce category                                                                                                                                  |
| srcType                             | source type from which the message was received, either api or smtp                                                                              |
| srcMta                              | source from which the message was received, the MTA name (from the HELO/EHLO command) for messages received through SMTP                         |
| envId                               | envelope Id                                                                                                                                      |
| format                              | either dsn or qmail or mxl or brightmail or interscan, or the name of the file containing the body-pattern-file pattern that matched this report |

Sample accounting entry for a remote bounce record:

```
rb,2010-12-01 14:24:44-0500,bounce@bounce.com,lksjflkj@somedomain.com,,5.1.1 (bad destination mailbox address), "smtp;550 5.1.1 no such local recipient: <lksjflkj@somedomain.com> in ""RCPT TO: <lksjflkj@somedomain.com>""",mail.remotedomain.com (169.3.169.30),my.local.server.com,bad-mailbox,smtp,my.local.server.com (192.168.0.150),
```

### 11.2.8 “Feedback Loop” Records

PowerMTA can log feedback loop reports in the accounting file. This logging is handled with the record type “f”. The records when using “f” are slightly different than those mentioned above. The fields available are as follows:

| CSV Accounting File (record-fields) | Description                                                                                           |
|-------------------------------------|-------------------------------------------------------------------------------------------------------|
| type                                | Type of record “F” (Note: this field is always logged and cannot be disabled or changed).             |
| timeLogged                          | Current time                                                                                          |
| repSourceIp                         | the IP of PowerMTA that received the message                                                          |
| feedbackType                        | mapped to ARF's Feedback-Type field                                                                   |
| format                              | arf or jmrp or the name of the file containing the body-pattern-file pattern that matched this report |
| userAgent                           | name & version of the program that generated the report                                               |
| envid                               | envelope ID (of original email)                                                                       |
| orig                                | originator address (of original email)                                                                |
| rcpt                                | recipient address (of original email)                                                                 |
| reportingMTA                        | the name of the MTA generating this feedback report                                                   |
| dlvSourceIp                         | address of MTA from which the message was received                                                    |
| reportedDomain                      | the domain about which the feedback report was generated                                              |
| header <name>                       | header extracted from original email                                                                  |
| arf <name>                          | field from ARF feedback-report                                                                        |

Sample accounting entry for a FBL record:

```
f,2010-04-12 11:40:37-0400,arf,AOL SComp,,,,mail1.domain.com,169.63.151.30
```

### 11.2.9 Encoding

As much as possible, standard CSV encodings are used for all fields, with most characters being as-is in the output. Time stamps are normally formatted as “MM/DD/YYYY HH:MM:SS+UTCoffset” but can be changed to UNIX time\_t, i.e. the number of seconds since midnight on January 1, 1970.

The format of the file is CSV (comma-separated values), UTF-8 encoded, as described in RFC 4180, with a header line at the beginning of the file describing what field is in which column. Here is the format from RFC 4180 in short: columns are separated by commas and records terminated with CRLF. Values may be quoted with double quotes (“”), and must be quoted if its value contains a comma, a double quote or CRLF. A double quote occurring within double quotes is repeated, like in:

```
two inches,"2""
```

To accommodate writing various record types in the same file, the first column is always the record type. If an accounting file is configured (or defaulted) to have multiple record types, all the possible columns from the selected records are included. For example, if a file was configured as:

```
records d, b
record-fields delivery timeLogged,orig,rcpt,dlvSourceIp
record-fields bounce timeLogged,orig,rcpt,bounceCat
```

the resulting file (including a few record of each type) might look like:

```
type,timeLogged,orig,rcpt,dlvSourceIp,bounceCat
d,1178807903,orig@port25.com,user1@domain.com,10.0.0.1,
d,1178807905,orig@port25.com,user2@domain.com,10.0.0.1,
b,1178807915,orig@port25.com,user3@domain.com,,policy-related
b,1178807915,orig@port25.com,user4@domain.com,,policy-related
```

Note that the dlvSourceIp field was omitted (empty field at the end) for the ‘b’ record, and that the bounceCat was omitted in the ‘d’ record, because they were not configured.

## 11.3 Configuring the accounting file

Use `<acct-file>` tag in the config file to enable accounting files. The tag takes one parameter - either the location of the CSV file or the location of the executable program to which PowerMTA should pipe the accounting records. If using pipe, the parameter should be preceded by a "|". It may be required to quote the acct file path if it contains spaces (like `<acct-file "c:\my stuff\acctproc.exe">`) or if arguments are passed (like `<acct-file "/bin/foo.pl -x -a -z">`). See [Examples](#) below.

In the current pipe implementation, if PowerMTA runs into a problem writing to the pipe, it gives up on it and won't retry either until the next `pmta reload` command is run, or after a certain predefined time limit. With this, it will be necessary for customers to configure PowerMTA to create a corresponding accounting data file as well, in order to have the data to recover, if there are problems with the customer's pipe application.

In the event of a broken pipe, PowerMTA writes:

```
Error TYPE "CMD": DETAILS
```

to the logging file, where TYPE is one of "starting", "writing to" or "executing", CMD is the configured command, and DETAILS is a string with details on what went wrong. Each time a pipe is started, the first line written to it is a headers line with the various fields (like in the CSV files). It is only re-started upon a `pmta reload`.

### 11.3.1 Examples

The accounting file is configured with the `<acct-file>` type directive with the date added as part of the file name. For example `<acct-file log/acct.csv>` yields a file name like `/pmta/log/acct-2007-09-05-0000.csv`. By default on a new installation, the settings will look similar to the following:

```
<acct-file log/acct.csv>
# move-to c:\myapp\pmta-acct # configure as fit for your application
move-interval 1d
max-size 250M # MB
</acct-file>
```

The data may also be written to pipe. This can be useful if a custom application is created in which the data would be written to a database in real time. PowerMTA instantiates the script only once at startup and then keeps passing data to the pipe as and when available. An example would look similar to the following:

```
<acct-file |C:\pmta\bin\customCode.exe --thisOption --thatOption>
</acct-file>
```

Multiple files can be used if needed. For example, it may be more convenient to have all bounces and deliveries in one file and transient reports in another file. An example of such would be similar to the following:

```
<acct-file log/acct-bd.csv>
  records b,d
  record-fields d *,!orcpt,header_myCustomHeader
  record-fields b *,!orcpt
</acct-file>

<acct-file log/acct-t.csv>
  records t,tq
</acct-file>
```

## 11.4 The pmtastats Accounting Statistics Application

PowerMTA comes with an accounting file parsing application, `pmtastats`, that calculates delivery throughput and general traffic statistics both on a global basis and/or on a per campaign or mailing basis.

If you wish to run the parsing application over more than one accounting file (for example, if the accounting file was rotated while sending a campaign), simply specify the various files within the same command. For example:

```
pmtastats [global options] [statistic [options] ..] acct.csv acct-1.csv
```

### 11.4.1 pmtastats Options

`pmtastats` has the following options, as shown by the `--help` output:

```
PowerMTA(TM) v5.0r1 statistics program
Copyright(c) 1999-2019, Port25 Solutions, Inc. All Rights Reserved.
```

```
Usage: bin/pmtastats [global options] [statistic [options] ...] file ...
```

One or more statistics can be specified, each with its own options. Use with one or more files, in either binary, CSV or XML format. '-' can be used for reading from standard input.

examples:

```
bin/pmtastats vmta-summary vmta-time-breakdown acct*
bin/pmtastats bounce-categories --bounceCat all acct*
bin/pmtastats --vmta vmta1 top-rates delivery-times acct*
bin/pmtastats --output html vmta-time-breakdown acct* > report.html
```

global options:

```
--help
  display this help and exit
--version
  display version information and exit
--quiet
  do not display progress information
--output <format>
  selects the output format (text, html; default: text)
--envId <id>
--from <originator>
--toDomain <domain>
--jobId <id>
--vmta <vmta>
--rcvSmtUser <username>
--header <name> <content>
  filter input: use only records whose envelope ID, originator,
  recipient domain, job ID, VirtualMTA, or the given header
  starts with the given string, respectively
  (default: no filtering)
--last DdHhMmSs
```

```

        filter input: use only records written in the last D days, H hours,
        M minutes and S seconds. Each part of the time specification can
        be omitted, such as in "3h15m" for 3 hours, 15 minutes.
        (default: no filtering)
--byEnvId
--byJobId
--byFrom
--byVmta
--byHeader <header>
    statistics split by envelope ID, job ID, originator, VirtualMTA, or a
    given header, respectively.
    Only one of these can be used.
    For --byHeader, specify the header.
    (default: statistics not split)
--splitLimit <num>
    set the maximum number of statistics a statistic can be split into
    (default: 5000)

--topDomains <num>
    display top <num> domains (passed on to all statistics)

"bounce-categories" statistic
Shows most frequent bounce categories.
Options:
--topCategories <num>
    display top <num> bounce categories (default: 10)
--bounceCat <option>
    selects categories for detailed display. Possible options are:
        <num> - top <num> categories
        <category> - category name
        'all' - all available categories
    (default: spam-related)
--byDsn <num>
    display bounce category detail for the top <num> DSN codes (default: 10)
--byDomain <num>
    display bounce category detail for the top <num> domains (default: 10)
--byDsnAndDomain <num>
    display bounce category detail for the top <num> DSN code / domain
    combinations (default: 10)

"delivery-times" statistic
Shows times and recipients / time from queueing to delivery.
Options:
    (none)

"message-counts" statistic
Total counts of recipients and volume for messages and bounces.
Options:
    (none)

"top-bounce" statistic
Top bounce domains, aggregated by DSN code (reason).
Options:
--topDomains <num>
    display bounce statistic for top <num> domains (default: 10)
--topStatusCodes <num>
    display bounce statistics for top <num> status codes (default: 10)

"top-domains" statistic
Domains with the most recipients, bounces, and delivery times.
Options:
--topDomains <num>
    display top <num> domains (default: 10)
--noDeliveryTimes
    omits list of domains with the highest average delivery times
    (default: includes it)

"top-rates" statistic
Data transfer and recipient rates over time.
Options:
    (none)

"vmta-summary" statistic
High-level Virtual MTA traffic summary.
Options:
--byDate
    split records by date (default: no splitting done)

"vmta-time-breakdown" statistic
Daily and hourly recipients and bounces for each Virtual MTA.
Options:
--noHourly
    do not include hourly breakdown (default: include it)

"vmta-top-bounce" statistic
Bounces for individual Virtual MTAs.
Options:
--topBounced <num>
    show top <num> domains by bounces (default: 20)
--byDate
    split records by date (default: no splitting done)

shortcut options:
--all
    create all available statistics with default options
--acctstats
    create statistics similar to what acctstats did

```

Note: On Windows, you can also use the "/" character instead of "--" (but "--" works on all platforms).

The following is a list of the various fields available, and which reports return those fields.

|              | bounce-categories | delivery-times | just-delivery-times | message-counts | top-bounce | top-domains | top-rates | vmta-summary | vmta-time-breakdown | vmta-top-bounce |
|--------------|-------------------|----------------|---------------------|----------------|------------|-------------|-----------|--------------|---------------------|-----------------|
| bounceCat    | *                 |                |                     |                |            |             |           |              |                     |                 |
| dlvSourceIp  |                   |                |                     |                |            |             |           | *            | *                   | *               |
| dsnDiag      | *                 |                |                     | *              | *          |             |           | *            | *                   | *               |
| dsnStatus    | *                 |                |                     |                | *          |             |           |              |                     |                 |
| envId        |                   |                |                     |                |            |             |           |              |                     |                 |
| header_<xyz> |                   |                |                     |                |            |             |           |              |                     |                 |
| jobId        |                   |                |                     |                |            |             |           |              |                     |                 |
| nRcpt        | *                 | *              |                     |                | *          |             | *         | *            | *                   | *               |
| orig         |                   |                |                     |                |            |             |           |              |                     |                 |
| rcpt         |                   |                |                     | *              | *          | *           |           |              |                     | *               |
| size         |                   |                |                     | *              |            |             | *         |              |                     |                 |
| timeLogged   |                   | *              |                     |                | *          |             | *         | *            | *                   | *               |
| timeQueued   |                   | *              | *                   |                | *          |             |           | *            | *                   | *               |
| vmta         |                   |                |                     |                |            |             |           | *            | *                   | *               |

## 11.4.2 pmtastats Report Breakdown

Given the various reports available, different data will or will not be included in each report based on the options used. Here is a sample of the reports available and their output. Multiple reports can be run at the same time.

### 11.4.2.1 Time Frame

|                   |                     |                     |
|-------------------|---------------------|---------------------|
| Time Frame:       | first               | last                |
|                   | -----               | -----               |
| received          | 2018-11-06 18:07:52 | 2018-11-06 19:29:33 |
| delivered/bounced | 2018-11-06 19:12:00 | 2018-11-06 19:29:33 |

This section is always generated for each report type, and shows the first and the last receipt time stamps and the first and the last delivery time stamps of the applicable messages. The scope of the output is based on the type of report specified, i.e., global or by originator or by envelope ID, and such can be used to calculate a campaign duration time. In this example, the mailing duration would be 13 hours 56 minutes and 35 seconds, which is the difference between when the first message was received (2007-01-15 14:19:11) and the last message was delivered (2007-01-17 03:15:46).

### 11.4.2.2 Bounce Categories

|                                                       |               |                                                                                                                                                                                                                                |               |
|-------------------------------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Total Bounce Breakdown By Category:                   |               |                                                                                                                                                                                                                                |               |
| Bounces (across 12 categories):                       |               |                                                                                                                                                                                                                                |               |
| total bounced                                         | 5.X.X bounces | 4.X.X bounces                                                                                                                                                                                                                  |               |
| -----                                                 | -----         | -----                                                                                                                                                                                                                          |               |
| 9,321                                                 | 8,994         | 327                                                                                                                                                                                                                            |               |
| bounced                                               | %bounced      | bounce category                                                                                                                                                                                                                |               |
| -----                                                 | -----         | -----                                                                                                                                                                                                                          |               |
| 2,748                                                 | 29.5%         | spam-related                                                                                                                                                                                                                   |               |
| 1,978                                                 | 21.2%         | policy-related                                                                                                                                                                                                                 |               |
| 1,628                                                 | 17.5%         | bad-mailbox                                                                                                                                                                                                                    |               |
| 812                                                   | 8.7%          | quota-issues                                                                                                                                                                                                                   |               |
| 482                                                   | 5.2%          | other                                                                                                                                                                                                                          |               |
| 451                                                   | 4.8%          | invalid-sender                                                                                                                                                                                                                 |               |
| 366                                                   | 3.9%          | inactive-mailbox                                                                                                                                                                                                               |               |
| 327                                                   | 3.5%          | no-answer-from-host                                                                                                                                                                                                            |               |
| 259                                                   | 2.8%          | relaying-issues                                                                                                                                                                                                                |               |
| 160                                                   | 1.7%          | protocol-errors                                                                                                                                                                                                                |               |
| Bounce Category Detail (spam-related)                 |               |                                                                                                                                                                                                                                |               |
| bounce category                                       | total bounced | 5.X.X bounces                                                                                                                                                                                                                  | 4.X.X bounces |
| -----                                                 | -----         | -----                                                                                                                                                                                                                          | -----         |
| spam-related                                          | 2,748         | 2,748                                                                                                                                                                                                                          | 0             |
| Sorted By DSN Code (Reason)                           |               |                                                                                                                                                                                                                                |               |
| bounced                                               | % in category | dsn code (reason)                                                                                                                                                                                                              |               |
| -----                                                 | -----         | -----                                                                                                                                                                                                                          |               |
| 43                                                    | 1.6%          | 5.7.1 (delivery not authorized); 554 5.7.1 Blacklisted by black.uribl.com                                                                                                                                                      |               |
| Contact the postmaster of this domain for resolution. |               |                                                                                                                                                                                                                                |               |
| 38                                                    | 1.4%          | 5.7.1 (delivery not authorized); 554 5.7.1 Blacklisted by Contact the postmaster of this domain for resolution. This attempt has been logged.                                                                                  |               |
| 35                                                    | 1.3%          | 5.7.1 (delivery not authorized); 554 5.7.1 Blacklisted by cim-10.net.multi.surbl.org Contact the postmaster of this domain for resolution. This attempt has been logged.                                                       |               |
| 32                                                    | 1.2%          | 5.7.1 (delivery not authorized); 554 5.7.1 Denied - Spam Suspected - http://<RCPT_SERVER>/services/spamwatch/blocked.php                                                                                                       |               |
| 32                                                    | 1.2%          | 5.7.1 (delivery not authorized); 554 5.7.1 Rejected... Message identified as UBE (spam). If you feel you have received this message in error CALL the intended recipient and ask them to report the issue to support services. |               |
| 32                                                    | 1.2%          | 5.7.1 (delivery not authorized); 554 5.7.1 Your message was rejected as spam                                                                                                                                                   |               |
| 32                                                    | 1.2%          | 5.7.1 (delivery not authorized); 571 5.7.1 Message contains spam or virus :                                                                                                                                                    |               |
| 1585:<HEXDUMP> 5439                                   |               |                                                                                                                                                                                                                                |               |
| 31                                                    | 1.1%          | 5.7.1 (delivery not authorized); 554 5.7.1 Blacklisted by cim-10.net.black.uribl.com Contact the postmaster of this domain for resolution. This attempt has been logged.                                                       |               |
| 31                                                    | 1.1%          | 5.7.1 (delivery not authorized); 554 5.7.1 Message blocked due to spam content in the message                                                                                                                                  |               |
| 30                                                    | 1.1%          | 5.7.1 (delivery not authorized); 554 5.7.1 <SENDER_IPADDRESS>: on real-time blacklist b.barracudacentral.org                                                                                                                   |               |
| Sorted By Domain                                      |               |                                                                                                                                                                                                                                |               |
| domain                                                | bounced       | % in category                                                                                                                                                                                                                  |               |
| -----                                                 | -----         | -----                                                                                                                                                                                                                          |               |
| yahoo.com                                             | 149           | 5.4%                                                                                                                                                                                                                           |               |

|                                    |    |      |
|------------------------------------|----|------|
| hotmail.fr                         | 94 | 3.4% |
| 中文邮箱.网络                            | 91 | 3.3% |
| hotmail.es                         | 71 | 2.6% |
| öbb.de                             | 61 | 2.2% |
| yahoo.fr                           | 58 | 2.1% |
| yahoo.es                           | 56 | 2.0% |
| blåbærsyltetøy.gulbrandsen.priv.no | 55 | 2.0% |
| live.ca                            | 46 | 1.7% |
| live.nl                            | 45 | 1.6% |

Sorted By DSN Code (Reason) By Domain

| domain                                                                   | bounced | % in category | dsn code (reason)                                                                                                                                                                                                              |
|--------------------------------------------------------------------------|---------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 中文邮箱.网络                                                                  | 7       | 0.3%          | 5.0.0 (undefined status); 553 5.0.0 We do not accept mail from spammers ([<SENDER_IPADDRESS>][<SENDER_IPADDRESS>])                                                                                                             |
| yahoo.com                                                                | 7       | 0.3%          | 5.7.1 (delivery not authorized); 554 5.7.1 Message blocked due to spam content in the message                                                                                                                                  |
| yahoo.com                                                                | 7       | 0.3%          | 5.7.1 (delivery not authorized); 571 5.7.1 Message contains spam or virus : 1585:<HEXDUMP> 5439                                                                                                                                |
| blåbærsyltetøy.gulbrandsen.priv.no                                       | 5       | 0.2%          | 5.0.0 (undefined status); 551 5.0.0 message classified as spam (RefID: str=0001.<HEXDUMP>.<HEXDUMP>.00C9,ss=4,fgs=0)                                                                                                           |
| hotmail.fr                                                               | 5       | 0.2%          | 5.7.1 (delivery not authorized); 554 5.7.1 Message rejected, SpamAssassin scored: 9.674 points                                                                                                                                 |
| TESTS=HTML_MESSAGE,IISDATED_4_25_2009,MIME_HTML_ONLY,NO_216_174_43CLASSC |         |               |                                                                                                                                                                                                                                |
| yahoo.com                                                                | 5       | 0.2%          | 5.7.1 (delivery not authorized); 554 5.7.1 Rejected... Message identified as UBE (spam). If you feel you have received this message in error CALL the intended recipient and ask them to report the issue to support services. |
| yahoo.es                                                                 | 4       | 0.1%          | 5.0.0 (undefined status); 553 5.0.0 We do not accept mail from spammers ([<SENDER_IPADDRESS>][<SENDER_IPADDRESS>])                                                                                                             |
| yahoo.fr                                                                 | 4       | 0.1%          | 5.3.0 (unknown mail system-related status); 553 5.3.0 <<RCPT_USER>@<yahoo.fr>>... Rejected by Iserv RBL. If this is an error, please contact our customer service department at 1-888-644-7378. - <SENDER_IPADDRESS>           |
| msn.com                                                                  | 4       | 0.1%          | 5.7.1 (delivery not authorized); 550 5.7.1 mail from <SENDER_IPADDRESS> rejected for spam/virus content                                                                                                                        |
| blåbærsyltetøy.gulbrandsen.priv.no                                       | 4       | 0.1%          | 5.7.1 (delivery not authorized); 550 5.7.1 rejected as confirmed spam                                                                                                                                                          |

This report list the top level bounce information. The scope of the output is based on the type of report specified, and the above example is done using no additional options.

Note that the report is based only on the messages that have already been handled by PowerMTA. If the report is generated while PowerMTA is still attempting to deliver some of the messages, these will not be included in the report.

#### 11.4.2.3 Delivery Times

Delivery Times And Numbers (total queue time):

Delivery times per percentage of recipients (successful deliveries only):

| recipients | avg. time | max. time | %rcpt. |
|------------|-----------|-----------|--------|
| 46,497     | 0:00:00   | 0:00:00   | 50%    |
| 55,797     | 0:00:00   | 0:00:00   | 60%    |
| 65,096     | 0:00:00   | 0:00:00   | 70%    |
| 74,396     | 0:00:00   | 0:00:00   | 80%    |
| 83,695     | 0:00:00   | 0:00:00   | 90%    |
| 88,345     | 0:00:00   | 0:00:01   | 95%    |
| 89,275     | 0:00:00   | 0:00:01   | 96%    |
| 90,205     | 0:00:00   | 0:00:01   | 97%    |
| 91,135     | 0:00:00   | 0:05:05   | 98%    |
| 92,065     | 0:00:06   | 0:10:00   | 99%    |
| 92,995     | 0:00:13   | 0:45:01   | 100%   |

Deliveries per time increment (successful deliveries only):

| interval (between) | histogram | recipients | %rcpt. |
|--------------------|-----------|------------|--------|
| 0 - 5 seconds      | *****     | 90,691     | 97.5%  |
| 6 - 30 seconds     |           | 0          | 0.0%   |
| 31 - 60 seconds    |           | 0          | 0.0%   |
| 1 - 2 minutes      |           | 48         | 0.1%   |
| 2 - 5 minutes      |           | 394        | 0.4%   |
| 5 - 10 minutes     |           | 1,222      | 1.3%   |
| 10 - 30 minutes    |           | 626        | 0.7%   |
| 30 - 60 minutes    |           | 14         | 0.0%   |
| 1 - 2 hours        |           | 0          | 0.0%   |
| 2 - 3 hours        |           | 0          | 0.0%   |
| 3 - 4 hours        |           | 0          | 0.0%   |
| 4 - 5 hours        |           | 0          | 0.0%   |
| 5 - 6 hours        |           | 0          | 0.0%   |
| 6 - 8 hours        |           | 0          | 0.0%   |
| 8 - 10 hours       |           | 0          | 0.0%   |
| 10 - 12 hours      |           | 0          | 0.0%   |
| after 12 hours     |           | 0          | 0.0%   |

Cumulative deliveries per time increment (successful deliveries only):

| interval          | histogram | recipients | %rcpt. |
|-------------------|-----------|------------|--------|
| within 5 seconds  | *****     | 90,691     | 97.5%  |
| within 30 seconds | *****     | 90,691     | 97.5%  |
| within 1 minute   | *****     | 90,691     | 97.5%  |
| within 2 minutes  | *****     | 90,739     | 97.6%  |
| within 5 minutes  | *****     | 91,133     | 98.0%  |
| within 10 minutes | *****     | 92,355     | 99.3%  |
| within 30 minutes | *****     | 92,981     | 100.0% |
| within 1 hour     | *****     | 92,995     | 100.0% |
| within 2 hours    | *****     | 92,995     | 100.0% |

|                 |       |        |        |
|-----------------|-------|--------|--------|
| within 3 hours  | ***** | 92,995 | 100.0% |
| within 4 hours  | ***** | 92,995 | 100.0% |
| within 5 hours  | ***** | 92,995 | 100.0% |
| within 6 hours  | ***** | 92,995 | 100.0% |
| within 8 hours  | ***** | 92,995 | 100.0% |
| within 10 hours | ***** | 92,995 | 100.0% |
| within 12 hours | ***** | 92,995 | 100.0% |
| after 12 hours  | ***** | 92,995 | 100.0% |

This section and respective outputs are always generated and the scope of the outputs are based on the type of report specified, i.e., global or by originator/envelope ID. They also are primarily based on the difference between each message's queue timestamp and transfer timestamp for only the messages that were successfully delivered by PowerMTA.

As mentioned in [Top Domains](#), for the avg. time and max. time calculations, each recipient is counted by itself, even if a message is transferred in multi-recipient SMTP transactions.

The avg. time is calculated by adding together the fastest n% transfer times (transfer timestamp - queue timestamp) and then dividing this sum by the number of recipients that make up n% of the total deliveries. For example, in the 80% fastest delivery line in the top output, one would read that it took an average of 21 seconds for each message to each recipient to be delivered once each was queued, for the fastest 80% of all transfer times (or to 7,504,928 recipients).

The max. time is the longest transfer time (transfer timestamp - queue timestamp) for one message from all of the fastest transfer times for the number of recipients. In the 80% fastest delivery case, one would read that the longest it took for a message to be transferred, for the fastest 80% of all transfer times (or to 7,504,928 recipients), was 4 minutes and 24 seconds. Conversely, all of the 80% messages were transferred in equal or less time. Note that this does not show total campaign duration time since these figures are only based on each message's own transfer time.

#### 11.4.2.4 Message Counts

Global Message Counts And Sizes:

Message counts:

|            | recipients | volume      |
|------------|------------|-------------|
|            | -----      | -----       |
| deliveries | 92,995     | 18,532.1 kb |
| bounces    | 9,321      |             |
| total      | 102,316    | 18,532.1 kb |

Message sizes:

| min. size | avg. size | max. size |
|-----------|-----------|-----------|
| -----     | -----     | -----     |
| 0.2 kb    | 0.2 kb    | 0.3 kb    |

This report list the basic numbers associated on the mailings being reported. Totals for delivered, bounced, and size of messages are included.

#### 11.4.2.5 Top Bounce

Top Bounce Domain Breakdowns:

| domain                                                                                | recipients | total bounced                                                                         | 5.X.X bounces | 4.X.X bounces |
|---------------------------------------------------------------------------------------|------------|---------------------------------------------------------------------------------------|---------------|---------------|
| -----                                                                                 | -----      | -----                                                                                 | -----         | -----         |
| yahoo.com                                                                             | 2,620      | 458                                                                                   | 458           | 0             |
| 中文邮箱.网络                                                                               | 1,577      | 342                                                                                   | 342           | 0             |
| noos.fr                                                                               | 257        | 257                                                                                   | 0             | 257           |
| hotmail.fr                                                                            | 846        | 232                                                                                   | 232           | 0             |
| öbb.de                                                                                | 1,643      | 194                                                                                   | 194           | 0             |
| yahoo.fr                                                                              | 737        | 183                                                                                   | 183           | 0             |
| yahoo.es                                                                              | 580        | 177                                                                                   | 177           | 0             |
| msn.com                                                                               | 668        | 172                                                                                   | 172           | 0             |
| blåbørsyltetø.gulbrandsen.priv.no                                                     | 960        | 166                                                                                   | 166           | 0             |
| hotmail.es                                                                            | 614        | 160                                                                                   | 160           | 0             |
| ...                                                                                   |            |                                                                                       |               |               |
| hotmail.fr                                                                            |            |                                                                                       |               |               |
| domain                                                                                | recipients | total bounced                                                                         | 5.X.X bounces | 4.X.X bounces |
| -----                                                                                 | -----      | -----                                                                                 | -----         | -----         |
| hotmail.fr                                                                            | 846        | 232                                                                                   | 232           | 0             |
| bounced                                                                               | %bounced   | dsn code (reason)                                                                     |               |               |
| -----                                                                                 | -----      | -----                                                                                 |               |               |
| 5                                                                                     | 2.2%       | 5.7.1 (delivery not authorized); 554 5.7.1 BRBL entry - see                           |               |               |
| <http://www.barracudanetworks.com/reputation/?pr=1&ip=<SENDER_IPADDRESS>>             |            |                                                                                       |               |               |
| 5                                                                                     | 2.2%       | 5.7.1 (delivery not authorized); 554 5.7.1 Command rejected                           |               |               |
| 5                                                                                     | 2.2%       | 5.7.1 (delivery not authorized); 554 5.7.1 Message rejected, SpamAssassin scored:     |               |               |
| 9.674 points TESTS=HTML_MESSAGE,IISDATED_4_25_2009,MIME_HTML_ONLY,NO_216_174_43CLASSC |            |                                                                                       |               |               |
| 4                                                                                     | 1.7%       | 5.7.1 (delivery not authorized); 554 5.7.1 <END-OF-MESSAGE>: End-of-data rejected:    |               |               |
| email account is over quota (<HEXDUMP>+<HEXDUMP>+<HEXDUMP>)                           |            |                                                                                       |               |               |
| 4                                                                                     | 1.7%       | 5.7.1 (delivery not authorized); 554 5.7.1 Abusive connections. Please try again      |               |               |
| later.                                                                                |            |                                                                                       |               |               |
| 4                                                                                     | 1.7%       | 5.7.1 (delivery not authorized); 554 5.7.1 Error: Mailbox over quota                  |               |               |
| 4                                                                                     | 1.7%       | 5.7.1 (delivery not authorized); 554 5.7.1 Service unavailable; Sender address        |               |               |
| [<ORIG>] found on high spam probability list                                          |            |                                                                                       |               |               |
| 4                                                                                     | 1.7%       | 5.7.1 (delivery not authorized); 554 5.7.1 Spam filter: Please Visit                  |               |               |
| http://www.ncmail.net/rejectedemails.htm to resolve issues                            |            |                                                                                       |               |               |
| 3                                                                                     | 1.3%       | 5.0.0 (undefined status); 550 5.0.0 Rejected-This domain is a chronic spammer.Contact |               |               |
| <OTHER_USER>@<hotmail.fr> via an alternate domain                                     |            |                                                                                       |               |               |
| 3                                                                                     | 1.3%       | 5.1.1 (bad destination mailbox address); 550 5.1.1 <<RCPT_USER>@<hotmail.fr>>:        |               |               |
| Recipient address rejected: User unknown in virtual alias table                       |            |                                                                                       |               |               |
| ...                                                                                   |            |                                                                                       |               |               |

The top bounce domain breakdown shows detailed bounce reasons for the top domains. By default, the top 10 domains (in terms of bounces) are printed, but this can be changed with the --topDomains num option. Data was removed from the above output to save space in this document.

For each domain the top bounce reasons are given as absolute number of bounces and as percentage of the bounces. The DSN (delivery status notification) code gives the reason for that bounce, as received by the foreign mail server. As a trade-off between detail and usability the DSN reason is only slightly modified (for example, email addresses in bounce messages are also replaced by ...@...to be able to generalize the bounces into bounce categories).

So, while the total bounces breakdown provides an aggregated overview of the bounces, the top domains breakdown gives full detail on bounces grouped by domain.

#### 11.4.2.6 Top Domains

##### Top Domains Breakdown:

Domains with largest number of queued recipients:

| domain      | recipients | bounced | avg. time | max. time | %rcpt. |
|-------------|------------|---------|-----------|-----------|--------|
| hotmail.com | 29,718     | 54      | 0:00:00   | 0:10:41   | 4.2%   |
| yahoo.com   | 18,326     | 3,326   | 0:00:17   | 0:21:45   | 2.6%   |
| gmail.com   | 14,181     | 0       | 0:00:00   | 0:00:03   | 2.0%   |
| mañana.com  | 12,021     | 908     | 0:00:06   | 0:20:00   | 1.7%   |
| 中文邮箱.网络     | 10,492     | 2,490   | 0:00:12   | 0:20:50   | 1.5%   |

Domains with highest number of bounces:

| domain     | recipients | bounced | avg. time | max. time | %rcpt. |
|------------|------------|---------|-----------|-----------|--------|
| yahoo.com  | 18,326     | 3,326   | 0:00:17   | 0:21:45   | 2.6%   |
| 中文邮箱.网络    | 10,492     | 2,490   | 0:00:12   | 0:20:50   | 1.5%   |
| noos.fr    | 1,648      | 1,648   | n/a       | n/a       | 0.2%   |
| hotmail.fr | 5,637      | 1,440   | 0:00:34   | 0:21:01   | 0.8%   |
| yahoo.fr   | 5,007      | 1,301   | 0:00:15   | 0:20:20   | 0.7%   |

Domains with highest average delivery times:

| domain         | recipients | bounced | avg. time | max. time | %rcpt. |
|----------------|------------|---------|-----------|-----------|--------|
| earthlink.net  | 1,940      | 1,119   | 0:31:43   | 1:00:59   | 0.3%   |
| mindspring.com | 634        | 380     | 0:28:28   | 1:00:50   | 0.1%   |
| freenet.de     | 507        | 0       | 0:10:50   | 0:45:01   | 0.1%   |
| 3363.com       | 3          | 1       | 0:10:00   | 0:20:00   | 0.0%   |
| 3471.com       | 1          | 0       | 0:10:00   | 0:10:00   | 0.0%   |

This report shows the counts and throughput for the top domains based on:

- the largest number of recipients initially submitted and queued for delivery,
- the highest number of bounces handled by PowerMTA, and
- the highest average delivery times.

The scope of the outputs are based on the type of report specified, i.e., global or by originator, envelope ID, or virtual mail transfer agent.

For all three sections, the `recipients` column shows the total number of recipients submitted and queued for delivery for the domains, while the `bounced` column provides just the number of recipients that PowerMTA itself bounced (local bounces). Note that this is not a count of all of the bounces since any bounces happening after PowerMTA has transferred the message to an intermediate mailer count as successful transfers. The difference between the `recipients` and `bounced` columns is then the number of recipients that were successfully transferred by PowerMTA.

For all three sections, the `avg. time` and `max. time` calculations are based on the difference between each message's queue timestamp and transfer timestamp. Only messages that were successfully delivered are included; bounces and other delivery reports are not. Messages transferred in multi-recipient (multi-"RCPT TO") SMTP transactions are broken down and each recipient is counted as a separate "delivery", each with its own transfer rate equal to that of the whole transaction itself. This is not an issue with single RCPT TO messages since messages delivered already equals recipients delivered.

The `avg. time` is calculated by adding together all of the transfer times (transfer timestamp - queue timestamp) to the respective domain for each recipient, and then dividing this by the number of recipients to the domain. Using AOL as an example in the first output, one would read that it took an average of 2 seconds for each message for each recipient to be delivered, once each message was queued, for the 92,891 recipients at AOL.com.

The `max. time` is the longest transfer time (transfer timestamp - queue timestamp) for one message to the respective domain. In the AOL case, one would read that the longest it took for one message to be delivered to AOL was 1 minute and 23 seconds. Conversely, all other messages to AOL for the 92,891 recipients were transferred in equal or less time.

The `%rcpt` column provides the percentage of recipients submitted and queued for the domain to the total number of recipients across the scope of the report. For example, AOL's 92,891 queued recipients were 23.2% of the 400,761 total queued recipients.

#### 11.4.2.7 Top Rates

##### Top Rates:

Top recipient delivery rates:

| measured over | rcpt./min. | recipients | beginning on        |
|---------------|------------|------------|---------------------|
| one minute    | 6,458      | 6,458      | 2018-11-06 21:05:17 |
| five minutes  | 5,495      | 27,479     | 2018-11-06 21:04:38 |
| one hour      | 5,290      | 317,450    | 2018-11-06 20:09:36 |

Top throughput rates:

| measured over | kbytes/s | kbytes   | recipients | beginning on        |
|---------------|----------|----------|------------|---------------------|
| one minute    | 21.4     | 1,286.1  | 6,458      | 2018-11-06 21:05:17 |
| five minutes  | 18.3     | 5,485.4  | 27,469     | 2018-11-06 21:01:40 |
| one hour      | 17.6     | 63,229.6 | 317,401    | 2018-11-06 20:12:19 |

This report shows the top throughput rates realized per recipient and the top data transfer rates during a time interval of one and five minutes and one hour. The scope of the output is *always* global and such is always based on the total numbers in the whole accounting file. All messages, normal deliveries as well as reports (including bounces) are taken into account when calculating these rates.

The top recipient delivery rates are calculated for each time increment by sorting all accounting records by delivery time and then searching for the time interval during which the largest number of recipients were handled. This number is then entered in then recipients column. The rcpt./min. column provides a uniform recipient transfer measurement across the three time intervals, and is calculated by taking the numbers in the recipients column and dividing each by the number of minutes that make up each respective time interval (i.e. dividing the five minutes result by five and the one hour result by sixty). The messages column shows the corresponding number of messages that were actually sent (i.e., number of SMTP transactions) while handling these recipients. This information is important when comparing performance numbers, since delivering a single message to multiple recipients is a lot less resource intensive than delivering individual messages.

The top throughput rates are calculated in the same manner as the recipient delivery rates, however the top amount of data transferred per time interval is reported rather than the number of recipients. This transfer rate output is independent to the recipient delivery rate output, so they may or may not reflect the same messages. The kbytes/s column provides a uniform data transfer measurement across the three time increments, and is calculated by taking the numbers in the kbytes column and dividing each by the number of seconds that make up each respective time increment (i.e., dividing the five minutes result by 300 and the one hour result by 3600). The messages column shows the corresponding number of messages that were actually sent comprising the number of kilobytes shown.

#### 11.4.2.8 VMTA Summary

VirtualMTA Traffic Summary:

| VirtualMTA | IP address  | Recipients | Delivered | Bounced | Bounced% |
|------------|-------------|------------|-----------|---------|----------|
| -----      | -----       | -----      | -----     | -----   | -----    |
| vmta816    | 127.0.0.1   | 920        | 802       | 118     | 12.8%    |
| vmta76     | 127.0.0.1   | 824        | 742       | 82      | 10.0%    |
| vmta447    | 127.0.0.1   | 794        | 716       | 78      | 9.8%     |
| vmta489    | 127.0.0.1   | 792        | 706       | 86      | 10.9%    |
| vmta545    | 127.0.0.1   | 772        | 691       | 81      | 10.5%    |
| vmta77     | 127.0.0.1   | 770        | 691       | 79      | 10.3%    |
| vmta816    | 10.59.14.69 | 763        | 707       | 56      | 7.3%     |
| vmta688    | 127.0.0.1   | 758        | 677       | 81      | 10.7%    |
| vmta941    | 127.0.0.1   | 748        | 665       | 83      | 11.1%    |
| vmta603    | 127.0.0.1   | 743        | 664       | 79      | 10.6%    |
| vmta76     | 10.59.14.69 | 737        | 670       | 67      | 9.1%     |
| vmta632    | 127.0.0.1   | 726        | 653       | 73      | 10.1%    |
| vmta602    | 127.0.0.1   | 722        | 647       | 75      | 10.4%    |
| vmta567    | 127.0.0.1   | 719        | 655       | 64      | 8.9%     |
| vmta523    | 127.0.0.1   | 713        | 623       | 90      | 12.6%    |
| vmta960    | 127.0.0.1   | 711        | 656       | 55      | 7.7%     |
| vmta718    | 127.0.0.1   | 710        | 636       | 74      | 10.4%    |
| vmta134    | 127.0.0.1   | 708        | 628       | 80      | 11.3%    |
| vmta13     | 127.0.0.1   | 703        | 618       | 85      | 12.1%    |
| vmta5      | 127.0.0.1   | 701        | 628       | 73      | 10.4%    |
| vmta387    | 127.0.0.1   | 698        | 621       | 77      | 11.0%    |
| vmta723    | 127.0.0.1   | 698        | 641       | 57      | 8.2%     |
| vmta825    | 127.0.0.1   | 693        | 629       | 64      | 9.2%     |
| vmta62     | 127.0.0.1   | 690        | 624       | 66      | 9.6%     |
| vmta941    | 10.59.14.69 | 688        | 630       | 58      | 8.4%     |
| vmta794    | 127.0.0.1   | 688        | 621       | 67      | 9.7%     |
| vmta945    | 127.0.0.1   | 687        | 630       | 57      | 8.3%     |
| vmta124    | 127.0.0.1   | 681        | 599       | 82      | 12.0%    |
| vmta136    | 127.0.0.1   | 679        | 598       | 81      | 11.9%    |
| vmta695    | 127.0.0.1   | 678        | 597       | 81      | 11.9%    |
| vmta51     | 127.0.0.1   | 676        | 615       | 61      | 9.0%     |
| vmta286    | 127.0.0.1   | 672        | 587       | 85      | 12.6%    |
| vmta545    | 10.59.14.69 | 670        | 636       | 34      | 5.1%     |
| vmta400    | 127.0.0.1   | 665        | 600       | 65      | 9.8%     |
| vmta346    | 127.0.0.1   | 662        | 592       | 70      | 10.6%    |
| vmta357    | 127.0.0.1   | 661        | 585       | 76      | 11.5%    |
| vmta447    | 10.59.14.69 | 661        | 614       | 47      | 7.1%     |
| vmta456    | 127.0.0.1   | 660        | 591       | 69      | 10.5%    |
| vmta718    | 10.59.14.69 | 659        | 611       | 48      | 7.3%     |
| ...        |             |            |           |         |          |
| Total      |             | 701,338    | 635,365   | 65,973  | 9.4%     |

VMTA Summary shows high level counters on the amount of messages that were sent to individual VirtualMTAs. This information includes a breakdown of deliveries, bounces, and percentage of bounces per VirtualMTA. The delivering IP is also included to help in determining if there is a reputation/sending issue with a given address.

#### 11.4.2.9 VMTA Time Breakdown

VirtualMTA Traffic Breakdown By Time:

VirtualMTA: vmta816, IP Address: 127.0.0.1

Daily Breakdown:

| Date       | Recipients | Delivered | Bounced | Bounced% |
|------------|------------|-----------|---------|----------|
| -----      | -----      | -----     | -----   | -----    |
| 2018-11-06 | 920        | 802       | 118     | 12.8%    |

Hourly Breakdown:

| ----- 2018-11-06 ----- |            |           |         |          |
|------------------------|------------|-----------|---------|----------|
| Interval               | Recipients | Delivered | Bounced | Bounced% |
| -----                  | -----      | -----     | -----   | -----    |
| 00:00 - 00:59          | 0          | 0         | 0       | 0.0%     |
| 01:00 - 01:59          | 0          | 0         | 0       | 0.0%     |
| 02:00 - 02:59          | 0          | 0         | 0       | 0.0%     |
| 03:00 - 03:59          | 0          | 0         | 0       | 0.0%     |
| 04:00 - 04:59          | 0          | 0         | 0       | 0.0%     |
| 05:00 - 05:59          | 0          | 0         | 0       | 0.0%     |
| 06:00 - 06:59          | 0          | 0         | 0       | 0.0%     |
| 07:00 - 07:59          | 0          | 0         | 0       | 0.0%     |
| 08:00 - 08:59          | 0          | 0         | 0       | 0.0%     |
| 09:00 - 09:59          | 0          | 0         | 0       | 0.0%     |
| 10:00 - 10:59          | 0          | 0         | 0       | 0.0%     |
| 11:00 - 11:59          | 0          | 0         | 0       | 0.0%     |
| 12:00 - 12:59          | 0          | 0         | 0       | 0.0%     |
| 13:00 - 13:59          | 0          | 0         | 0       | 0.0%     |

|               |     |     |    |       |
|---------------|-----|-----|----|-------|
| 14:00 - 14:59 | 0   | 0   | 0  | 0.0%  |
| 15:00 - 15:59 | 0   | 0   | 0  | 0.0%  |
| 16:00 - 16:59 | 0   | 0   | 0  | 0.0%  |
| 17:00 - 17:59 | 0   | 0   | 0  | 0.0%  |
| 18:00 - 18:59 | 0   | 0   | 0  | 0.0%  |
| 19:00 - 19:59 | 424 | 368 | 56 | 13.2% |
| 20:00 - 20:59 | 390 | 344 | 46 | 11.8% |
| 21:00 - 21:59 | 106 | 90  | 16 | 15.1% |
| 22:00 - 22:59 | 0   | 0   | 0  | 0.0%  |
| 23:00 - 23:59 | 0   | 0   | 0  | 0.0%  |
| ...           |     |     |    |       |

The VMTA time breakdown shows bounce counts for the top VirtualMTAs by time interval. By default all the VirtualMTAs are printed. For each VirtualMTA the bounce counts are given as absolute number of bounces and as percentage of the bounces for a given time period.

#### 11.4.2.10 VMTA Top Bounces

| VirtualMTA Top Bounce Domains:             |            |           |         |          |
|--------------------------------------------|------------|-----------|---------|----------|
| VirtualMTA: vmta816, IP Address: 127.0.0.1 |            |           |         |          |
| Domain                                     | Recipients | Delivered | Bounced | Bounced% |
| blåbærsyltetøy.gulbrandsen.priv.no         | 22         | 15        | 7       | 31.8%    |
| telefonica.net                             | 14         | 8         | 6       | 42.9%    |
| öbb.de                                     | 19         | 14        | 5       | 26.3%    |
| centurytel.net                             | 6          | 1         | 5       | 83.3%    |
| 中文邮箱.网络                                    | 19         | 15        | 4       | 21.1%    |
| windowslive.com                            | 11         | 7         | 4       | 36.4%    |
| live.nl                                    | 9          | 5         | 4       | 44.4%    |
| knology.net                                | 5          | 1         | 4       | 80.0%    |
| orange.fr                                  | 17         | 14        | 3       | 17.6%    |
| voila.fr                                   | 10         | 7         | 3       | 30.0%    |
| live.com.au                                | 9          | 6         | 3       | 33.3%    |
| att.net                                    | 8          | 5         | 3       | 37.5%    |
| gmx.de                                     | 5          | 2         | 3       | 60.0%    |
| gmx.fr                                     | 5          | 2         | 3       | 60.0%    |
| surewest.net                               | 4          | 1         | 3       | 75.0%    |
| 光明网.cn                                     | 19         | 17        | 2       | 10.5%    |
| free.fr                                    | 12         | 10        | 2       | 16.7%    |
| wanadoo.fr                                 | 11         | 9         | 2       | 18.2%    |
| rocketmail.com                             | 9          | 7         | 2       | 22.2%    |
| wowway.com                                 | 6          | 4         | 2       | 33.3%    |
| ...                                        |            |           |         |          |

The VMTA Top Bounces report shows bounce counts for all the VirtualMTAs. By default, the top 20 domains (in terms of bounces) are printed, but this can be changed with the `--topBounced <num>` option.

For each domain the top bounce reasons are given as absolute number of bounces and as percentage of the bounces. So, while the top bounces report provides an aggregated overview of the bounces, the VMTA top bounces report gives high level data on bounces grouped by VirtualMTA.

## 11.5 The acctfind Accounting Reporting Application

acctfind is a powerful accounting file search and reporting tool. It can be found in the `\pmta\bin` directory on Windows, and in the `/usr/sbin` directory (named `pmtaacctfind`) on Unix.

The tool is not required for exporting data from the accounting file because the file is no longer in a proprietary binary format. With the CSV accounting file written by PowerMTA, reporting can be done without this tool. This tool, however, will still work with the CSV file format, and is provided to assist with extracting subsets of the data.

acctfind allows one to search for records in the accounting file, as well as to produce a custom output based on specific fields in the matching records. For example, one can easily use acctfind to generate a list of all addresses that bounced with a persistent (5xx) error for a particular campaign, sorted by the bounce reason, in a tab separated output that contains the recipient address first, followed by the MAIL FROM/originator address, then the bounce code and reason.

acctfind currently supports seven different output formats including standard, comma separated values, tab separated values, html, xml, "pretty" xml, and custom-defined outputs.

This Section provides a quick overview of the feature set and options, along with sample commands, however the more detailed help obtained by passing the `--help` option to acctfind itself should be referenced for further insight.

The general format for the command is

```
acctfind [options] accounting files ...
```

The top level segmentation options are based on the record types, which include searching for matches in records for messages successfully delivered (`--delivered`), bounce reports delivered (`--bounced`), run status records (`--status`), and general mta information (`--mta`).

For example,

```
acctfind --bounced acct.csv
```

would search for matching records only among the bounce report records. If none of these are specified in the command, the default is to search for matches in both message delivered records and bounce report records.

All of the fields in the accounting records (listed in "[Successful delivered](#)" Records & "[Transient Queue](#)" Records) are available for matching (with the `--match` option), and the match pattern can be either a perl regular expression (e.g., `m/^5\./`) or some string for case-independent string matching.

If you want to restrict the matching to some fields, you can do that by providing a list of fields to match enclosed in square brackets in front of the pattern (e.g. `--match [from,to]port25.com`).

Shortcut options for matching single fields are also available. For example, `--match [to]port25` is equivalent to `--to port25`. To further refine the output, more than one matching needle can be specified. For example, to get a listing of all yahoo.com addresses that bounced with a persistent (5xx) error, you would have the following option combinations in the command

```
acctfind --bounced --dsnStatus m/^5\./ --to yahoo.com acct.csv
```

As mentioned, seven different output formats are supported. You choose the desired format using the `--output` option, along with the format and the fields desired to be outputted. For example,

```
acctfind --delivered --to hotmail.com --output csv[to,from,envId,vmta,dstIp] acct.csv
```

would search all message delivered records for recipient addresses at hotmail.com and generate a comma separated output which lists, on each line,

- the hotmail recipient address,
- the message originator address,
- the envelope ID used for tracking the message,
- the VirtualMTA used to deliver the message, and
- the IP address of the Hotmail machine that accepted the message.

To get an HTML output vs. csv output of the same, you would run

```
acctfind --delivered --to hotmail.com --output html[to,from,envId,vmta,dstIp] acct.csv
```

A full detailed listing and description of all of the options available can be found in the help by running `acctfind --help`. Included below are a few sample commands and options to help better understand the formats and capabilities. Do not hesitate to contact us at [support@port25.com](mailto:support@port25.com) with any questions or when seeking the proper command sequence for a desired result.

### 11.5.1 Examples

1. To generate a list of all addresses that bounced in a comma separated output, that contains the recipient address first, followed by the MAIL FROM/originator address, then VirtualMTA used, then the bounce codes and reason, sorted by the bounce reason (DSN Diagnostics):

```
acctfind --bounced --sortBy dsnDiag --output tsv[to,from,vmta,dsnStatus,dsnDiag] acct.csv
```

Note that the `--sortBy` option is only supported on Unix at this time.

2. To generate a list of all addresses that bounced with a persistent (5xx) error, in a comma separated output that contains the recipient address first, followed by the MAIL FROM/originator address, envelope ID, then the bounce code and reason:

```
acctfind --bounced --dsnStatus m/^5\./ --output csv[to,from,envId,dsnStatus,dsnDiag] acct
```

3. To generate a list of all hotmail.com addresses that bounced, in an HTML output that contains the recipient address, and bounce reasons:

```
acctfind --bounced --to hotmail --output html[to,dsnStatus,dsnDiag] acct
```

4. To generate a list of all hotmail.com addresses that were successfully transferred, in a "pretty XML" output that contains the recipient addresses, VirtualMTA used to deliver the messages, the queue and transfer time stamps in ISO format, and the IP address of the hotmail machine that accepted the messages:

```
acctfind --iso-times --delivered --to hotmail --output prettyxml[to,vmta,timeQueued,timeDelivered,dstIp] acct.csv
```

5. To search for a particular email address (`test@port25.com`) to see whether it was delivered or bounced, with a tsv output showing the address, whether the message failed, and the ISO-formatted queue time and delivery time:

```
acctfind --iso-times --to test@port25.com --output tsv[to,dsnAction,timeQueued,timeDelivered] acct.csv
```

6. To search for a particular email address (`test@port25.com`) to see whether it was delivered or bounced, with the standard output:

```
acctfind --iso-times --to support acct.csv
```

7. To search for a particular header that is being logged via the `acct-log-header` directive:

```
acctfind --output csv[to,from,header:x-foo] acct.csv
```

## 11.6 Bounce Categories

For easier classification of the actual reason for a bounce, `pmtastats` groups bounces in *bounce categories*. All bounces in a category have probably the same root cause, e.g. all messages in a category may be considered spam or another category may indicate the recipient is unknown. Using the various bounce codes and DSN reasons, `pmtastats` aggregates these message in likely bounce categories. This gives a good overview of why messages could not be delivered.

The categories are customizable using the `<bounce-category-patterns>` tag (see [Use of Bounce Category Patterns](#)). Any new or custom categories would be show in the `pmtastats` output as well as in the accounting file itself.

The bounce category is included in the accounting file (the `bounceCat` tag, see ["Successful delivered" Records](#)). It can be obtained directly from the CSV file or be used in `acctfind`.

Currently, these bounce categories are defined:

| category                   | description                                                                                           |
|----------------------------|-------------------------------------------------------------------------------------------------------|
| <b>bad-configuration</b>   | messages rejected due to configuration issues with remote host, 5.X.X error                           |
| <b>bad-connection</b>      | messages bounced due to bad connection issues with remote host, 4.X.X error                           |
| <b>bad-domain</b>          | messages bounced due to invalid or non-existing domains, 5.X.X error                                  |
| <b>bad-mailbox</b>         | messages rejected due to bad, invalid, or non-existent recipient addresses, 5.X.X error               |
| <b>content-related</b>     | messages refused or blocked due to content related reasons, 5.X.X error                               |
| <b>inactive-mailbox</b>    | messages rejected due to expired, inactive, or disabled recipient addresses, 5.X.X error              |
| <b>invalid-sender</b>      | messages bounced due to invalid DNS or MX entry for sending domain                                    |
| <b>message-expired</b>     | messages bounced due to not being delivered before the bounce-after, 4.X.X error                      |
| <b>no-answer-from-host</b> | messages bounces due to receiving no response from remote host after connecting, 4.X.X or 5.X.X error |
| <b>other</b>               | messages rejected due to other reasons, 4.X.X or 5.X.X error                                          |
| <b>policy-related</b>      | messages refused or blocked due to general policy reasons, 5.X.X error                                |
| <b>protocol-errors</b>     | messages rejected due to SMTP protocol syntax or sequence errors, 5.X.X error                         |
| <b>quota-issues</b>        | messages rejected or blocked due to mailbox quota issues, 4.X.X or 5.X.X error                        |
| <b>relaying-issues</b>     | messages refused or blocked due to remote mail server relaying issues, 5.X.X error                    |
| <b>routing-errors</b>      | messages bounced due to mail routing issues for recipient domain, 5.X.X error                         |
| <b>spam-related</b>        | messages refused or blocked due to spam related reasons, 5.X.X error                                  |
| <b>virus-related</b>       | messages refused or blocked due to virus related reasons, 5.X.X error                                 |

Note that this list and the classifications may be subject to change, since DSN status messages may vary.

## 12 Processing Inbound email

### 12.1 Introduction

There may be times when PowerMTA needs to handle inbound mail outside of messages created by a submission (e.g. list manager, CRM, etc.) application. These messages may be remote (asynchronous) bounces, corporate mail, or any other type of email originating from outside an organization's infrastructure. In these cases, PowerMTA can accept the messages and do one of a few things with the messages. It can:

- Relay them to another mail server
- Deliver them to file on the local system
- Discard them (blackhole)
- Deliver to pipe (local mail processing application)
- Process as an asynchronous (remote) bounce
- Process as a feedback loop email

Any combination of these options may be used, and PowerMTA will create a subsequent delivery record in the accounting file for mail that is processed in one of these manners.

### 12.2 Relaying

To relay messages to another mail server, PowerMTA will need to be configured to accept the messages, and then deliver them to their final destination.

```
relay-domain yourdomain.com
relay-domain otherdomain.com
relay-address bounce@bounces.yourdomain.com

<domain bounces.yourdomain.com>
  smtp-hosts [1.2.3.4]:25
</domain>

<domain otherdomain.com>
  smtp-hosts [5.6.7.8]:2525
</domain>
```

In the above example, messages for bounces.yourdomain.com are accepted for delivery and then forwarded to server 1.2.3.4 on port 25 and messages for otherdomain.com are forwarded to 5.6.7.8 on port 2525 and marked as such in the accounting file.

### 12.3 Deliver to File

To deliver messages to file, PowerMTA will need to be configured to accept the messages, and then deliver them to somewhere on the local disk system.

```
relay-domain bounces.yourdomain.com
relay-domain otherdomain.com
relay-address bounce@bounces.yourdomain.com

<domain bounces.yourdomain.com>
  type file
  file-format append-mbox
  file-destination /etc/pmta/inbound/$domain
</domain>

<domain otherdomain.com>
  type file
  file-format newfile-pickup
  file-destination /etc/pmta/inbound/$domain
</domain>
```

In the above example, messages for bounces.yourdomain.com and otherdomain.com are accepted for delivery and then each message is delivered to the individual files (based on domain) on the local disk and marked as such in the accounting file.

### 12.4 Discarding

To discard (blackhole) the inbound messages, PowerMTA will need to be configured to accept the messages, and then discard them based on domain.

```
relay-domain bounces.yourdomain.com
relay-domain otherdomain.com
relay-address bounce@bounces.yourdomain.com

<domain bounces.yourdomain.com>
  type discard
</domain>

<domain otherdomain.com>
  type discard
</domain>
```

In the above example, messages for bounces.yourdomain.com and otherdomain.com are accepted for delivery, then discarded and marked as such in the accounting file.

### 12.5 Delivering to local application (pipe)

To deliver messages to a local application via pipe, PowerMTA will need to be configured to accept the messages, and then deliver them based on domain.

```
relay-domain bounces.yourdomain.com
relay-domain otherdomain.com
relay-address bounce@bounces.yourdomain.com

<domain bounces.yourdomain.com>
  type pipe
  command "/my/bounce/processor --envid \"$envid\" \"$user\""
```

```

</domain>

<domain otherdomain.com>
    type pipe
    command "/my/bounce/processor --envid \"${envid}\" \"${user}\""
</domain>

```

In the above example, messages for bounces.yourdomain.com and otherdomain.com are accepted for delivery, then piped to the custom application (with associated parameters) on the local file system and marked as such in the accounting file.

## 12.6 Processing Asynchronous (Remote) Bounces and remote status reports

Remote bounce processing works as a "filter" located after receipt of an email and before it is routed to a delivery queue. If PowerMTA recognizes the format and no errors occur while processing it, it writes a new type of accounting record ("rb" and "rs", see below). For any "filtered" emails you have the option of discarding the email at that point or letting it pass to regular delivery (e.g. see [deliver-matched-email](#)). Depending on these settings, or if an error occurs, it is passed to a delivery queue and delivered normally (by whatever method assigned to the domain -- usually 'file' or 'pipe').

The email address used should be unique for each email sent to allow for more detailed tracking. For example, instead of [<bounce@bounce.yourdomain.com>](mailto:bounce@bounce.yourdomain.com) or an address starting with postmaster@ or abuse@, an address like the following should be considered:

[customer1-campaign563-uniqueID498294@bounce.yourdomain.com](mailto:customer1-campaign563-uniqueID498294@bounce.yourdomain.com)

### Accounting Records

There are two kinds of records for asynchronous bounces: "rb" for remote bounce and "rs" for reporting other kinds of remote status (used for DSN positive and "delayed" status notifications). Most people will only be interested in "rb" records. The following columns are available for these records:

| Name            | Description                                        |
|-----------------|----------------------------------------------------|
| timeLogged      | current time                                       |
| orig            | originator (taken from report's recipient address) |
| rcpt            | (final) recipient                                  |
| orcpt           | original recipient                                 |
| dsnAction       | DSN action                                         |
| dsnStatus       | DSN status                                         |
| dsnDiag         | DSN diagnostic                                     |
| dsnMta          | DSN remote MTA                                     |
| dsnReportingMta | DSN reporting MTA                                  |
| bounceCat       | bounce category                                    |
| srcType         | type of source from which the report was received  |
| srcMta          | MTA from which the report was received             |
| envId           | envelope ID                                        |
| header_<name>   | header extracted from original email               |

### Configuration

You define a list of email addresses that are bounce addresses, i.e., original MAIL FROM addresses. Only email received for those addresses gets filtered/looked at looking for bounces. Use of the following requires a restart of the PowerMTA service. You can define entire domains which get filtered, or for more granular control you can use regular expressions to match entire addresses. There are also options to deliver filtered emails that are recognized (matched) or not recognized:

```

<bounce-processor>
    deliver-unmatched-email yes          # default: yes
    deliver-matched-email yes           # default: no
    forward-unmatched-to auto-feedback@port25.com
    forward-errors-to auto-feedback@port25.com
    <address-list>
        domain domain.to.filter        # whole domain
        address /regex@domain.to.filter/ # regex
    </address-list>
</bounce-processor>

```

The accounting records are configured just like the other types in the configuration. The example below has both local (sync) and remote (async) bounces written to the same file, and includes a custom "x-id" header:

```

<acct-file /var/log/pmta/bounces.csv>
    records b, rb
    record-fields b *,header_x-id
    record-fields rb *,header_x-id
</acct-file>

<acct-file /var/log/pmta/status.csv>
    records rs
    record-fields rs *,header_x-id
</acct-file>

```

If you are having trouble, setting the following directive will log all filtered emails:

```

<domain {bounce}>
    log-messages yes
</domain>

```

## 12.7 Processing Feedback Loop Emails

To enable feedback loop processing, you add something like:

```

<feedback-loop-processor>
    deliver-unmatched-email no
    deliver-matched-email yes          # default: no
    forward-unmatched-to auto-feedback@port25.com
    forward-errors-to auto-feedback@port25.com
    <address-list>
        address /fbl@fbl.yourdomain.com/
    </address-list>
</feedback-loop-processor>

```

to the configuration file. The "forward-..." directives can be specified more than once to have email forward to multiple addresses. "<address-list>" accepts individual domains like domain foo.com and regular expressions like in the example above, so you have great flexibility in specifying the addresses for which we should attempt FBL processing. As with bounce processing, you have the option to continue to deliver the emails after processing (e.g., to save them all to a folder) or to have them end there. Failed emails always proceed to delivery.

The email address used should be something easy to parse. For example:

[fbl@fbl.yourdomain.com](mailto:fbl@fbl.yourdomain.com)

PowerMTA should be the MX to handle these messages. If there are MTAs that handle the messages prior to being delivered to PowerMTA, PowerMTA may be unable to recognize the message as a FBL email.

Use of postmaster@ or abuse@ is discouraged.

To allow for more detailed tracking a custom header may be added to the email (assuming the remote mailer doesn't redact the header).

```
x-tracker: customer1-campaign563-uniqueID498294
```

Custom headers are not included by default in the PowerMTA accounting file, so if the above header is needed in the accounting make sure to add it as a custom header.

Emails matching the address list go into a special {feedbackLoop} queue, where they are processed. You can configure an accounting file to receive the resulting data:

```
<acct-file /var/log/pmta/fbl.csv>
  records feedback-loop
  record-fields f *,header_subject
</acct-file>
```

"f" and "feedback-loop" can be used for the record name interchangeably. In the accounting file you actually see 'f'. The available columns (all of which, except for 'header', are included by default) are:

The following columns are available for f records:

| Name           | Description                                              |
|----------------|----------------------------------------------------------|
| timeLogged     | current time                                             |
| repSourceIp    | the IP of PowerMTA that received the message             |
| format         | arf or jmrp                                              |
| userAgent      | name & version of the program that generated the report  |
| envId          | envelope ID                                              |
| orig           | originator (taken from report's recipient address)       |
| rcpt           | (final) recipient                                        |
| reportingMta   | the name of the MTA generating this feedback report      |
| divSourceIp    | address of MTA from which the message was received       |
| header <name>  | header extracted from original email                     |
| arf <name>     | field from ARF feedback-report                           |
| reportedDomain | the domain about which the feedback report was generated |
| feedbackType   | mapped to ARF's Feedback-Type field                      |

'header\_From', 'header\_Return-Path', 'header\_X-job' and 'header\_Subject' are included by default.

If a feedback report includes multiple recipients, a separate 'f' record is written for each recipient, repeating any common fields. Since a field requested via 'arf\_...' may occur multiple times in the report, the column may contain a (CSV-encoded) list of values.

### 12.7.1 Hotmail Feedback Loop Emails

Hotmail does not follow the standard ARF format as the others. To get FBL emails to work for Hotmail messages, sign up for the RFC attachments option, since their "original" option does not include the key headers that PowerMTA parses on to indicate that it is an FBL, and the original format includes less info in general.

Because the RFC attachment option is still non-standard, the header fields from the original message will need to be extracted in contrast to standard ARF messages. Otherwise the RCPT field is empty by default in Hotmail JMRP accounting records.

In this case, PowerMTA will need to add the header "X-HmXmrOriginalRecipient" into the csv accounting file, which is where Hotmail includes the original recipient address that complained in the original message. To add custom headers to an acct record, add the header within the syntax "header\_xxx" in the record-fields directive, for example:

```
<acct-file /var/log/pmta/fbl.csv>
  records feedback-loop
  record-fields f *,header_X-HmXmrOriginalRecipient
</acct-file>
```

Alternatively, the map-header-to-field feature can be used to cause the custom header "X-HmXmrOriginalRecipient" to be mapped to a standard field. With this, the same field for all FBL emails will be used. To map this Hotmail specific header to the existing rcpt field use something like the following:

```
<acct-file /var/log/pmta/fbl.csv>
  records feedback-loop
  record-fields f *
  map-header-to-field fbl header_X-HmXmrOriginalRecipient rcpt
</acct-file>
```

Lastly, what many sites do instead is create a custom header in each message that obfuscates the recipient address, perhaps using a mailing or campaign ID as well, for example:

```
x-rid: 12345-hdhgjkeyudfpdfldfd
```

and then have this header included in the FBL accounting file. When parsing the csv file just key on this header, since again, it is included in all FBLs sent back, which is the only thing consistent across the FBL generating sites.

## 12.8 Tracking a campaign in PowerMTA with a JobID

To track campaigns in PowerMTA it is best to use the x-job header. This will show as a JobID or a Job in PowerMTA and PowerMTA Management Console. For this to work properly the application feeding PowerMTA needs to set an x-job header. The value of this header should be unique to each campaign. For example:

```
x-job: UniqueID-Date
```

With this set, PowerMTA needs to know to look for the header with a setting similar to the following:

```
<source 0/0>
  process-x-job yes
</source>
```

If needed, an alternate header may be used with something similar to the following:

```
<source 0/0>
  jobid-header x-custom-header
  process-x-job yes
</source>
```

In the above example x-custom-header will be used as the JobID in place of x-job.

Once the above is set, PowerMTA will log the JobID to the accounting file once a delivery or bounce has taken place. Information about messages still in the queue can be found on the [Jobs tab](#) of the PowerMTA web monitor or by using the [pmta show jobs](#) command.

When using a Job ID, it may become necessary during the course of a mailing to pause the job. Instances of when this would be needed might be to make sure the injected job was setup correctly, or that it is using the correct VirtualMTAs. The command to pause job Campaign1234 would be:

```
pmta pause job Campaign1234
```

And to resume the job:

```
pmta resume job Campaign1234
```

Lastly, additional information about current queue and historical reporting can be found in the PowerMTA Management Console dashboard, monitoring, and reporting sections.

## 12.9 Aliases / Forwarding

There may be a need to use aliases or to forward emails. This directive allows a list of email addresses to be defined for which PowerMTA should forward messages to different addresses. For example, if a message is received by PowerMTA for `example@domain1abc.com`, then forward the message on to `support@port25.com`.

Alias expansion happens after a pattern list is applied and a VirtualMTA is selected, but before the message is added to the VirtualMTA. The same applies to messages injected using default-virtual-mta or an x-virtual-mta header.

Defining just the local part of the email address (the text to the left of the `@`) is also supported, however forwarding will only occur on addresses that have the matching local part and the domain of one of the local host names. Domain matching or regex pattern matching is not supported at this time.

The directives are applied on a global basis. The format is an email address (or local part of an address) followed by one or more than one comma separated email addresses. For example:

```
host-name yourdomain.com
host-name yourotherdomain.com
relay-domain example.com
relay-address user@example.com
<aliases>
  alias example@yourdomain.com support@port25.com,info@port25.com
  alias jdoe someone@example.com
</aliases>
```

In the above example emails sent to `example@yourdomain.com` will be forwarded to `support@port25.com` and `info@port25.com`. Emails sent to `jdoe@yourdomain.com` or `jdoe@yourotherdomain.com` will be forwarded to `someone@example.com`.

Multiple addresses can be defined with the same exact local part across different domains. For example:

```
<aliases>
  alias newsletters@domain1abc.com support@port25.com
  alias newsletters@domain2def.com info@port25.com
  alias newsletters@domain3xyz.com support@port25.com,info@port25.com
</aliases>
```

To forward all messages from a domain to a given address the following may be used:

```
<aliases>
  alias *@domain1abc.com support@port25.com
</aliases>
```

## 12.10 Handling non-standard DSN/FBL reports

While the major ISPs return FBL/DSN reports in standard format that PMTA can parse, some ISPs return these reports in non-standard formats.

Here's an example of such an email:

```
From: "ABC" <xyz@reports.spamcop.net>
To: abuse@foobar.com
Subject: [SpamCop] Is your data safe?
Precedence: list
Message-ID: <asdfkajsdfj>
Date: Fri, 31 Mar 2017 06:11:38 -0400
X-Mailer: https://www.spamcop.net/ v4.8.6

[ SpamCop V4.8.6 ]
This message is brief for your comfort. Please use links below for details.

Email from 1.2.3.4 / Fri, 31 Mar 2017 06:11:38 -0400
https://www.spamcop.net/w3m?i=askdjfkasjsadjfajs

[ Offending message ]
<the original email follows>
```

The above feedback report is not in the standard ARF format. To handle such reports, PMTA allows you to define the patterns that can match the body of such emails and extract fields from them that can be written to accounting files.

For the email above, you would create a regexp from the relevant part of the body of the email in a file, say `/etc/pmta/report-pattern.txt` such as:

```
Email from (?<dlvSourceIp>[0-9\\\.]+)
```

The above pattern file is configured for use in PMTA using directives like the following:

```
<feedback-loop-processor>
  <email-patterns>
    body-pattern-file /etc/pmta/report-pattern.txt
  </email-patterns>
</feedback-loop-processor>

<bounce-processor>
  <email-patterns>
    body-pattern-file /etc/pmta/report-pattern.txt
    log-match-info true
  </email-patterns>
</bounce-processor>
```

Whenever PMTA receives such reports, it will process the email through this pattern, and extract the source IP address(1.2.3.4 in the example above) into a field named `dlvSourceIp` - the source IP that was originally used to deliver the message. The field is then written in an 'f' record to the accounting file.

For feedback loop reports, you can extract the following fields from the body of the message:

- `reportedDomainTo` - domain about which the feedback report was generated
- `reportedDomainFrom` - domain that generated the feedback report
- `dlvSourceIp` - IP address from which the connection was made/attempted
- `rcpt` - recipient
- `orig` - originator

For DSN reports, you can extract the following fields from the body of the message:

- `r` - recipient
- `or` - originator
- `orig` - Original Recipient
- `diag` - DSN diagnostic string for failure
- `action` - DSN action
- `repMta` - DSN reporting MTA
- `remMta` - DSN remote MTA

Please contact [support@port25.com](mailto:support@port25.com) if you have questions about the use of this feature, or for help with handling such reports.

## 13 Acknowledgments

Regular expression support is provided by the [PCRE library package](http://ftp.csx.cam.ac.uk/pub/software/programming/pcre/), which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. PCRE is available from [ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/](http://ftp.csx.cam.ac.uk/pub/software/programming/pcre/).

PowerMTA uses, and may in its distribution kits include, [OpenSSL](http://www.openssl.org/). The OpenSSL license requires that we reproduce it here:

```
/* =====
 * Copyright (c) 1998-2006 The OpenSSL Project. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
 * distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 * software must display the following acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 * endorse or promote products derived from this software without
 * prior written permission. For written permission, please contact
 * openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 * nor may "OpenSSL" appear in their names without prior written
 * permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 * acknowledgment:
 * "This product includes software developed by the OpenSSL Project
 * for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * =====
 *
 * This product includes cryptographic software written by Eric Young
```

```

* (eay@cryptsoft.com). This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * "This product includes cryptographic software written by
 * Eric Young (eay@cryptsoft.com)"
 * The word 'cryptographic' can be left out if the rouines from the library
 * being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 * the apps directory (application code) you must include an acknowledgement:
 * "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed. i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
*/

```

PowerMTA uses, and may in its distribution kits include, [ICU](#). The ICU license requires that we reproduce it here:

```

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of
the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY
SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF
CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder
shall not be used in advertising or otherwise to promote the sale, use
or other dealings in this Software without prior written authorization
of the copyright holder.

All trademarks and registered trademarks mentioned herein are the

```

property of their respective owners.

-----

### Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

#### 1. Unicode Data Files and Software

##### COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2016 Unicode, Inc. All rights reserved.  
Distributed under the Terms of Use in  
<http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that

- (a) this copyright and permission notice appear with all copies of the Data Files or Software,
- (b) this copyright and permission notice appear in associated documentation, and
- (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.  
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

#### 2. Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

```
# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
# - https://sourceforge.net/project/?group_id=1519
# - Its license terms and conditions are shown below.
#
# * IPADIC (Japanese)
# - http://chasen.aist-nara.ac.jp/chasen/distribution.html
# - Its license terms and conditions are shown below.
```

```

#
# -----COPYING.libtabe ---- BEGIN-----
#
# /*
#  * Copyrighy (c) 1999 TaBE Project.
#  * Copyright (c) 1999 Pai-Hsiang Hsiao.
#  * All rights reserved.
#  *
#  * Redistribution and use in source and binary forms, with or without
#  * modification, are permitted provided that the following conditions
#  * are met:
#  *
#  * . Redistributions of source code must retain the above copyright
#  * notice, this list of conditions and the following disclaimer.
#  * . Redistributions in binary form must reproduce the above copyright
#  * notice, this list of conditions and the following disclaimer in
#  * the documentation and/or other materials provided with the
#  * distribution.
#  * . Neither the name of the TaBE Project nor the names of its
#  * contributors may be used to endorse or promote products derived
#  * from this software without specific prior written permission.
#  *
#  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#  * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#  * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#  * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#  * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#  * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#  * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#  * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#  * OF THE POSSIBILITY OF SUCH DAMAGE.
#  */
#
# /*
#  * Copyright (c) 1999 Computer Systems and Communication Lab,
#  * Institute of Information Science, Academia
#  * Sinica. All rights reserved.
#  *
#  * Redistribution and use in source and binary forms, with or without
#  * modification, are permitted provided that the following conditions
#  * are met:
#  *
#  * . Redistributions of source code must retain the above copyright
#  * notice, this list of conditions and the following disclaimer.
#  * . Redistributions in binary form must reproduce the above copyright
#  * notice, this list of conditions and the following disclaimer in
#  * the documentation and/or other materials provided with the
#  * distribution.
#  * . Neither the name of the Computer Systems and Communication Lab
#  * nor the names of its contributors may be used to endorse or
#  * promote products derived from this software without specific
#  * prior written permission.
#  *
#  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#  * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#  * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#  * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#  * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#  * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#  * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#  * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#  * OF THE POSSIBILITY OF SUCH DAMAGE.
#  */
#
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsai4@uiuc.edu http://casper.beckman.uiuc.edu/~c-tsai4
#
# -----COPYING.libtabe-----END-----
#
# -----COPYING.ipadic-----BEGIN-----
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED

```

```

# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# -----COPYING.ipadic-----END-----

```

### 3. Lao Word Break Dictionary Data (laodict.txt)

```

# Copyright (c) 2013 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: http://code.google.com/p/lao-dictionary/
# Dictionary: http://lao-dictionary.googlecode.com/git/Lao-Dictionary.txt
# License: http://lao-dictionary.googlecode.com/git/Lao-Dictionary-LICENSE.txt
# (copied below)
#
# This file is derived from the above dictionary, with slight
# modifications.
# -----
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification,
# are permitted provided that the following conditions are met:
#
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in
# binary form must reproduce the above copyright notice, this list of
# conditions and the following disclaimer in the documentation and/or
# other materials provided with the distribution.
#
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
# -----

```

### 4. Burmese Word Break Dictionary Data (burmesedict.txt)

```

# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
# github.com/kanyawtech/myanmar-karen-word-lists
#
# -----
# Copyright (c) 2013, LeRoy Benjamin Sharon

```

```
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
# disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
# with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
# TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
# THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.
# -----
```

## 5. Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

```
# 7. Database Ownership
#
# The TZ database itself is not an IETF Contribution or an IETF
# document. Rather it is a pre-existing and regularly updated work
# that is in the public domain, and is intended to remain in the
# public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do
# not apply to the TZ Database or contributions that individuals make
# to it. Should any claims be made and substantiated against the TZ
# Database, the organization that is providing the IANA
# Considerations defined in this RFC, under the memorandum of
# understanding with the IETF, currently ICANN, may act in accordance
# with all competent court orders. No ownership claims will be made
# by ICANN or the IETF Trust on the database or the code. Any person
# making a contribution to the database or code waives all rights to
# future claims in that contribution or in the TZ Database.
```

# 14 Directives index

[accept-invalid-recipients \(<source>\)](#)  
[add-date-header \(<source>\)](#)  
[add-message-id-header \(<source>\)](#)  
[add-received-header \(<source>\)](#)  
[add-x-msys-api-header \(SparkPost Traffic Redirection Support\)](#)  
[add-x-virtual-mta-header \(<source>\)](#)  
[address \(<address-list>\)](#)  
[address-file \(<address-list>\)](#)  
[alias \(Global\)](#)  
[allow-auth \(<source>\)](#)  
[allow-cancel-during-transfer \(<domain>\)](#)  
[allow-chunking \(<source>\)](#)  
[allow-empty-x-virtual-mta \(<source>\)](#)  
[allow-mailmerge \(<source>\)](#)  
[allow-priority-interruption \(<domain>\)](#)  
[allow-priority-interruption-during-transfer \(<domain>\)](#)  
[allow-smtpuf8 \(<source>\)](#)  
[allow-starttls \(<source>\)](#)  
[allow-unencrypted-plain-auth \(<source>\)](#)  
[always-allow-api-submission \(<source>\)](#)  
[always-allow-relaying \(Relaying Control\)](#)  
[api-key \(<transmissions-api>\)](#)  
[api-key \(<signals>\)](#)  
[assume-delivery-upon-data-termination-timeout \(<domain>\)](#)  
[auth-password \(<domain>\)](#)  
[auth-username \(<domain>\)](#)  
[authentication-method \(<smtp-user>\)](#)  
[auto-qualify-domain \(<source>\)](#)  
[await-slot \(<smtp-listener>\)](#)  
[backoff-max-connect-rate \(<domain>\)](#)  
[backoff-max-data-volume-rate \(<domain>\)](#)  
[backoff-max-msg-per-hour \(<domain>\)](#)  
[backoff-max-msg-rate \(<domain>\)](#)  
[backoff-max-smtp-out \(<domain>\)](#)  
[backoff-notify \(<domain>\)](#)  
[backoff-reroute-to-virtual-mta \(<domain>\)](#)  
[backoff-retry-after \(<domain>\)](#)  
[backoff-to-normal-after \(<domain>\)](#)  
[backoff-to-normal-after-delivery \(<domain>\)](#)  
[backoff-upon-all-sources-disabled \(<domain>\)](#)  
[bcc \(<source>\)](#)  
[bcc-upon-delivery \(<domain>\)](#)  
[body-pattern-file \(<email-patterns>\)](#)  
[bounce-after \(<domain>\)](#)  
[bounce-upon-5xx-greeting \(<domain>\)](#)  
[bounce-upon-no-mx \(<domain>\)](#)

[bounce-upon-pix-transfer-failure \(<domain>\)](#)  
[bounce-upon-transfer-failure \(<domain>\)](#)  
[broken-auth-clients \(<source>\)](#)  
[check-dkim-inbound \(<source>\)](#)  
[check-iprev-inbound \(<source>\)](#)  
[check-spf-inbound \(<source>\)](#)  
[check-spf-inbound-best-guess \(<source>\)](#)  
[check-spf-inbound-max-void-lookups \(<source>\)](#)  
[check-spf-outbound \(<domain>\)](#)  
[client \(<proxy>\)](#)  
[cold-virtual-mta \(<virtual-mta>\)](#)  
[command \(Pipe Delivery\)](#)  
[connect-timeout \(<domain>\)](#)  
[connection-idle-timeout \(<domain>\)](#)  
[count-moved-records \(<acct-file>\)](#)  
[custom-dsn-from-header \(Global\)](#)  
[custom-message-id-header \(<signals>\)](#)  
[data-send-timeout \(<domain>\)](#)  
[date-header-time \(<source>\)](#)  
[default-smtp-port \(<domain>\)](#)  
[default-virtual-mta \(<source>\)](#)  
[delete-after \(<acct-file>\)](#)  
[delete-file-holders \(<spool>\)](#)  
[deliver-email \(<bounce-processor>\)](#)  
[deliver-email \(<feedback-loop-processor>\)](#)  
[deliver-local-dsn \(<domain>\)](#)  
[deliver-matched-email \(<bounce-processor>\)](#)  
[deliver-matched-email \(<feedback-loop-processor>\)](#)  
[deliver-only \(<spool>\)](#)  
[deliver-unmatched-email \(<bounce-processor>\)](#)  
[deliver-unmatched-email \(<feedback-loop-processor>\)](#)  
[delivery-priority \(<domain>\)](#)  
[disable-acct-records \(<domain>\)](#)  
[disable-mx-rollup \(<domain>\)](#)  
[disabled-source-ip-notify \(<domain>\)](#)  
[discard-as-bounce \(Discard Delivery\)](#)  
[disconnect-on-transient-error \(<source>\)](#)  
[dkim-add-body-limit \(<domain>\)](#)  
[dkim-add-timestamp \(<domain>\)](#)  
[dkim-algorithm \(<domain>\)](#)  
[dkim-body-canon \(<domain>\)](#)  
[dkim-disallow-adding-headers \(<domain>\)](#)  
[dkim-expire-after \(<domain>\)](#)  
[dkim-headers \(<domain>\)](#)  
[dkim-headers-canon \(<domain>\)](#)  
[dkim-identity \(<domain>\)](#)  
[dkim-identity-fallback \(<domain>\)](#)  
[dkim-query-method \(<domain>\)](#)  
[dkim-sign \(<domain>\)](#)  
[domain \(<address-list>\)](#)  
[domain \(<virtual-mta-pool>\)](#)  
[domain-key \(General Queueing and Delivery\)](#)  
[domain-key \(<virtual-mta>\)](#)  
[domain-key \(<virtual-mta-pool>\)](#)  
[domain-key \(<domain-key-list>\)](#)  
[domain-macro \(General Queueing and Delivery\)](#)  
[domain-suffix \(Global\)](#)  
[dsn-format \(<domain>\)](#)  
[dsn-return-default \(<source>\)](#)  
[dummy-smtp-await-slot \(Dummy SMTP \("blackholing"\)\)](#)  
[dummy-smtp-has-chunking \(Dummy SMTP \("blackholing"\)\)](#)  
[dummy-smtp-has-pipelining \(Dummy SMTP \("blackholing"\)\)](#)  
[dummy-smtp-has-verb \(Dummy SMTP \("blackholing"\)\)](#)  
[dummy-smtp-ip \(Dummy SMTP \("blackholing"\)\)](#)  
[dummy-smtp-latency \(Dummy SMTP \("blackholing"\)\)](#)  
[dummy-smtp-port \(Dummy SMTP \("blackholing"\)\)](#)  
[dummy-smtp-update-stats \(Dummy SMTP \("blackholing"\)\)](#)  
[enable-signals \(Global\)](#)  
[engagement-tracking \(<signals>\)](#)  
[file-destination \(File Delivery\)](#)  
[file-format \(File Delivery\)](#)  
[forward-errors-to \(<bounce-processor>\)](#)  
[forward-errors-to \(<feedback-loop-processor>\)](#)  
[forward-unmatched-to \(<bounce-processor>\)](#)  
[forward-unmatched-to \(<feedback-loop-processor>\)](#)  
[from-domain \(<rewrite-list>\)](#)  
[header \(<pattern-list>\)](#)  
[hide-message-source \(<source>\)](#)  
[host-id \(Global\)](#)  
[host-name \(Global\)](#)  
[host-name \(<virtual-mta>\)](#)  
[http-access \(Web-Based Monitor\)](#)  
[http-api-key \(Web-Based Monitor\)](#)  
[http-csrf-protection \(Web-Based Monitor\)](#)  
[http-default-max-items \(Web-Based Monitor\)](#)  
[http-default-sort-order \(Web-Based Monitor\)](#)  
[http-delivery-extra-headers \(HTTP Delivery\)](#)  
[http-delivery-url \(HTTP Delivery\)](#)  
[http-log-data \(Web-Based Monitor\)](#)  
[http-log-requests \(Web-Based Monitor\)](#)  
[http-log-tls \(Web-Based Monitor\)](#)  
[http-mgmt-port \(Web-Based Monitor\)](#)  
[http-mgmt-source & no-http-mgmt-source \(Web-Based Monitor\)](#)  
[http-redirect-to-https \(Web-Based Monitor\)](#)  
[http-strict-transport-security-max-age \(Web-Based Monitor\)](#)  
[http-tls-allow-sslv3 \(Web-Based Monitor\)](#)  
[http-tls-allow-tlsv1 \(Web-Based Monitor\)](#)  
[http-tls-allow-tlsv1.1 \(Web-Based Monitor\)](#)  
[http-tls-allow-tlsv1.2 \(Web-Based Monitor\)](#)  
[http-tls-allow-tlsv1.3 \(Web-Based Monitor\)](#)  
[http-tls-ca-file \(Web-Based Monitor\)](#)  
[http-tls-ciphers \(Web-Based Monitor\)](#)  
[http-tls-tlsv1.3-ciphersuites \(Web-Based Monitor\)](#)

[ignore-8bitmime \(<domain>\)](#)  
[ignore-chunking \(<domain>\)](#)  
[ignore-dsn \(<domain>\)](#)  
[ignore-pipelining \(<domain>\)](#)  
[immediate-tls \(<smtp-listener>\)](#)  
[include \(Global\)](#)  
[include \(<source>\)](#)  
[include \(<domain>\)](#)  
[include \(<virtual-mta>\)](#)  
[include \(<smtp-pattern-list>\)](#)  
[include-headers-from \(General Queueing and Delivery\)](#)  
[include-headers-from \(<virtual-mta>\)](#)  
[include-headers-from \(<virtual-mta-pool>\)](#)  
[invalid-virtual-mta-fallback \(General Queueing and Delivery\)](#)  
[ip-connection-limit \(MX and IP Connection Limit\)](#)  
[iso-times \(<acct-file>\)](#)  
[jobid-header \(<source>\)](#)  
[jobid-prefix \(<source>\)](#)  
[log-auto-rotation \(Logging\)](#)  
[log-commands \(<source>\)](#)  
[log-connections \(<source>\)](#)  
[log-data \(<source>\)](#)  
[log-data \(<transmissions-api>\)](#)  
[log-disabled-ips \(<domain>\)](#)  
[log-file \(Logging\)](#)  
[log-file-world-readable \(Logging\)](#)  
[log-match-info \(<email-patterns>\)](#)  
[log-requests \(<transmissions-api>\)](#)  
[log-resolution \(<domain>\)](#)  
[log-resolution \(<domain>\)](#)  
[log-rotate \(Logging\)](#)  
[log-tls \(<source>\)](#)  
[log-tls \(SSL/TLS\)](#)  
[log-tls \(<transmissions-api>\)](#)  
[log-transfer-failures \(<domain>\)](#)  
[log-verbose \(<signals>\)](#)  
[macro-variables-preserve-case \(Pipe Delivery\)](#)  
[mail-from \(<pattern-list>\)](#)  
[mail-from \(<rewrite-list>\)](#)  
[mailmerge-expands-undefined-variables \(Global\)](#)  
[map-header-to-field \(<acct-file>\)](#)  
[max-cold-virtual-mta-msg \(<domain>\)](#)  
[max-connect-rate \(<domain>\)](#)  
[max-data-volume-rate \(<domain>\)](#)  
[max-errors-per-connection \(<domain>\)](#)  
[max-events-recorded \(<domain>\)](#)  
[max-message-hops \(<source>\)](#)  
[max-message-size \(<source>\)](#)  
[max-msg-per-connection \(<domain>\)](#)  
[max-msg-per-hour \(<domain>\)](#)  
[max-msg-rate \(<domain>\)](#)  
[max-rcpt-per-message \(<source>\)](#)  
[max-rcpt-per-message \(<domain>\)](#)  
[max-rcpt-per-transaction \(<domain>\)](#)  
[max-size \(<acct-file>\)](#)  
[max-smtp-data-volume-rate \(<virtual-mta>\)](#)  
[max-smtp-data-volume-rate \(<virtual-mta-pool>\)](#)  
[max-smtp-in \(<source-group>\)](#)  
[max-smtp-msg-rate \(<virtual-mta>\)](#)  
[max-smtp-msg-rate \(<virtual-mta-pool>\)](#)  
[max-smtp-out \(<domain>\)](#)  
[max-smtp-out \(<virtual-mta>\)](#)  
[max-smtp-out \(<virtual-mta-pool>\)](#)  
[max-smtp-out-per-source-ip \(<domain>\)](#)  
[min-free-space \(<spool>\)](#)  
[min-free-space \(<signals>\)](#)  
[move-interval \(<acct-file>\)](#)  
[move-to \(<acct-file>\)](#)  
[mx-connection-limit \(MX and IP Connection Limit\)](#)  
[name-server \(Global\)](#)  
[notify-of-delay-every \(<domain>\)](#)  
[password \(<smtp-user>\)](#)  
[pattern-list \(<source>\)](#)  
[pickup \(Pickup\)](#)  
[pickup-remove-dot \(Pickup\)](#)  
[pickup-retry-interval \(Pickup\)](#)  
[pix-bug-workaround \(<domain>\)](#)  
[pmc-acct-min-free-space \(Global\)](#)  
[postmaster \(Global\)](#)  
[precached-domains-file \(<dns>\)](#)  
[precached-max-domains \(<dns>\)](#)  
[precached-refresh-interval \(<dns>\)](#)  
[priority \(<spool>\)](#)  
[privacy-policy \(<acct-file>\)](#)  
[privacy-policy \(<signals>\)](#)  
[process-x-dkim-key \(<source>\)](#)  
[process-x-dkim-options \(<source>\)](#)  
[process-x-envid \(<source>\)](#)  
[process-x-job \(<source>\)](#)  
[process-x-schedule \(<source>\)](#)  
[process-x-sp-subaccount-id \(<signals>\)](#)  
[process-x-virtual-mta \(<source>\)](#)  
[queue-filter \(<acct-file>\)](#)  
[queue-priority \(<domain>\)](#)  
[queue-to \(<domain>\)](#)  
[rcpt-to \(<pattern-list>\)](#)  
[record-fields \(<acct-file>\)](#)  
[records \(<acct-file>\)](#)  
[reenable-source-ip-after \(<domain>\)](#)  
[reject-invalid-virtual-mta \(<source>\)](#)  
[reject-iprev-check-fail \(<source>\)](#)  
[reject-iprev-check-permerror \(<source>\)](#)

[reject-iprev-check-temperror \(<source>\)](#)  
[reject-spf-check-fail \(<source>\)](#)  
[reject-spf-check-permerror \(<source>\)](#)  
[reject-spf-check-temperror \(<source>\)](#)  
[relay-address \(Relaying Control\)](#)  
[relay-debug \(Relaying Control\)](#)  
[relay-domain \(Relaying Control\)](#)  
[remove-header \(<source>\)](#)  
[replace-smtp-421-service-response \(<domain>\)](#)  
[require-auth \(<source>\)](#)  
[require-starttls \(SSL/TLS\)](#)  
[require-starttls-before-auth \(<source>\)](#)  
[reroute-to-virtual-mta \(<domain>\)](#)  
[reserved-smtp-in \(<source-group>\)](#)  
[retain-x-dkim-key \(<source>\)](#)  
[retain-x-dkim-options \(<source>\)](#)  
[retain-x-job \(<source>\)](#)  
[retain-x-schedule \(<source>\)](#)  
[retain-x-sp-subaccount-id \(<signals>\)](#)  
[retain-x-virtual-mta \(<source>\)](#)  
[retry-after \(<domain>\)](#)  
[retry-after-dns-error \(<domain>\)](#)  
[retry-interval \(<acct-file>\)](#)  
[retry-recipients-after \(<domain>\)](#)  
[retry-upon-new-mail \(<domain>\)](#)  
[reuse-ssl-session \(SSL/TLS\)](#)  
[rewrite-list \(<virtual-mta>\)](#)  
[rewrite-list \(<virtual-mta-pool>\)](#)  
[route \(<domain>\)](#)  
[run-as-root \(Global\)](#)  
[sender-or-from \(<signals-filter>\)](#)  
[server \(<proxy>\)](#)  
[smtp-421-means-mx-unavailable \(<domain>\)](#)  
[smtp-553-means-invalid-mailbox \(<domain>\)](#)  
[smtp-await-slot \(Global\)](#)  
[smtp-command-timeout \(<source>\)](#)  
[smtp-data-termination-timeout \(<domain>\)](#)  
[smtp-data-timeout \(<source>\)](#)  
[smtp-greeting-timeout \(<domain>\)](#)  
[smtp-hosts \(<domain>\)](#)  
[smtp-ip \(Global\)](#)  
[smtp-listener \(Global\)](#)  
[smtp-max-auth-failure-rate \(<source>\)](#)  
[smtp-out-rampup-interval \(<domain>\)](#)  
[smtp-pattern-list \(<domain>\)](#)  
[smtp-port \(Global\)](#)  
[smtp-server-tls-allow-sslv3 \(Global\)](#)  
[smtp-server-tls-allow-tlsv1 \(Global\)](#)  
[smtp-server-tls-allow-tlsv1.1 \(Global\)](#)  
[smtp-server-tls-allow-tlsv1.2 \(Global\)](#)  
[smtp-server-tls-allow-tlsv1.3 \(Global\)](#)  
[smtp-server-tls-ca-file \(Global\)](#)  
[smtp-server-tls-certificate \(Global\)](#)  
[smtp-server-tls-ciphers \(Global\)](#)  
[smtp-server-tls-v1.3-ciphersuites \(Global\)](#)  
[smtp-service \(<source>\)](#)  
[smtp-source-host \(Global\)](#)  
[smtp-source-host \(<virtual-mta>\)](#)  
[smtp-tls-allow-sslv3 \(SSL/TLS\)](#)  
[smtp-tls-allow-tlsv1 \(SSL/TLS\)](#)  
[smtp-tls-allow-tlsv1.1 \(SSL/TLS\)](#)  
[smtp-tls-allow-tlsv1.2 \(SSL/TLS\)](#)  
[smtp-tls-allow-tlsv1.3 \(SSL/TLS\)](#)  
[smtp-tls-ca-file directive \(SSL/TLS\)](#)  
[smtp-tls-ciphers \(SSL/TLS\)](#)  
[smtp-tls-tlsv1.3-ciphersuites \(SSL/TLS\)](#)  
[source \(<smtp-user>\)](#)  
[source \(<smtp-listener>\)](#)  
[source-group \(<source>\)](#)  
[source-ip-max-connect-rate \(<domain>\)](#)  
[source-ip-max-data-volume-rate \(<domain>\)](#)  
[source-ip-max-msg-rate \(<domain>\)](#)  
[source-ip-max-msg-rate \(<domain>\)](#)  
[source-ip-max-smtp-out \(<domain>\)](#)  
[sparkpost-ip-pool \(SparkPost Traffic Redirection Support\)](#)  
[sparkpost-sandbox \(SparkPost Traffic Redirection Support\)](#)  
[spool \(Global\)](#)  
[spool-delete-corrupted \(Global\)](#)  
[spool-max-recipients \(Global\)](#)  
[spool-min-free-space \(Global\)](#)  
[suppress-local-dsn \(<source>\)](#)  
[suppression-lists \(<source>\)](#)  
[sync \(<acct-file>\)](#)  
[thread-min-priority, thread-max-priority \(Global\)](#)  
[thread-reuse \(Global\)](#)  
[tls-allow-sslv3 \(<smtp-listener>\)](#)  
[tls-allow-sslv3 \(<transmissions-api-listener>\)](#)  
[tls-allow-tlsv1 \(<smtp-listener>\)](#)  
[tls-allow-tlsv1 \(<transmissions-api-listener>\)](#)  
[tls-allow-tlsv1.1 \(<smtp-listener>\)](#)  
[tls-allow-tlsv1.1 \(<transmissions-api-listener>\)](#)  
[tls-allow-tlsv1.2 \(<smtp-listener>\)](#)  
[tls-allow-tlsv1.2 \(<transmissions-api-listener>\)](#)  
[tls-allow-tlsv1.3 \(<smtp-listener>\)](#)  
[tls-allow-tlsv1.3 \(<transmissions-api-listener>\)](#)  
[tls-ca-file \(<smtp-listener>\)](#)  
[tls-ca-file \(<transmissions-api-listener>\)](#)  
[tls-certificate \(<smtp-user>\)](#)  
[tls-certificate \(<smtp-listener>\)](#)  
[tls-certificate \(<transmissions-api-listener>\)](#)  
[tls-ciphers \(<smtp-listener>\)](#)  
[tls-ciphers \(<transmissions-api-listener>\)](#)  
[tls-tlsv1.3-ciphersuites \(<smtp-listener>\)](#)

[tls-tls1.3-ciphersuites \(<transmissions-api-listener>\)](#)  
[too-many-rcts-fails-message \(<source>\)](#)  
[total-max-smtp-in \(Global\)](#)  
[total-max-smtp-out \(Global\)](#)  
[trace-dkim-check \(<source>\)](#)  
[trace-ipv6-check \(<source>\)](#)  
[trace-spf-check \(<source>\)](#)  
[track-recipient-events \(<domain>\)](#)  
[track-time-first-attempt \(Global\)](#)  
[type \(<domain>\)](#)  
[upload-url \(<signals>\)](#)  
[use-dane \(SSL/TLS\)](#)  
[use-proxy \(<virtual-mta>\)](#)  
[use-starttls \(SSL/TLS\)](#)  
[use-unencrypted-plain-auth \(<domain>\)](#)  
[user-string \(<acct-file>\)](#)  
[verp-default \(<source>\)](#)  
[virtual-mta \(<virtual-mta-pool>\)](#)  
[world-readable \(<acct-file>\)](#)  
[write-timeout \(<acct-file>\)](#)