MPP Version 4 Configuration Guide
August 2008

Table of Contents

1 Introduction

This guide represents a complete re-write of the MPP documentation. For the first time we have centered our documentation on our GUI rather than the configuration file. We have attempted to present information in clear and concise manner, avoiding obvious explanations while showing practical uses of a command or feature.

NOTE: The documentation is structured around the format of the MPP GUI menu tree

1.1 MPP Configuration File

The MPP configuration file, mppd.conf.xml, has become unwieldy so we are focusing heavily on or GUI. However, if you still love hand editing configurations you can edit /usr/local/MPP/mppd.conf.xml as you like. We don't recommend it for most applications, however. The MPP GUI will reflect any changes that you make in the configuration file and you can switch back and forth between configuration styles if you like.

1.1.1.1 Avoid Mistakes in the Configuration File

If you make a mistake hand editing the configuration file and then reload mppd your new configuration file will not be loaded, however mppd will stay running with the old working configuration. This accounts for a large percentage of support cases that we deal with. MPP will do it's best to stay running in spite of a bad configuration. On the other hand, if there is a bad configuration and mppd is stopped, not reloaded, mppd will not load the configuration file and mppd will not start.

While hand editing the configuration file is an option if you are more comfortable in this environment please be careful. Read the DTD, which shows all XML options, to see the available options and use the GUI enabled configurations as a model.

1.2 MPP Documentation Set

The complete MPP documentation set includes this manual, the MPP Installation Guide, the MPP Archive and Quarantine Administration Guide, release notes in your installation package and the knowledgebase found on http://messagepartners.com/esupport.

2 Status

2.1 Status

The status page provides basic operational statistics, message count, licensing information and shortcuts to configure the primary MPP services.

2.1.1 Server Status

Server status should be green if MPP is operational. If server status is Red it is advisable to check the logs of MPP to ascertain the reason that MPP is not running. Logs can be accessed from the shortcut on the status page or in Status->Monitor and Report->MPP Logs.

2.1.2 License Key

This section shows information about the MPP license key and allows you to install a new key. Please note that MPP trial keys will mark each message with a scanned by MPP link. If you would like to trial MPP without the message marks please contact Message Partners or your local partner to obtain a trial key that will remove this mark.

2.1.3 Services

This section contains links to configure the primary services of MPP. Services show as enabled if they are configured in the default MPP policy group. Please see the Advanced->Policy Engine section for more details on policy configurations

2.1.4 Shortcuts and Plug-in Updates

This section provides shortcuts to common MPP tasks and an interface to update certain MPP Plug-in Modules

2.1.4.1 Content Filter Updates

MPP provides an interface to update filter modules that are integrated in the MPP installation package.

- Sophos daily updates should be scheduled a few times a day and Sophos monthly updates should be scheduled once a month, usually in the first week.
- Cloudmark Cartridge updates should be run once a month, micro updates need not be scheduled.
- Mailshell updates should be scheduled a few times a day.

Please see the MPP Installation Guide for more details on content module updates.

2.2 Monitor and Report

2.2.1 Statistics

This section contains critical statistics to monitor MPP's performance.

2.2.1.1 Message Rates

Statistics are gathered from MPP logs using a parser script that is automatically run a few times a day. The frequency can be changed in crontab for user root. To change the date range of statistics, force a rebuild or change the location where the parser script should look for statistics click the *change date range* link.

Most tabular stats are self-explanatory, here are the less obvious explained:

WBL Matched	Messages that matched a spam white or black list
RBL Matched	Messages that hit a Real Time Black hole list
Unauthorized	Messages that matched a content filter
Spamatraps	Messages from an IP that matched a spam trap entry
Client-Host BL	Messages that matched a client-host (SMTP) black list or auto-blacklist.

2.2.1.2 Per Engine Statistics

These are generated from the log parser script and give general guidelines as to engine hits but are not completely accurate due to configuration options. There are more accurate statistics available from tAdvanced->Message Tracking if you have enabled this feature.

2.2.1.3 Thread Utilization

These statistics show how MPP is using the scanning threads configured in Advanced->Daemon. If you see many threads with zero counts or very low counts then you have configured too many scanning threads. Too many or too few scanning threads will impact performance so it is a good idea to monitor these statistics to look for an even distribution of scanning amongst the threads.

2.2.1.4 Top 5 Viruses

Self-Explanatory, derived from scanning reports.

2.2.1.5 Greylist Report

These statistics are only available if Greylisting has been configured. Greylisting is only available when the Postfix Policy Server is enabled.

2.2.2 Graphs

This section provides graphical representation of the statistics. Graphs are only available if lib-gd and the GD perl module are installed. http://www.libgd.org/Main_Page

2.2.3 Email Reports

Configure this section to have MPP statistics sent to you via email on a regular basis.

2.2.4 MPP Logs

This section allows you view MPP logs in real time. You can filter for a specific string in real-time, such as a domain name, to only see log lines related to an area of interest. To view logs in real time select *view*, press *stop realtime viewing* to stop viewing the logs. If the stop process fails select *clear log* to clear the window contents and view again.

2.2.5 Postfix/Mail Logs

This option provides a real-time view of the system mail logs.

2.2.6 Search Logs

Use this section to search existing MPP logs. There is an option for sub-searches of query results.

2.2.7 Manage Logs

This section is used to delete, compress or archive log files on the MPP server.

3 Services

The Services section provides an interface to configure the essential services of MPP. Since MPP is extremely flexible and configurable a concerted effort has been made to move less frequently configured items to 'advanced' sections to reduce configuration clutter.

MPP supports multiple configuration policies to apply configuration options to groups of users. More information is available in the Advanced->Policy Engine section. The dropdown menu on the top right hand of the screen shows which policy you are configuring. Most installations have one default policy.

> **NOTE:** For many configuration options there are options to store lists in the configuration file, in text files, in a MySQL database or a LDAP directory. Storing long lists is not practical in the configuration file, hence the option to store lists in text files. The downside of storing information in the configuration file or text files is that the lists are only loaded if the mppd daemon is restarted.
>
> Store configurations in databases or directories to have changes take effect immediately or in configurable reload intervals or to share lists between multiple instances of MPP.

3.1 Connections

This section allows you to configure options pertaining to the SMTP connection. MPP has the most control over SMTP connections with the Postfix MTA, especially when the Postfix Policy server is enabled.

3.1.1 Block Connection/Never Scan Connections

Add IP addresses or host names of SMTP servers that should be exempt from all MPP scanning or always blocked. This feature is known as 'client-host white-black lists (CH-WBL's). In most cases simply enter the IP's or hostnames in the text boxes provided and save and restart to have your configuration options applied.

If you have long lists of hosts to block or allow use a text file and enter the full path to the file such as /usr/local/MPP/mpp_wbl. If the file does not exist MPP Manager will create it. The format of entries in the file are host, value with one per line, for example:

raeinternet.com, whitelist
12.12.124.124, blacklist
192.154.121.152, whitelist

Use a database to store long lists that are shared among multiple instances of MPP or if changes need to be real-time without requiring a reload of MPP.

> **Note:** MPP has multiple levels of white and black lists. Client-host WBL's exempt or block SMTP hosts from all scanning, per-user spam WBL's exempt users from spam checks, optionally content filters

3.1.2 Reputation Lists

Use this section to enter Real Time Black Hole lists that MPP should check. Only use reject if you use the Postfix MTA, for all other MTA's use discard or quarantine options to avoid being a source of backscatter email.

3.1.3 Greylisting

Greylisting is a process where all SMTP servers that connect to MPP must first pass a 'tempfail' test before they can send. Many spam bots do not respond to temporary failure messages, thus they will never be able to send to your network. Greylisting can dramatically reduce spam but can also introduce delays in email processing that will annoy some. For example if you are waiting for an instant confirmation on an auction bid or reservation and this email is delayed by thirty minutes it can be quite frustrating. Luckily MPP has many controls to apply greylisting to groups of users or to exempt IP's and domains from grey list processing, however, greylisting should be applied with this caution in mind.

Greylisting has the following requirements:

1) Postfix must be the MTA and the Postfix Policy Server must be enabled.
2) A MySQL database must be available

To configure greylisting enable the grey list feature and enter the database credentials for the MySQL server. Other options can generally be left alone and are self-explanatory.

The Greylist Whitelist feature allows you list hostnames or IP's that are exempt form Greylist processing and the URI option should be used for very long lists of exemptions.

3.1.4 Spamtraps

Spamtraps allow you configure email addresses or patterns as 'traps'. Any IPs that sends to these traps is automatically placed in a temporary automatic client-host blacklist. MPP Manager has a GUI interface to configure simple regular expressions but the real power of spamtraps is unlocked with a full understanding of the construction of regular expressions.

Spamtraps can be stored in text files for complex regular expressions, but this is not required in most cases. The simplest method to add a spam trap address is to select *add new* and then add a 'wildcard expression. MPP will automatically create the correct regular expression syntax for a wildcard match of the value that you provide. For example raeinternet.com will match any host with raeinternet.com somewhere in the name.

After defining the spam traps define the time to block hosts that are matched and the storage method. Use RAM storage for small installations and MySQL is you wish to share the spam traps IP's amongst multiple instances of MPP.

Enter IP's or hostnames to exempt from spam trap processing the spam traps white list section. If you have a long list of IP's or hostnames to exempt then put them in a text file and define a path to the text file in the URI field, /usr/local/MPP/spamtrap. MPP will make the text file if it does not exist.

3.1.5 Sender Policy Framework

Sender Policy Framework (http://www.openspf.org/) is a standards based method of verifying that host is allowed to send email for a domain. MPP supports SPF on a per-policy basis. When SPF is enabled MPP will query the DNS server for the sending domain to check for a valid SPF record and will take the configured action based on response.

It is only recommended to take an action on spffail, using a condition on spfpas for example can cause sapm with legitimate spf records to not be scanned.

If you want to take actions with spf states other than fail it is highly recommended to use SPF in conjunction with MPP Spam Scoring.

3.1.5.1 Enabling SPF

To enable SPF set to enable and in most cases this should be enough. There are numerous options to customize for advanced users, must are self-explanatory, but a brief background discussion will help to understand the options.

3.1.5.2 SPF Advanced Options

The following options are available for SPF results:

- *quarantine*: all further checks (spam, virus, etc.) will be skipped, message will go to quarantine and then ON_QUARANTINE_... will be performed.

- *reject*: all further checks will be skipped, message will not be relayed further, reject reply will be returned to MTA
- *discard*: all further checks will be skipped, message will not be relayed further, accept reply will be returned to MTA.
- *pass*: all further checks will be skipped, message will be relayed further, accept reply will be returned to MTA.
- *markheader*: all further checks will be skipped, header specified with SPF_HEADER option will be added to message, message will be relayed, accept reply will be returned to MTA.
- *scan*: "do nothing" message will be scanned further as usual.
- *defer*: all further checks will be skipped, message will not be relayed further, and temporary failure reply will be returned to MTA.
- *add_mpp_spam_score*: value specified with corresponding MPP_SPAM_SCORE_SPF_... will be added to total MPP spam score, message will be scanned further as usual.

MPP users a binary, spfquery, from /usr/local/MPP/spf. This is invoked for each SPF query, however, optimizations have been made to reduce system overhead for this process. All options in the advanced section are provided to fine tune how this command is called. Defaults are recommended for most cases, however, slow network response times or low system resources may make it necessary to increase timeout values for read/write operations.

3.1.5.3 SPF And MPP Custom Spam Scoring

SPF is integrated in MPP custom spam scoring to create aggregate scores form SPF results, spam scanner scores, RBL sites and content filter expressions.

3.2 Antivirus

This section controls how MPP reacts when a virus is detected by a plug-in module. Available options are discard, reject, and quarantine, mark header, disinfect and delete. Delete will attempt to delete a virus-infected attachment. Disinfect will attempt to clean the virus from a legitimate payload. Since many viruses can't be disinfected or deleted as the emails to which they are attached contain no legitimate payload, as in the case of worms, these actions should be used with caution.

3.3 Antispam

The antispam section contains many configurations related specifically to antispam protection. However, since antispam protection is a central application for MPP this is not a comprehensive place for all antispam configurations. For example spam traps and thresholds are effective antispam tools, however, they are configured in other sections that fall into the organizational structure of the configuration. For example reputation lists are an antispam measure but since they related to SMTP connections we put them into the connections area.

3.3.1 Threshold Scenarios

Each MPP policy has three spam thresholds – high, medium and low – each with its own action. Possible actions are marksubject, markheader, forward, quarantine, reject, discard, and pass.

Each MPP plug-in module submits a numerical score to MPP telling us the likelihood that a message is spam.

In this section plug-in scores are correlated to MPP thresholds and actions are assigned to each threshold. These settings are applied to an entire MPP policy but spam actions can be applied to individual users in the subsequent configuration section.

It is generally not a good idea to move the threshold scores from their defaults unless you have a very good reason to do it. If you find that an engine is not reliable at the low threshold it is better to set that action to pass or markheader rather than moving the threshold higher.

In the case of the Co touch plug-in do not adjust the thresholds.

3.3.1.1 Understanding Spam Actions

Pass	No action is taken on the message based on spam score and the message continues on processing path.
Discard	The message is silently discarded. Considered bad etiquette, but it sure can be convenient.
Marksubject	Renames the subject of the email with the subject of your choice.
Markheader	Adds a header of your choice to an email
Quarantine	Mail is stored in quarantine and discarded
Forward	Mail is forwarded to an address of your choice

Note on Quarantine and Forward - By default email is discarded after it is quarantined or forwarded, however, this is configurable.

Note on MPP Headers – There are many options for header additions. The markheader option here applies to a header applied to each message identified as spam. To control scanned by MPP headers check in Services -> Utility.

Macros Available for Header and Subject Marking

- %RCPT_NAME%: the user id from the receiver's e-mail address before the '@' symbol.
- %SUBJECT%: the original subject of the message.
- %RCPT_DOMAIN%: the domain of the receiver's e-mail address after the '@' symbol.
- %RCPT%: the full e-mail address of the receiver.
- %SENDER%, %SENDER_NAME%, %SENDER_DOMAIN% are also available
- %GROUP% is the name of the group that the user belongs to.
- %SPAMSCORE%. SpamAssassin will calculate a value based on numeric weightings of matches. Partial matches will have a value less than one.
- %SCORESYMBOLS% will be replaced with a number of symbols (default is *) equal with the difference between SpamAssassin current score and SpamAssassin max score between brackets, e.g. (****). The * symbol can be replaced using the parameter spam_spamassassin_subject_symbol. Only applicable in subjects, not in other message headers.
- The %TYPE% macro that expands to one of 'virus', 'spam', 'harass', 'other' replaces the SPAM_QUARANTINE_DIRECTORY and HARASS_QUARANTINE_DIRECTORY options from version 2.0.

3.3.2 Per-user Spam Settings

Within each policy group MPP offers per-user antispam settings to override the group actions. The available per-user actions are white and black lists and spam actions. Per-user settings can be stored in the MPP configuration file, in a text file (convenient for very long lists), or in a MySQL database.

MySQL is the ideal choice for per-user storage if you have long lists of user preference\s and you require instantaneous changes. MySQL is also ideal if you have multiple MPP instances that need to share the same data.

If you store per-user settings in a text file or in the default location MPP must be restarted for changes to apply. Restarting MPP is non-disruptive. If you don't make many changes to per-user settings it is recommended to leave all settings in the default and restart MPP after you make changes.

While it is possible to configure the per-user settings here, it is preferable to do this within the MPP quarantine manager for a few reasons. For one, individual users can make the changes for themselves, additionally, a domain administrator can make changes for there entire domain and finally, it is a bit clearer to understand.

3.3.2.1 Per-user Spam Actions

Define the domain or email address that the action applies to and then define the spam action. Again, it is recommended to make these changes within the MPP quarantine manager.

3.3.2.2 Per-User WBL's

Regardless of storage method a few conventions apply. Storage methods are configuration file, text files or database. Use database or text files for long lists, leave as default for all other applications.

User – The email address for which the WBL entry applies.

Contact – The contact affected by the WBL, think of an address book, wildcard (*) accepted

Direction – Refers to user, if they are sender or recipient

For example, my email address is mike@gmail.com and I want to whitelist all email from Citibank.com.

User = mike@gmail.com, contact = citibank.com, direction recipient

I am the domain administrator for company.com and I want to whitelist all outgoing email from spam checks.

User = company.com, contact = *, direction = sender

I am mike@gmail.com and I want to blacklist all email from myex.com

User = mike@gmail.com, contact = myex.com, direction = recipient

3.4 Customer Spam Scores

Custom spam scoring is a major MPP feature module that allows you to create aggregate scores from MPP tests. Currently custom spam scoring supports plug-in module spam scores, RBL matches and SPF results as scoring metrics. Here are some common reasons to use MPP custom spam scoring.

- Mark a message header if there is an RBL match rather than make a reject decision.
- Aggregate the scores of multiple spam engines to make a quarantine decision
- Aggregate SPF, RBL and content scanner results to create one score.

The basic idea behind custom spam scoring is that you create an aggregate score for each threshold, say 10 for high, 8 for medium and 6 for low. Then you assign a point value to a test. For example, a Cloudmark high score is 7 and a Spamhaus RBL hit is 3. Now you need both a Cloudmark high score and an RBL hit to make a spam decision.

3.5 Spam Score Headers

Use this control to add the spam score , from an anti-spam plug-in or from MPP custom spam scoring, to each spam email. You may customize the name of the header or use our default X-Spam-Score

3.5.1 Add SpamAssassin Rule Hits

This control allows you to insert the complete SpamAssassin score report in the header of an email. We do not recommend enabling this feature because performance degrades considerably, but it is an available option.

3.5.2 Alternate spam engine subject symbol

By default, if you are using the %SCORESYMBOLS% macro as a subject mark then MPP will put the * character for each number of the spam score. For example, if the spam score is 20 then 20 *'s would be inserted in the message subject. Use this option to use a character different than * as the symbol.

3.6 Archival

The Archival section controls where email is stored and fine-tunes which email stored. There are many archive controls in the MPP Archive Review application, MPP Manager, such as import controls, user access.

3.6.1 Location of Email Archive

MPP email archives can be stored as files, in a database or in a hybrid metadata approach.

3.6.1.1 Archive to Directory

When this option is enabled all email is stored in the file system as MIME encoded files. The file path is constructed with macros as shown below. File paths can be scaled with %RAND16% and %RAND64% macros. These create randomly named directories that can be nested to avoid file system limits on the number of files within a directory.

NOTE: It is mandatory to use the %RCPT_DOMAIN% macro for the archive review application to work correctly. It is not recommended to use file f

- `%RAND16%`: generates random number between 00 and 0F for use in directory names
- `%RAND256%`: generates random number between 00 and FF
- `%RCPT_NAME%`: the user id from the receiver's e-mail address before the '@' symbol.
- `%SUBJECT%`: the original subject of the message.
- `%RCPT_DOMAIN%`: the domain of the receiver's e-mail address after the '@' symbol.
- `%RCPT%`: the full e-mail address of the receiver
- `%SENDER%, %SENDER_NAME%, %SENDER_DOMAIN%` are also available
- `%GROUP%` is the name of the group that the user belongs to.
- `%TYPE%` macro which expands to one of 'virus', 'spam', 'harass', 'other'
- `%UID%` UserID, same as %RCPT_NAME% except that user should be a Unix system user. If the user doesn't exist it will be replaced with "root".
- `%GID%` GroupID, same as %RCPT_NAME% except that user should be a Unix system user. If the user doesn't exist it will be replaced with "root".

Here is an example directory location:

```
/usr/local/MPP/archive/%RCPT_DOMAIN%/%RCPT_NAME%/%RAND16%
```

Email stored on the file system cannot be indexed for full text searching as of this writing.

3.6.1.2 Archive to MySQL Database

Define the MySQL database permissions for your message store and define if the entire message is stored in the MPP message database or if only 'metadata' is stored in the database while the entire message is stored on the file system.

> **NOTE:** Storing the complete message in MySQL is the recommended archive storage message method. MySQL replication is strongly recommended for scalable storage.

If you opt to use metadata then information about the message is stored in a database while the actual message is stored on a file system, hence, you must define a path to store the email on the file system.

If you are storing the complete message in MySQL set the max storage size equal to the largest field size supported in MySQL. Messages that exceed this amount will be stored on the file system. If you have many large files file system or metadata storage may be more appropriate than complete MySQL storage.

3.6.1.3 Archive to Maildir

Maildir storage is identical to file system storage, however, email is stored in the Maildir format. This is useful if you want standard MUA's such as Squirrelmail to access MPP email archives. Besides directory name there are two additional required fields – owner and group for the messages, %UID% and %GID%

Sample URI's

```
maildir:///usr/local/MPP/quarantine/Maildir
maildir:///usr/local/MPP/quarantine/Maildir:root,root
maildir:///usr/local/MPP/quarantine/%GID%/%UID%:%UID%,%GID%
maildir:///home/%UID%/Maildir:%UID%,%GID%
maildir:///home/vpopmail/domains/%RCPT_DOMAIN%/%RCPT_NAME%:vpopmail,vchkpw
```

3.6.1.4 Archive to Message Store

MPP has a hierarchical message storage model that can be deployed in larger sites. Even for smaller sites it makes because in this model MPP on your MTA will never have a direct connection to your database. This is important because large emails or database failures can impede mail flow if MPP must wait for the database to complete a write.

In the hierarchical model a remote instance of MPP will use ESMTP to transport email messages to a remote MPP that maintains a direct connection to the database. This way any failure or slowness in the database will not affect email flow.

The 'remote' instance of MPP can be on the same server or on a different server. If the email store is on a remote server use this type of socket identifier:

inet:10027@192.168.56.12

If the message store is on a separate instance of MPP on a local server use this type of socket path:

unix:/var/run/other.mpp

If you are considering this model of storage it is recommended to work with MPP support. This is a very scalable model for MPP archive storage.

Setting up the 'remote' instance of MPP, or message store, is handled in the advanced section.

3.6.1.5 Archive Message Settings

Archive MS Exchange Journaled Email is an analog for MPP option of archive with with/without SMTP envelope. MPP supports Message Journaling as found in Exchange 2003 and other journaling schemes such as email sent via Postfix always_bcc.

When this option is set to yes MPP will archive messages with information from the SMTP envelope. When this option is set to no, MPP will extract the receipt/sender information from the actual message rather than from the SMTP envelope.

Note: When archiving journaled email set 'Action on Archive Success' to discard to avoid the redelivery of messages after archival.

3.6.1.6 Archiving MS Exchange 2007 Email

MS Exchange 2007 introduced new Journal formats and they are supported by MPP. When this is enabled MPP will archive the journal reports with correct sender/receiver information, however, the actual message will be stored in TNEF format, just as Exchange encapsulates the message.

3.6.1.7 Archive Other Message Types

These commands are self-explanatory - archive infected, spam, malformed, etc.

3.6.1.8 Actions

Set on archive success to discard when you are archiving journaled email, otherwise leave as the default.

3.6.2 Importing Email into MPP Archives

MPP can import email from MIME Files, IMAP Message Stores or MBOX files. IMAP import is managed from MPP Manger in the Archive -> Import section. File import is managed from an available script which can be obtained from support.

3.7 Content Inspection

Content filtering is a major feature module of MPP and is supported by our own content filter engine. The content filter module has been completely redesigned in MPPv4 to be more efficient and far more flexible.

The MPP GUI assists with creation of simple regular expressions with an intuitive editor, however, the real power of MPP content filtering is unlocked with an understanding of regular expressions. MPP supports many formats of expression including Perl, egrep, emacs, POSIX, awk, sed, EMCAScript, Javascript and more. MPP supports ASCII and UTF-8 expressions. A detailed explanation of regular expression construction can be found in the Appendix. In most cases the MPP GUI provides all of the tools that you will need to construct powerful expressions.

Some uses of MPP content filtering include…

- Surveillance – Forward all email that matches some criteria to a supervisor.
- Quarantine – Quarantine all email that matches a filter and deliver the message or stop it for review.
- Blocking – Block email that matches some criteria from leaving our entering your organization
- Routing – Route email based on content to some department or mailbox for processing

The content filter module is one of the most powerful and flexible modules of MPP and there are endless possibilities to use content-based controls with MPP.

The key to MPP's powerful content filtering module is the breadth of expressions and matching algorithms that we support combined with content filtering actions and MPP policies. Since MPP content filter policies can be applied to groups of users they can be applied to small groups within your organization.

3.7.1 Content Filter Actions

Messages that match a static content filter can be forwarded, delivered, rejected, quarantined or discarded.

3.7.2 Header Filters

Header filters are applied to all message headers, including the subject.

3.7.3 Message Content Filtering

Message content filtering is applied to text within the body of a message.

3.7.4 Attachment Filtering

MPP supports filtering by attachment name, but since this is typically used for extension blocking this capability is found in the Utility->Extension Blocking section. MPP can filter within text based attachments, please contact support for more information regarding this capability.

3.7.5 Regular Expressions

All MPP filters used regular expressions, however, the GUI obviates the need to understand regular expressions for most cases. As outlined previously there are tremendous capabilities to construct complex expressions with MPP, but the GUI will suffice for most cases.

Long regular expressions can be stored in text files defined in the ADVANCED section of the content inspection page.

To add a regular expression using the editor clicks SHOW next to ADD EXPRESSION. There are two options available – to use a wildcard expression or create your own expression. The wildcard option requires no knowledge of regular expressions and there is a drop down of common message headers to filter on.

For example, if you wanted to block all email with 'sex' in the subject you would select the wildcard filter, select the Subject: header and type sex in the text box. If you want to match words like Viagra, but you know that the spelling is usually off, you can try via*. The help pop-up provides some useful examples.

You can add expressions that don't use wild cards, such as email address in the Regexp filter section. MPP will add the regexp symbols corresponding to case sensitivity, single or multiple line matches.

In most cases these are sufficient for filters, however, if you are comfortable with expressions you can select SHOW REGEX LIST FILE and modify the regexp file manually. Again, see the Appendix for more details on complex expression construction.

3.8 Access Control Lists

Access Lists or ACL's are a powerful module of MPP that allow you to limit the senders or receivers of email on your email system. This is useful for a number of applications:

- Check valid recipient lists
- Limit employee communication
- Quarantine email sent to or from certain locations
- Reject email from unauthorized sender/receiver pairs
- Create 'Chinese Firewall' types of environments

ACL's can be stored in the MPP configuration file, text files, MySQL databases or an LDAP directory. Use text files for long lists and databases or directories for long lists that are very dynamic. The default storage is fine for most applications.

There are two types of matches for ACL's. Senders must be on the list, or they must be off the list, in other words, either exclusive or non-exclusive to the list.

Actions for ACL violations include quarantine, discard or reject.

MPP can limit by sender, receiver or sender or receiver. Sender or receiver is seldom used.

The format for ACL text files is *address,direction*, one entry per line. It is recommended to only use recipient ACL's and specify direction in the ACL list – either in or out.

Example ACL file:

mike@raeinternet.com, in

mike@raemail.net, out

3.9 Utility

The utility section controls features that relate to MPP as a general-purpose email processing utility.

3.9.1 Signatures

Signatures are text that is appended to each email. Signatures can be stored in text files or in the MPP configuration. The MPP configuration is fine for most cases. MPP supports both any character encoding of signatures, however, the character set of the email must match the character set of the signature. If an email is ASCII encoded and you have defined a UTF-8 signature the signature will not be applied in order to preserve the integrity of the email message.

MPP takes care to preserver the integrity of a message when inserting a signature. Signatures are added in the message for text-formatted messages and as a new body part for multipart messages. You may use HTML code in the signatures but HTML code will not display properly in text-formatted emails.

How to use a text file for signatures:

1) Define the name of the text file in the text box next to 'Append contents of file,' and then save the configuration.
2) Edit the file

3.9.2 Add MPP Header to Scanned Emails

If enabled MPP adds one header for each content module that scans a message:

`X-Scanned-By: MPP/Cloudmark http://www.messagepartners.com`

3.9.3 Custom header added to every message

The custom header can be tailored to suit your needs and it is added to each message.

3.9.4 Message Conditions

3.9.4.1 Maximum File Size to Scan

Maximum size to scan and action if file exceeds this value. There is a corresponding configuration option in Advanced->Scanners, max file size for spam engine to scan.

3.9.4.2 Maximum Recursion Level

This controls how many levels of MIME recursion MPP should scan. There are DoS attacks with thousands of levels of recursion.

3.9.4.3 Messages Without Recipients

Some worms and viruses will send invalid messages without recipients in the messages, this option controls how these are handled.

3.9.4.4 Empty Messages

Empty messages are a byproduct of worms and viruses and this option controls how they are handled.

3.9.5 Block Attachments

To enable attachment-blocking enable the radio button and click save. Upon save a new text box will appear where you can enter a comma-separated list of extensions to block:

exe, mp3, com

3.9.6 Strip Attachments

Attachment stripping, or body stripping, is a major feature of MPP. The idea behind bodystripping is to remove a message part such as an attachment or MIME Type and replace it with a link to the file. This is very useful for a number of applications:

- Stripping and storing large attachments from outgoing mail, thus avoiding attachment size limitations at a remote site.
- Removing reports from automated emails and placing in processing queues.
- Allowing monitors to examine outbound attachments
- Removing potentially malicious files from incoming email

3.9.6.1 Configuration Concepts

To enable bodystripping define a pattern to match , a storage path and a retrieval link temlate. For example, you can configure that all .pot files over 10MB are stripped from emails and saved on an ftp or http server and the attachment is replaced with a link to the stored file. .

Here is an example that illustrates body stripping

User A sends 25MB PowerPoint file to Users B C and D, thus creating 4 copies of this file; 1 each for Users B, C and D and 1 in the sent items of user A.

With body stripping enabled when user A sends the 25MB attachment it is replaced with a link to a user specific ftp directory. Users B, C and D see a link to the file, but only access the file if they need it. After the attachment is stripped it is no longer a part of the end-users email corpus and never enters their mailbox files.

Steps to Enable Body Stripping

1) Define minimum file size to match, in bytes. E.g. 100000 is 100kb.
2) Define regular expressions to match the name of MIME type or MIME part name to strip.
    a. /.*\.pdf$/ will match file name.pdf, all files with .pdf extension
    b. /pdf/ will match pdf anywhere in the file name
    c. Here is a complete regular expression list of extensions

```
/(.*\.gif$)|(.*\.jpeg$)|(.*\.bmp$)/i
```

3) Specify the location where the MIME part should be stored

    file:///usr/local/MPP/attachments/$FILENAME$:apache,apache,0644

4) Specify the location where users should retrieve the files

http://mydomain.com/body_strip/$MESSAGEID$/$FILENAME$

ftp://mydomain.com/pub/body_strip/$SENDER_DOMAIN$/$MESSAGEID$/$FILENAME$

MPP takes care to insert the retrieval link into the email with the correct MIME type. Hence, there are three formats to choose – HTML, Text, External-Body. You may define the order that the file types are attempted or leave as default.

There is the option to define both an HTML and Text version of the text that is inserted into an email after stripping.

Text version:

```
Body part of type $CONTENTTYPE$/$CONTENTSUBTYPE$ was stripped by MPP.
It can be found at $LINK$
```

HTML Version:

```
<html>
<body>
Body part of type $CONTENTTYPE$/$CONTENTSUBTYPE$ was stripped by MPP.
<br>
It can be found at <a href="$LINK$">$LINK$</a>
</body>
</html>
```

As in other places, MPP can store this information in text files or in the configuration. Text files are suitable for longer messages. If you wish to use text files, specify the path, save the configuration, then edit the file.

3.9.6.2 Location Macros

`$SYSTEM_DOMAIN$` - name of the current system

`$SENDER$` - full sender string (I. e. name@domain.com).

`$SENDER_NAME$` - just sender name (i. e. name for name@domain.com).

`$SENDER_DOMAIN$` - just sender domain (i. e. domain.com from name@domain.com).

`$MESSAGEID$` - message id as extracted from message headers or, if absent, generated at runtime.

`$RAND16$` - random hexadecimal number between 00 and 0F presented as string. This number is stripping-session-wide one, i. e. it is the same for when generating URL's from STRIP_BODY_URI_SAVE and STRIP_BODY_URI_LINK options for one body part (but different for another body part).

`$RAND256$` - random hexadecimal number between 00 and FF presented as string. This number is stripping-session-wide one, i. e. it is the same for when generating URL's from STRIP_BODY_URI_SAVE and STRIP_BODY_URI_LINK options for one body part (but different for another body part).

`$CONTENTID$` - content ID of body part as extracted from headers or, if absent, generated at runtime. This is equal to Content-ID field putted with External Body method to headers of link.

`$CONTENTTYPE$` - MIME primary content type (i. e. image from image/gif).

`$CONTENTSUBTYPE$` - MIME subtype (i. e. gif from image/gif)

`$FILENAME$` - name of body part (filename for attachment) as proposed by message headers or, if absent, generated at runtime. This macro should always be present as last component of both STRIP_BODY_URI_SAVE and STRIP_BODY_URI_LINK. Also it should not be combined with other strings and macros (i. e. file:///dir/file-$FILENAME$) because of the limitation of algorithm for generating unique file names and further substitution it to link.

`$SIZE$` - size in bytes of decoded body part.

`$LINK$` - special macro available only within text and html templates. Macro is substituted with ready URL obtained by substituting other macros into STRIP_BODY_URI_LINK. This macro MUST be used in templates instead of trying to form link from other macros as for STRIP_BODY_URI_SAVE and STRIP_BODY_URI_LINK options.

Time macros:

`$TIMESTAMP$, $YEAR$, $MONTH$, $DAY$,$HOUR$, $MINUTE$ ,$SECOND$`

Macro values are validated before substitution to not contain potentially dangerous characters and character combinations that might come from message as following:

- each character '/' is substituted with 'x';
- standalone strings “.” and “..” are changed to “x” and “xx”;

- all characters that must be encoded (including control characters) for URL are encoded when macro value is substituted into link URL;

4 Advanced

The advanced section of configuration contains many important elements of MPP regarding message processing

4.1 MPP Daemon

The core component of MPP is mppd, the daemon process of MPP. This section configures critical elements of daemon operations.

4.1.1 Log Level

Info - s the recommended log level and minimum required logging level for parser scripts to works

Debug – Very Verbose

Debug Data – Extremely Verbose

Critical – Crashes and Sever Warnings

Error – Virus infections, errors and warnings

4.1.2 Advanced Logging Options

By default MPP logs are stored in /var/log/MPP, one file per date. MPP logs can be broken up by size, time and other variables and can also be sent to a syslog facility.

4.1.2.1 Type (or engine) to use for logging

Dirlog – Default

Syslog – Syslog mail facility is used

> **NOTE:** All of the following options only apply to dirlog style logging, not syslog.

4.1.2.2 Base directory to put log files to

Self explanatory, /var/log/MPP is default

4.1.2.3 Template to generate log file name and location

Default log name is $YEAR$$MONTH$$DAY$.log,

Variables for custom names and locations:

$YEAR$ - periodical macro that stands for current year (four-digit integer).

$MONTH$ - periodical macro that stands for current month (two-digit integer).

DAY - periodical macro that stands for current day (two-digit integer).

$HOUR$ - periodical macro that stands for current hour (two-digit integer).

$MINUTE$ - periodical macro that stands for current minute (two-digit integer).

$SECOND$ - periodical macro that stands for current second (two-digit integer).

S_YEAR - macro that stands for log starting year (four-digit integer).

S_MONTH - macro that stands for log starting month (four-digit integer).

S_DAY - macro that stands for log starting day (four-digit integer).

S_HOUR - macro that stands for log starting hour (four-digit integer).

S_MINUTE - macro that stands for log starting minute (four-digit integer).

S_SECOND - macro that stands for log starting second (four-digit integer).

Examples:

/var/log/$YEAR$/$MONTH$/DAY.log

Log for August 1, 2008 would be stored in

/var/log/MPP/2008/08/01.log

4.1.2.4 Time for log file to be closed and new one started

By default one log is generated per day. When a time value, in minutes, is entered in this option MPP will create a new log file every x minutes. MPP adds a .1 to increment log files for a single day.

For example if new logs were created every 12 hours using the macro from the above example, they would appear this way in the log directory.

/var/log/MPP/2008/08/01.log

/var/log/MPP/2008/08/01.log.1

4.1.2.5 Size limit for log file to be closed and new one started

Specify size in Kilobytes when MPP should create a new log file.

4.1.2.6 Whether to start new log when mppd is restarted

This option will create a new MPP log each time mppd is reloaded

4.1.3 Message Tracking

Message Tracking is a major feature of MPP that provides a compact record of every stage of message processing in a MySQL database. With message tracking you can derive detailed statistics about any message, find the result of any message that was scanned, search for message by remote relay server, message ID or many other elements.

Message Tracking is enabled by default on the MPP Virtual Appliance and it is highly recommended for all installations that require detailed statistics about message processing.

4.2 MPPD Configuration File

This section allows you to view or edit the XML configuration file, mppd.conf.xml, restore a default configuration or import a configuration from a TFTP server.

4.3 MPP Daemon Options

4.3.1 Number of threads

This value refers to the number of threads that mppd has available to scan messages. In general two scanning threads are sufficient, however, busy sites will need to increase this value. Thread counts should not exceed 20 unless there is a very good reason to do so.

It is recommended to look at the thread utilization statistics in Status->Monitor and Report to look for an even distribution of messages amongst scanning threads. Too many scanning threads will degrade performance while too few will do the same. Slower plug-in modules such as SpamAssassin will require more processing threads.

These scanning threads are distinct from threads that MPP uses to process SMTP or LMTP messages such as in the Postfix Policy Server or Content Filter implementation.

4.3.2 Advanced Options

4.3.2.1 Start Daemons on Startup

This configuration option controls if MPP should start the daemons of plug-in modules on start-up. F-PROT, Clamd, Spamd and Commtouch can all be started by mppd.

MPP will start spamd with user nobody and all SpamAssassin prefs, databases and bayse filters will be associated with this user. If you prefer to start spamd under a different user than disable this feature and start spamd as you normally would.

4.3.2.2 Unix owner and group

Changing the user and group of MPP is an involved process as many ancillary changes must be accounted for. Please consult support if you require changing the default ownership of mppd.

4.3.2.3 Location for MPP magic number

The 'magic number' is a value that mppd uses to mark messages that should not be scanned for some reason, such as alerts, forwarded messages, etc. The magic number is recomputed when mppd is reloaded.

If you are using MySQL spam quarantine then it is advisable to store the magic value in the same database as the spam quarantine. This way, when messages are released from quarantine the correct value will be tagged in the message headers.

If you have multiple instances of MPP only one should store it's magic in the spam quarantine database – the instance of MPP that will scan a message that is released from quarantine.

4.3.3 Enable Internal Queue

The MPP internal queue is used when messages are handed to MPP for processing by SMTP and other processes. With the internal queue enabled messages are written to a disk queue before heavy processing if resources are not available for immediate processing.

4.4 Alerts

This section is a consolidated configuration center for all MPP alerts. Alert sections and conditions are self-explanatory in GUI. There is one template available for sender, receiver and admin for all alert messages, with the exception of threshold alerts.

The macros available for warning templates are:

%FROM% will be replaced with the e-mail address that the warning is being sent to.

%TO% and %BCC% Will be replaced with the sender of the infected message if this is the template for a sender, or the receivers of the infected message if this is the receiver's template, or "postmaster" if this is the admin's template.

%SUBJECT% will be replaced with some program defined subject text.

%REPORT% will be replaced with scanning report and actions that have been taken.

%REPORT_HEADERS% will be replaced with full mail headers of the original message.

4.5 Scanners

4.5.1 Scanners to Use

Set the order of scanners in this section. Scanners appear in drop down menus if the components are installed on the system.

4.5.2 Message Conditions

This section is to configure how mppd reacts to various conditions. These options are also available in Services->Utility.

4.5.3 Error Conditions

This section controls how MPP reacts to various scanning error conditions. The action of Defer is only available with Postfix.

> **NOTE:** Pay close attention to the action configured for 'when a scanning error occurs'. Scanning errors can occur for many reasons but the most common reasons are SpamAssassin timeouts and out of date Sophos monthly virus definitions (804021e error code). If the error action is set to quarantine then all messages will be quarantined on the event of a scanner error.
>
> **It is recommended to enable warning messages for admins for scanning errors and set this action to defer or pass.**

4.5.4 Configure Scan Engine

4.5.4.1 Max file size (KB) for spam engine(s)

Use this variable to fine tune how much data sent to a spam engine. Most spam engines only need the first few bytes of messages. Generally this variable does not need to be changed.

4.5.4.2 Scan inside archives and compressed files

This option applies to virus scanners only.

4.5.4.3 Maximum file size to scan (in MB)

Self-explanatory

4.5.4.4 Maximum recursion level

This setting controls how many MIME recursion levels MPP will descend into a message. This is meant to avoid certain types of DoS attacks.

4.5.5 Scan engine socket

This is an important section of MPP configuration and in most cases it need not be touched. However some spamd, f-protd and clamd all support TCP sockets, hence, they can be located on remote servers.

MPP has a round-robin protocol for load balancing amongst multiple instances of spamd, either locally or remote. To add a new instance of SpamAssassin click *add new* to create a new socket definition.

Examples:

inet:783@localhost

inet:783@192.168.100.21

4.5.6 Scan engine timeouts

Use this section to fine tune scan engine timeouts. This setting generally only needs to be modified for SpamAssassin, which is notoriously slow. A read and write timeout of 30 seconds (30000 milliseconds) is required to keep SpamAssassin from timing out on message scans.

> **NOTE:** SpamAssassin timeouts are the most common scanning errors and they are fixed by adjusting this parameter to at least 30 seconds.

4.5.7 Scanner Optimizations

Optimizations are settings to control the serialization of scanners. There are a few factors that control how serial scanners are handled. Scanners are processed in the order of configuration. It is generally

desirable to use spam scanners first. Since the majority of email in today's environment is spam why waste system resources on scanning spam for viruses? Of course there are many arguments why to place virus scanning first, but we recommend spam or content scanning first.

4.5.7.1 Virus Scanner Optimizations

Use this to configure the behavior of subsequent scanners after a virus is found. Options are to skip spam scanners or content scanners

4.5.7.2 Spam Scanner Optimizations

Use this option to configure how subsequent scanners behave after a spam is identified.

> **NOTE:** MPP Custom Spam Scoring allows you to create aggregate scores from spam scanner results, RBL tests, SPF tests and content filter expressions.

4.6 Thresholds

Threshold or Auto Blacklists are used to automatically detect senders that are out of compliance with policies that you define. Thresholds track message counts and spam counts over a sample period. If a sender violates a message rate then you can take action against them – either warn an admin or block them for a specified period of time.

The values of a threshold are - Message Count, Clean Count, Spam Count, Sample Time and Time in Cache

Since many botnet attacks are from many sources, each sending the same message, MPP can group SMTP hosts by subject. So if one SMTP sender violates a threshold, then all senders that send with the same subject will be blocked.

MPP will also treat content violations as 'spam' when counting thresholds.

Many people use MPP thresholds to find outbound spammers or heavy senders that may lead to their email servers becoming blacklisted. For applications such as these define a threshold with zero spam count, a high message rate and a short sample period. Any sender who is sending too many messages will now be found quickly.

Valid threshold actions are block or alert. When a sender is blocked they are placed on a temporary Client-Host blacklist. If Client Host Blacklists (from Services->Connection) are configured to use MySQL then blocked hosts will be stored in a database. If you have multiple MPP instances, they can all share the same database to block the same hosts. If a database is not defined for Client-Host blacklists then auto-blacklist entries are stored in RAM.

If the Postfix policy server is in use the hosts are immediately disconnected.

Threshold alerts are unique and allow you to define the senders and receivers. On a busy mail system there will be many warning messages and generally the warnings are sent to automatically monitored mailboxes as they can very rapidly. Threshold alert text cannot be customized.

Thresholds have their own white lists to exempt hosts from threshold checks. Whitelists are entered as CIDR IP addresses, such as 192.168.100.0/24 or as host names.

4.7 Message Stores

The Message Stores section is a consolidated location to define email archives and quarantine locations.

4.8 Message Tracking

Message tracking is an important feature for MPP. Like thresholds, reject templates and per-user WBL's it is another feature for a big deal that never materialized ☺. Message Tracking keeps a very compact record of each message that MPP scans in a MySQL database.

Message tracking is designed to scale for very large message counts and care has been taken to store the minimal amount of information. There are many configurable options to custom tailor how MPP stores tracking data about different facilities of MPP.

Message tracking makes it possible to query message status by message id, scan state, remote relay server, domain, email address and many more variables. Message tracking also gives detailed statistics that can be grouped per-user, per-domain, per-engine, by scan state, by scan result and more.

For each MPP scanning stage and result MPP can store no information, basic or detailed statistics. If Message tracking is enabled MPP will store detailed information about parsed, quarantined, quarantine_failed, archived, archive_failed, relayed, relay_failed,queued, queue_failed and forwarded.

Other facilities that can be enabled are as follows in the common section of mppd.conf.xml:

TRACK_MAIL_TRANSFER_ACL

e.g. < TRACK_MAIL_TRANSFER_ACL>detailed</ TRACK_MAIL_TRANSFER_ACL>

```
TRACK_MAIL_TRANSFER_ARCHIVED
 TRACK_MAIL_TRANSFER_ARCHIVE_FAILED
 TRACK_MAIL_TRANSFER_CLIENT_BL
 TRACK_MAIL_TRANSFER_CLIENT_DF
 TRACK_MAIL_TRANSFER_CLIENT_DNS_FAILED
 TRACK_MAIL_TRANSFER_CLIENT_WL
 TRACK_MAIL_TRANSFER_DELETED
 TRACK_MAIL_TRANSFER_DELETION_FAILED
 TRACK_MAIL_TRANSFER_DISINFECTED
 TRACK_MAIL_TRANSFER_DISINFECTION_FAILED
 TRACK_MAIL_TRANSFER_EMPTY_MESSAGE
 TRACK_MAIL_TRANSFER_ENCRYPTED
 TRACK_MAIL_TRANSFER_ERROR
 TRACK_MAIL_TRANSFER_FORWARDED
 TRACK_MAIL_TRANSFER_GROUP_BL
 TRACK_MAIL_TRANSFER_GROUP_WL
 TRACK_MAIL_TRANSFER_HARASS
 TRACK_MAIL_TRANSFER_INFECTED
 TRACK_MAIL_TRANSFER_LDAP_FAILED
 TRACK_MAIL_TRANSFER_MAGIC_FOUND
 TRACK_MAIL_TRANSFER_MALFORMED
 TRACK_MAIL_TRANSFER_MAX_FILE_SIZE
 TRACK_MAIL_TRANSFER_MAX_RECURSION
 TRACK_MAIL_TRANSFER_MPP_SPAM_SCORED
 TRACK_MAIL_TRANSFER_NO_RECIPIENTS
 TRACK_MAIL_TRANSFER_PARSED
 TRACK_MAIL_TRANSFER_QUARANTINED
 TRACK_MAIL_TRANSFER_QUARANTINE_FAILED
 TRACK_MAIL_TRANSFER_QUEUED
 TRACK_MAIL_TRANSFER_QUEUE_FAILED
 TRACK_MAIL_TRANSFER_RBL
 TRACK_MAIL_TRANSFER_RBL_FAILED
 TRACK_MAIL_TRANSFER_RELAYED
 TRACK_MAIL_TRANSFER_RELAY_FAILED
 TRACK_MAIL_TRANSFER_SPAM
 TRACK_MAIL_TRANSFER_SPAM_TRAPPED
 TRACK_MAIL_TRANSFER_SPF
```

```
TRACK_MAIL_TRANSFER_USER_BL
TRACK_MAIL_TRANSFER_USER_WL
```

4.8.1 Per-Engine Reports

Message tracking reports are very accurate accounts of the activity of your email system. Each entry of a message-tracking event can be expanded to show the result of every recorded processing stage.

Valid search criteria in the advanced section are email, domain, message ID, remote relay server, date, time, message events and results. Future releases will include more reporting options. A single message can have multiple events, such as discard and quarantine.

4.8.2 Message Events

Spam, infected, disinfected, mpp_spam_score, parsed, disinfection_failed, deleted, deletion_failed, harass, rbl, rbl_Failed, unauthorized, encrypted, malformed, max)recursion, max_file_size, quarantined, quarantined_failed, archived, archive_failed, acl, aclent_wl, client_bl, client_df (deffered), client_dns_failed (could not resolve IP for hostname), group_wl, group_bl, user_bl, user_wl, error, relayed, relay_failed, queued, queue_failed, no_recipients, empty_message, forwarded, ldap_failed, magic_found

Every message will have a parse event and most will have a relayed event.

4.8.3 Message results

Passed, discarded, deferred, rejected

4.9 Reject Templates

Reject templates are another one of countless MPP features designed to win one big deal or another. This feature is actually very cool and one of the truly unique features of MPP. On the surface customer reject templates do what they say they do – they allow you to create custom rejection notices that contain detail about why message was rejected (550 SMTP Code). This is useful if you want to customize SMTP rejection (550) notices to tailor some business or technical need. However, Reject Templates can also be used an API to MPP, allowing MPP to function as a stand-alone SMTP scanner.

Reject notices are only useful for pre-queue features such as rejections from the Postfix policy server or Sendmail Milter interface. Sending post-queue rejection notices is very bad practice as you will be bouncing complete messages, thus creating potentially huge basckatter problems.

If you can't reject a message before the message is queued then don't use the reject action.

4.9.1 Reject Condition Templates

Templates are constructed in two parts, a mandatory part and text customized each of the following conditions:

- virus condition
- spam condition
- unauthorized condition
- error condition
- maximum recursion condition
- maximum size condition
- acl condition
- encrypted condition
- malformed condition

- blacklist condition
- RBL condition

4.9.2 Reject Notice Construction

The mandatory part of a rejection notice will always be present, while one or more conditional parts will follow after, if the corresponding condition is true. Conditional parts are separated with spaces.

For the SMTP protocol, there is one response for all recipients. If the client initially specified multiple recipients, and they were processed differently, and different rejection text should be reported for at least one of the recipients, then rejection text will be combined as follows:

[text for recipient 1][text for recipient 2][text for recipient n]

Text for each recipient will be in "[]" brackets - Otherwise, there will be single text for all recipients without brackets.

The LMTP protocol provides one response for each recipient, so the described scheme is not used.

The following predefined macros can be used specified within mandatory or any of the conditional parts:

`%INTERNALID%`	Internal ID of the message, this is to search in MPP logs for transaction flow.
`%GROUP%`	MPP group that was used to scan the message.
`%STATE%`	Substituted with a comma-separated list of scan states. Here's the full list: virus,spam,harass,unauth,error,max_recursion,max_size,acl,encrypted,malformed,bl, rbl.
`%VIRUSLIST%`	Substituted with a comma-separated list of found viruses.
`%SPAMSCORE%`	Spam score for the message as returned by a scanner.
`%SPAMLEVEL%`	Spam level for the message. This could be "low", "medium", "high", or empty string.

4.10 Reject Templates – the MPP API

Reject templates can also be thought of as a type of API into MPP. When used in this fashion all actions of MPP are set to reject and a program will parse MPP rejection notices to take appropriate actions. In this way MPP can be a scanning brick that accepts messages on SMTP and hands back response codes.

Since MPP will use its entire arsenal of plug-ins, embedded functions and features MPP any script or entity that can pass a message to MPP via SMTP can benefit from our functionality.

In order to use MPP in this function set <on_clean> to reject, this can be done in the GUI in Advanced->Policy Engine -> Advanced->Action for Messages. Also set every other action to reject – all spam thresholds, on virus, acl violation, rbl, acl, etc.

By setting all actions to reject your application will get a custom message back from MPP that uses your template, telling the application the result of MPP processing. So the general flow is:

Your app → SMTP to MPP → MPP Scans → MPP sends reject report → Your app reacts

For more details on using MPP in this fashion please send an email to support@messagepartners.com

4.11 Policy Engine

The Policy Engine is central to MPP and provides the capability to create distinct configuration sets that can be applied to groups of users. Groups can be identified by email address, domain, CIDR IP address and direction. Membership information can be stored in the MPP configuration file, text files or can be added as an attribute to any standard LDAP directory. Please see the appendix for details of using LDAP for policy storage.

4.11.1 How can I use the policy engine?

In most cases the uses of the policy engine are simple – create different rules for inbound and outbound processing, archive email for a few people, monitor the emall of a few, etc. However, since MPP can support many thousands of policies and policy membership can be stored in a subscriber directory the policy engine can lead to elaborate configurations.

It is very common for service providers to use MPP to query their subscriber directory to see if a user has subscribed for a service such as an archival. It is also common to classify processing options by IP address of trusted SMTP servers.

The options of the MPP policy engine are limitless and it is really one of the most interesting abilities of MPP.

MPP takes great care to preserve the integrity of multi-recipient email. If an email goes to two users or domain in different policies then both policies will have their preferences respected.

As each email message gets analyzed during processing pipeline, MPP must determine the appropriate policy for the message.

4.11.1.1 Policy Match Logic

Data sources are searched in the order described below. If the configuration parameters have multiple values, they are searched in order.

1. Search through merged list of IP address entries from static LDAP, global text files and local text files.
2. Search through dynamic LDAP
    i. Sender
    ii. Sender Domain
    iii. Receiver
    iv. Receiver Domain.
3. Search through merged list of entries from static LDAP, global text files and local text files for
    i. Sender
    ii. Sender domain
    iii. Receiver
    iv. Receiver Domain

4.11.2 Default Policy and Inherited Options

Email that does not have a specific policy match will be processed with the default policy group settings.

When creating new policies most settings are inherited from the default policy group s such as choice of scanners, spam actions, quarantine location and all others except those identified in the next section.

4.11.3 Non-inherited Configuration Options

```
GROUP_ADDRESSLIST_IN
GROUP_ADDRESSLIST_OUT
ACCESS_LIST_MEMBERS_ADDRESSLIST
ACCESS_LIST_MEMBERS_ADDRESSLIST_RECIPIENT
ACCESS_LIST_MEMBERS_ADDRESSLIST_SENDER
ACCESS_LIST_MEMBERS_ADDRESSLIST_BOTH
```

```
ACCESS_LIST_MEMBERS_URI
ACCESS_LIST_MEMBERS_URI_RECIPIENT
ACCESS_LIST_MEMBERS_URI_SENDER
ACCESS_LIST_MEMBERS_URI_BOTH
ACCESS_LIST_USE_RECIPIENT
ACCESS_LIST_USE_SENDER
ARCHIVE
ARCHIVE_VIRUSES
ARCHIVE_SPAM
ARCHIVE_HARASS
ARCHIVE_UNAUTHORIZED_ATTACHMENT
ARCHIVE_UNAUTHORIZED_CONTENT
ARCHIVE_UNAUTHORIZED_HEADER
ARCHIVE_MALFORMED
ARCHIVE_WITH_ENVELOPE
SPAM_WHITELIST
BLACKLIST
WBL_URI
CLIENT_HOST_WHITELIST
CLIENT_HOST_BLACKLIST
CLIENT_HOST_WBL_URI
RBL_SITES
SPAM_ACTION_URI
STRIP_BODY_NAME
STRIP_BODY_TYPE
STRIP_BODY_NAME_NOT
STRIP_BODY_TYPE_NOT
CLIENT_HOST_AUTOBLACKLIST_THRESHOLD
CLIENT_HOST_AUTOBLACKLIST_GROUPING
CLIENT_HOST_AUTOBLACKLIST_ACTION
SPAM_TRAPS_TEMPLATE
```

4.11.4 Policy Dropdown

The MPP GUI generates a list of all available policies and presents them as a drop down menu on the upper right of pages in the Services and Advanced sections. To configure options for a policy select the policy from the drop down on the upper right hand corner of the interface.

4.11.5 Creating New Policies

1) Name the policy by typing the name in the policy drop down, next to the ADD button. This is a special drop down that allows you to type new entries.
2) Create the matching criteria by typing in the match criteria for sender, recipient or both setting. Entries are separated by commas. mpp@messagepartners.com, mpp2@messageparnters.com, raeinternet.com.

   Valid entries are email address, domains or CIDR IP Address – mpp@messagepartners.com, messpartners.com, 209.212.12.0/24
3) Save and restart to have the policy take effect.
   a. If you require dynamic provisioning of policies use LDAP for policy storage.

4.11.6 Policy Actions

The default action for an MPP policy is to pass clean messages. However, there are interesting possibilities with other actions of MPP policies. If you expand the advanced section of this page you can change the policy page you can set the default action for a group.

Scan Default action, messages are scanned

Pass Messages are passed and not scanned, analogous to Client Host White List

Reject	Messages that match this group are rejected
Discard	Messages that match this group are silently discarded
Forward	Messages that match this group are forwarded to the address specified

4.11.6.1 Message Forwarding

If message forwarding is used it is possible to deliver the message after forward or discard.

> The ‘action for forwarded email’ is valid for all email with action of ‘forward’. Even if the action for a group is scan, this option is valid for forward action for content filtering, ACL’s and other places where forward is a valid option.

4.11.7 LDAP

MPP can consult and LDAP directory to find group membership. An entry has an object class (raeMPP) that has attributes raeMPPGroupNameRecipient and raeMPPGroupSender. MPP looks for the Email address LDAP attribute and then for the group name.

Please see the appendix for a complete explanation of LDAP attributes.

5 System

The system section of the MPP GUI controls system wide parameters and configuration parameters of the GUI itself.

5.1 SMTP Relay

MPP is very commonly used on SMTP relays and this section allows you to configure how MPP relays mail. This section is only available when MPP is installed with the Postfix MTA. If enabled, by selecting *'Allow MPP to control relay domains and transports'.* By selecting this option MPP will modify the Postfix configuration files, saving originals, to use relay and transport files that the MPP GUI can control.

5.1.1 Adding a Relay Domain

Type the name of the domain, such as messagepartners.com, the IP address of the real email server, 192.168.100.3 for example and decide if the mx record for the domain should be checked. For private networks, i.e. your own email server behind your firewall, don't check the mx record. If you are forwarding for a domain that may move their email server and mx then check this, but generally this should be left unchecked.

5.2 Database Monitor

This section displays red, green or yellow to show if a database is online, non-functional or has in an unknown state, such as extra tables (orange).

From this page you can show, drop, empty or dump (back up) tables.

5.3 Database Setup

This section is to configure, monitor and create MySQL databases for use with MPP. There is a lot of functionality with this section and a few key concepts that must be understood.

5.3.1 Default Database

The default database is stored in the MPP GUI configuration files, it is not extracted from MPP configurations. It is a good idea to define a default database that MPP monitors.

5.3.2 How Many Databases?

We recommend that spam quarantines and email archives use separate databases. While it is possible to share the database connections between all purposes it is best to keep the functions separate.

5.3.3 Database Functions for Spam Quarantine

The following MPP features should use the same database if spam quarantine is in use.

- Primary Quarantine
- White and Black Lists
- Per-User Spam Actions
- Recipient ACL
- Client-Host WBL
- Magic Location

5.3.4 Database Functions for Archive

To use MySQL for email archive only the archival feature needs to be set to use MySQL, however, it is strongly recommended to use the MPP virtual appliance for this purpose. If you opt to set up MPP email archival on your own it is mandatory to also install the Sphinx indexing engine, as per the appendix, to enable full-text searching.

5.3.5 Creating a Database

NOTE: Read this section carefully. If you have multiple policies be mindful of the policy selection.

To create a database for MPP to use follow these steps.

1) Create a blank database with your favorite MySQL tool and a user that can access the database. The database can be remote or local and multiple instances of MPP can share the same database. Here are the commands to create an empty database and a user with access rights from the localhost:

```
mysql> create database spam_quarantine;
Query OK, 1 row affected (0.00 sec)

mysql> grant all on spam_quarantine.* to 'mpp@localhost'
identified by 'mpp_password';
Query OK, 0 rows affected (0.02 sec)
Exit;
```

2) Define the settings of database in *Global MySQL Database Connection.*

3) Decide if this database should be the 'Global' database. The Global Database is an analog of MPP GUI that it uses as the default to monitor. After entering the settings select *Save and Create Tables.*

5.3.6 Applying a Database

To apply the database settings to places in the MPP configuration check the applicable boxes in the right column of the page. For spam quarantine you should apply the same database to quarantine, recipient ACL, White/Black List, Per-User Spam Settings, Magic Location and Client-Host WBL.

If you have configured multiple policy groups it is possible to copy the database permissions to other policy groups by selecting the functions of database (i.e. wbl, spam settings, etc.) and the name of the policy that you want to *COPY TO:*.

5.4 MTA Integration

This section controls how MPP interoperates with a Mail Transfer Agent, or MTA. In most cases it is not necessary to ever change the parameters in this section as the default settings will work in most cases.

Most parameters are self-explanatory, however, the options in the Postfix section require some explanation because this represents the primary SMTP interface for MPP.

5.4.1 Generic SMTP Interface,

The Postfix interface is a generic SMTP interface that any application can use to submit messages to MPP. Any SMTP application can submit messages to MPP via SMTP and the socket address may be configured

5.4.2 Postfix Options

LMTP is the best protocol choice to use with Postfix because it provides per-user status codes. So if you are using Postfix use LMTP.

5.4.3 ESMTP Extensions for Quarantine Transport

MPP supports a hierarchical storage model that transports quarantine or archive email via ESMTP. To configure a listener for this model enable the *mailstore* selection from the *List of extensions to use for SMTP/LMTP protocol* option.

In order to create a 'listener' for this traffic configure a remote instance of MPP to recognize the ESMTP extensions in System -> MTA Integration set MTA to Postfix and *List of extensions to use for SMTP/LMTP protocol* to mailstore.

5.4.4 SMTP Listening Sockets

By default MPP listens on 10025, however this can be changed in *the Socket type and address of the for incoming connections* field. This is useful if you have multiple Postfix content filters or need MPP to listen on some other socket for an external application.

5.5 MPP GUI

The options in MPP GUI section are largely self-explanatory and need not be changed in virtually all cases. The commonly changed fields are outlined here:

5.5.1 Log Files Directory

Change this if necessary.

5.5.2 MPP parser time period

By default the parser script is run every 20 minutes, change this value if you have very large log files.

5.5.3 MPP GUI Certificate Files

This section allows you to upload you own certificates if SSL was enabled during the installation of MPP Server. It is not necessary to install your own cert files to use SSL with MPP Manager.

5.6 Postfix Policy Server

The Policy Server is a very important piece of the MPP filtering architecture for its pre-queue filtering possibilities. Since MPP has both a Postfix Content Filter and a Policy Server, and they share information, MPP is the only integrated pre and post queue-filtering tool for Postfix that we are aware of.

MPP processes spam white and black lists, spam traps, access control lists, real-time black hole lists and

5.6.1 Policy Server Architectures

The policy server can be used in 3 different architectures:

5.6.1.1 Policy Server and Content Filter Coexist

In this model, the default, it is assumed that the Content Filter and Policy Server co-exist on the same server. This is fine for the majority of sites.

5.6.1.2 Policy Server Only

This model is suitable for sites where there are high volumes of SMTP traffic that needs to be processed by connection services such as spam traps, black lists, client-host black lists, access-control lists and real time black hole lists. In this model MPP is only a policy server and there is no content filter configured. In this model MPP cannot support per-recipient actions.

5.6.1.3 Remote Policy Server

In this model the policy server is on a dedicated server that is only a policy server. Postfix is not operational on this server, only MPP is accepting policy connections and responding to queries.

5.6.2 Policy Server Configuration

In any usage case configuration is basically the same. The MPP configure.pl script will setup the Policy Server within MPP and the Postfix configuration files. If you decide after installation to run the policy server enabling it in the GUI will configure all components necessary.

When MPP policy server is NOT co-existent with the content filter it is important to leave the filter string as blank. By default the filter string is mppscan, this should be set to blank when the policy server and content filter do not co-exist.

5.6.3 Greylisting

Greylisting is only active if the Postfix Policy Server is enabled. If the Policy Server is enabled configure Greylisting in Services->Connections.

5.6.4 Pre and Post Queue Processing with Policy Server

MPP has a unique capability to intelligently process pre or post-queue with Postfix using the following logic:

At first RCPT command from client the Policy Server receives a request from Postfix with IP, sender and recipient info. MPP then checks whether IP is in Client Host Whitelist. If yes then the message is NOT routed to post-queue MPP scanning at all and is only processed by Postfix. If the message is not on the CHWL then MPP checks whether IP is in Client Host Blacklist. If yes then recipient is rejected with a 421 code, which causes postfix to disconnect from client immediately. No further processing is performed.

If the message is not on any CHBL then the MPP policy-engine processes the message according to sender and recipient information. If the group matched has an action of "pass" then the message is NOT routed to post-queue MPP scanning. If it is "reject" recipient is rejected without disconnecting. If it is "discard" message is discarded.

This scheme is the same for ACL's and WBL's except that if quarantine is required or there is a spam whitelist hit, the message is routed to post-queue processing. If blacklist violation action is to "mark header" then it is marked and is passed and NOT routed to post-queue scanning.

If the message is not black- or whitelisted per-group then RBL's are queried. Here message can be discarded, rejected or left for further processing depending on corresponding RBL options.

A special case exists for messages with multiple recipients. Some actions cannot be applied on per-user basis at pre-queue stage due to postfix limitations. Those actions include pass, discard and markheader. In this case, if any of these actions appear for one recipient, and there is other non-rejected recipient, then decision is postponed until post-queue processing. For example if for one recipient group action is "pass" and for other it is "scan" then message is passed to post-queue processing for BOTH recipients because "pass" cannot be applied on per-user basis at pre-queue.

6 Design Concepts

This section will provide some basic design concepts and hints for success with MPP.

6.1 Archive Considerations

MPP provides a complete, standards based solution for email archives. MPP archives email from a number of sources including other instances MPP, Postfix always_bcc and MS Exchange 2003 and 2007 journal reports. MPP imports email from MIME encoded files and IMAP message stores.

6.1.1 Email Archival Formats

MPP stores archived email in a variety of formats - directly into a MySQL database, in MIME encoded files, in maildir formats or in a hybrid format (metadata) where file locations are stored in a MySQL database and the actual emails are stored as files on the file system. MPP has macros available to scale storage of files on the file system. These macros allow MPP to create random numbers of directories, 16 or 256, which can be nested (/usr/local/MPP/archives%rnd256%/%rnd256%) to avoid file system limitations on the number of files that may be stored in a directory.

When the same email is archived for many users only a single copy is stored in a MySQL database, however, when files are used, a copied of the email is stored for each user.

It is highly recommended to use full database storage of email if full-text searching is a requirement.

6.1.2 Database Replication

It is highly recommended to use database replication with multiple slave databases. MPP is cluster aware, as such the archive review application supports separate read and write channels. It is recommended for reads to be performed on slaves while mppd writes to a database master.

6.2 Archival Architectures

MPP archival systems can be complex with multiple archival agents speaking to an aggregator, or they can be simple, with a single archival agent and a single database.

6.2.1 Simple Archival Architecture

In this model a single instance of MPP is archiving to one message store.

6.2.2 Hierarchical Archive Architecture

In a hierarchical setup there is a central MPP instance that writes to the message store and accepts read requests from remote instances of MPP that archive with the smtp:// method. MPP has special EMSTP extensions for this architecture that are described in the MTA Integration section above. It is highly recommended to consider this architecture for MPP archive applications. Even if you have a single server it is possible to have two instances of MPP running to support this architecture.

6.2.2.1 Archive Agent

The archive agent is an instance of MPP running in conjunction with an email server or SMTP relay. If there is a single archival agent it can connect directly to the MySQL database

6.2.2.2 Aggregator

If there are multiple archive agents MPP can use ESMTP to transport archived email to an instance of MPP that functions as an aggregator of remote archival instances. The aggregator will hand traffic to a queued MPP process that stores the email into a database, file system or hybrid message store.

6.2.2.3 Email Store - Database or File System

MPP stores email in 3 formats; i) email and message info in a MySQL database ii) email and message info on a file system or iii) email in the file system and message info, aka meta data, in a MySQL database

6.3 Email Filtering

6.3.1 Small Site Email Filtering

MPP is perfect for email filtering. It is highly recommended to use our Virtual Appliance as everything is pre-configured and setup.

6.3.1.1 Mac OS X

If your small site uses MacOS X we have a bundle installer that will install the complete MPP kit and all dependant libraries. This package will save a lot of time for you.

6.3.2 Large Scale Email Filtering

MPP is especially well suited to large-scale email filtering applications. Here are a few design tips to make you successful with MPP in large environments.

- Use two levels of filtering servers, one to handle SMTP transactions and one to perform content filtering. The SMTP filters should use the Postfix MTA since MPP can make pre-queue decisions Postfix. The SMTP transaction servers should perform RBL checks, spamtraps, greylisting if applicable, access control lists and process client-host blacklists. All of these services should be processed with MPP's policy server.
- Use shared SQL servers for all front-end's. These SQL servers should store client-host blacklists, access-control lists, and per-user black lists, Greylist.
- Use hierarchical quarantine storage to reduce direct connections between mppd processing agents and message stores. This is achieved with remote mppd instances storing messages with smtp:// method and a central mppd listening on an smtp socket defined in Advanced -> MTA Settings with Postfix configured and SMTP Extension set to messagestore. It is important to note that there need not be an instance of Postfix running on the central mppd as mppd provides its own smtp instance for this purpose.

6.4 MySQL Considerations

6.4.1 Commercial Licenses

If you are using archive it is highly recommended to have a commercial license of MySQL server to have access to their management tools. Don't reinvent the wheel with things like replication, backups, etc. as these are all solved problems in the MySQL world. Leverage the community.

6.4.2 Replication

It is highly recommended to use MySQL replication for archive and large spam quarantine applications. Replication doesn't help much for spam settings databases, but it will make a huge difference for archive and spam quarantine. MPP Manager is replication aware and as such supports separate read and write

channels. In replication architecture mppd should write to the master and MPP Manager should be configured to read from slaves. Simultaneous reads and writes on the same database, especially if it is too big to fit in memory will clobber performance and replication solves this problem.

6.4.3 Max Packet Size and Max Rows

Make sure to set the max rows to a large number if you are using archival. We have set this for you in the virtual appliance. Set the max packet size to the largest attachment size that you want to store.

6.4.4 Centralized Databases

If you have multiple mppd filtering servers use a centralized database for spam settings, white and black lists, client-host blacklists and access-control lists. All instances should share the same database for maximum efficiency.

6.4.5 Message Tracking

Message tracking is a powerful feature for MPP and it should have its own dedicated database.

6.4.6 Shared Databases

Do not share the same database for spam quarantine, archive and message tracking. It is ok to share the same DB for spam quarantine, acl, wbl, spam settings and client-host WBL.

7 Appendix

This section provides some technical details for your information. Perl Compatible Regular Expression Basics

> **NOTE:** PCRE is the original regular expression syntax supported by MPP; however, in version 4 we have significantly expanded our regular expression capabilities. This guide is provided for legacy compatibility. See the next section, Boorex Expressions, to learn about the expanded regular expression capabilities of MPP.

Regular expressions are defined in file(s), one per line. Lines beginning with # are ignored at reading/parsing.

The format of regular expression is:

```
/regular expression/{flags}
```

{flags} are optional and they can be combined. Available flags are:

`i` - Search case insensitive.

`m` - Match on multiple lines, thus ^ and $ are interpreted as the start and end of the entire string, not of a single line.

`s` - A dot in an expression matches newlines too, which is normally not the case.

Examples for flags:

`/regular expression/` - search case sensitive

`/regular expression/i` - search case insensitive

`/regular expression/im` - search case insensitive on multiples lines of the string, with ^ and $ matching begging and end of string

`/regular expression/is` - search case insensitive, having dot (.) matching newlines too

Important: When searching with regular expressions through headers or body, the headers or body are searched as single string, not line by line. Thus using of "s" or "m" flags is useful.

Regular expressions Meta characters:

`\` Quote the next meta character

`^` Match the beginning of the line

`.` Match any character (except newline)

`$` Match the end of the line (or before newline at the end)

`|` Alternation

`()` Grouping

`[]` Character class

Regular expressions standard quantifiers:

`*` Match 0 or more times

`+` Match 1 or more times

`?` Match 1 or 0 times

`{n}` Match exactly n times

`{n,}` Match at least n times

`{n,m}` Match at least n but not more than m times

Examples for header regular expressions:

`/From:.*ebay.com.*/is` - search case insensitive for From: header with anyone@ebay.com as sender

`/From:.*ebay.com.*Return-Path:.*seb\@ns2\.shanghai\.fr.*|Return-Path:.*seb\@ns2\.shanghai\.fr.*From:.*ebay.com.*/is` - search case insensitive for anyone@ebay.com and Return-Path: seb@ns2\.shanghai.fr considering that From: and Return-Path: might occur in different order.

Note: Using `|` enables internal regexp alternation. `reg1 | reg 2` - search reg1 or reg2

Examples for attachment names regular expressions:

`/(.*\.pif)|(.*\.scr)/i` - search case insensitive for PIF or SCR extensions in attachment name

Examples for content regular expressions:

`/secret/i` - search for "secret" word case insensitive

7.1 Boorex Engine Guide

MPP4 introduces an entirely new content filter engine with greatly expanded capabilities for construction of regular expressions and sharing of expressions amongst policy groups. This section contains brief description of the features from usage point of view. Entities referenced in this section are described in “Structural View” section.

The Boorex Engine is not yet fully implemented in the MPP GUI thus more granular detail is provided for editing the mppd.conf.xml file directly. **This is for advanced users only.**

7.1.1 Introduction

Boorex Engine incorporates a list of pattern expressions that are matched against input text with “first-match” principle and returns an ID of matched expression. It is based on Boost Regex library [1] and supports all expression formats that this library does. The engine can be used by other MPP functionality (Content Filter, Spam Traps, etc.) whenever pattern matching is required and supported.

7.1.2 Use-case View

This section contains brief description of the features from usage point of view. Entities referenced in this section are described in “Structural View” section.

7.1.2.1 Define Boorex Engine

Administrator defines an instance of Boorex Engine in mppd.conf.xml file under

/mppd/engines node with node name <boorex>. ID of the engine is defined with “id” attribute of <boorex> node. Multiple Boorex Engines with different ID's can be defined. Boorex Node Structure is described further in “Structural View” section

7.1.2.2 Use Boorex Engine

Boorex Engine can be used by specifying its ID in any context that supports this. Engine that is not used is not actually loaded.

7.1.3 Structural View

7.1.3.1 Boorex Node Structure

Boorex node defines configuration of Boorex Engine. The following example is intended to provide quick inside into its structure:

```
<mppd>
<engines>
    <boorex id=”engine_id”>
       <encoding>ASCII</encoding>
       <defaults id=”expression_id” options=”perl,match_perl”>global</defaults>
       <inline id=”expression_id2” options=”perl,match_perl”>\.gif$</options>
        <i id=”expression_id3” o=”perl,match_perl,match_single_line”>\.pdf$</i>
       <file id=”expression_id4” options=”perl,match_perl”> patterns.txt</file>
       <f id=”expression_id5 o=”perl,match_perl”>/usr/local/MPP/patternts2.txt</f>
    </boorex>
 </engines>
</mppd>
```

All options and attributes are optional. If any is omitted then default value is used. Minimum engine specification looks can look like this:

```
<mppd>
 <engines>
    <boorex><i>\.gif$</i></boorex>
 </engines>
</mppd>
```

Specifies: Boorex Engine.

Attributes: id

Specifies: Engine ID; must be unique witching <engines>.

Value: string

Default: “boorex” (i. e. name of the node)

Value: breaks further on sub-nodes.

Specifies: Internal encoding to be used by this engine; if you are going to work only with ASCII characters you should use 'ASCII'; if you are going to work with international characters then you should use 'Unicode' however this will require about 4x more RAM resources. In any case

implementation expects that expressions are specified using UTF8.

Value: ASCII | Unicode

Default: ASCII

Specifies: Default values for expression attributes and the way defaults are applied; "global" defaults means that defaults are defined once and remains the same during loading of all expressions; "global" defaults should be fine for most cases however for more flexibility "positional" defaults can be used which mean that if some attribute is defined by some expression defaults for this attribute is overwritten with this value.

Attributes:

id

Specifies: Expression ID (see further).

Value: string

Default: "match"

options

Specifies: list of comma-separated Expression Options (see further).

Value: string

Default: "perl, match_perl,match_not_null,match_single_line, match_not_dot_newline,match_any"

o

Short synonym for "options".

Value: (empty) | global | positional

Default: global

Specifies: Single pattern expression as UTF8 string; format of expression is defined "options" attribute (perl, posix, etc.).

Attributes:

id

Specifies: Expression ID.

Value: string

Default: (as defined by <defaults>)

options

Specifies: list of comma-separated Expression Options.

Value: string

Default: (as defined by <defaults>)

o

Short synonym to "options"

Value: string (expression)

Default: (n/a)

<i>

Short synonym to <inline>

<file>

Specifies: Path to file in Expression File Format (see further).

Attributes:

id

Specifies: Default Expression ID for expressions from the file.

Value: string

Default: (as defined by defaults)

options

Specifies: Default Expression Options for expressions from the file.

Value: string

Default: (as defined by defaults)

o

Short synonym for “options”

Value: string (path to file)

Default: (n/a)

<f>

Short synonym for <file>

7.1.3.2 Expression ID

Each pattern expression has ID assigned to it. This ID is returned to client code in case corresponding pattern is found in provided text. If expression doesn't have explicitly specified ID current default ID is assigned to it. Multiple expressions can have same ID in this case client code will not know what pattern from the group was found.

There are a few special ID's:

“mismatch” - this id is returned when no pattern where found;

“match” - this id is initial hardcoded default id for every pattern;

“*” - this id is equivalent to current default id; useful mainly for

Expression File Format (see further);

7.1.3.3 Expression Options

Each pattern expression has options associated with it. Options affect the way a pattern is interpreted and searched for. Options are the list of comma-separated strings. Each string denotes single option. For example: “perl,match_perl, match_not_null,match_single_line”. Each option string has short synonym. For example “icase” has short synonym “i”. Some options take effect only if other option (or options) is set. This is called preconditions.

There are a few special directives that specifies how options for an expression is merged with default options:

Single '*' char as option instruct implementation to add all default options to expression options. This can be useful if you want to use default option but add or remove a few other options and don't want to repeat long list of default options. For example: “*,icase” mean “use default plus case insensitive”.

'!' char before any option instruct implementation to remove this option from expression option. For example “*,!match_single_line” mean “use default minus match single line”.

Most of syntax and matching flags options available for Boost Regex library are mirrored with Expression Options. The following table contains all supported options with explanation that was taken from [1].

Option	***Synonym***	***Precondition***	***Explanation***
literal	l		Treat the string as a literal (no special characters).
icase	i		Specifies that matching of regular expressions against a character container sequence shall be performed without regard to case.
collate	c	all except *literal*	Specifies that character ranges of the form `[a-b]` should be locale sensitive.
optimize	o		Specifies that the regular expression engine should pay more attention to the speed with which regular expressions are matched, and less to the speed with which regular expression objects are constructed. Otherwise it has no detectable effect on the program output. This currently has no effect for Boost.Regex.
bk_plus_qm	bq	*basic* or *sed* or *grep* or *emacs*	When set then `\?` acts as a zero-or-one repeat operator, and `\+` acts as a one-or-more repeat operator.
bk_vbar	bv	*basic* or *sed* or *grep* or *emacs*	When set then `|` acts as the alternation operator.
no_intervals	v	*basic* or *sed* or *grep* or *emacs*	When set then bounded repeats such as a{2,3} are not permitted.
no_char_classes	e	*basic* or *sed* or *grep* or *emacs*	When set then character classes such as `[[:alnum:]]` are not allowed.
no_escape_in_lists	p	*extended* or *egrep* or *awk* or *basic* or *sed* or *grep* or *emacs*	When set this makes the escape character ordinary inside lists, so that `[\b]` would match either '\' or 'b'. This bit is on by default for POSIX-Extended regular expressions, but can be unset to force escapes to be recognised inside lists.
no_mod_m	m	ECMAScript or perl or JavaScript or JScript or normal	Normally Boost.Regex behaves as if the Perl m-modifier is on: so the assertions ^ and $ match after and before embedded newlines respectively, setting this flags is equivalent to prefixing the expression with (?-m).

Option	*Synonym*	*Precondition*	*Explanation*
mod_x	x	ECMAScript or perl or JavaScript or JScript or normal	Turns on the perl x-modifier: causes unescaped whitespace in the expression to be ignored.
mod_s	s	ECMAScript or perl or JavaScript or JScript or normal	Normally whether Boost.Regex will match "." against a newline character is determined by the match flag `match_dot_not_newline`. Specifying this flag is equivalent to prefixing the expression with `(?s)` and therefore causes "." to match a newline character regardless of whether `match_not_dot_newline` is set in the match flags.
no_mod_s	nd	ECMAScript or perl or JavaScript or JScript or normal	Normally whether Boost.Regex will match "." against a newline character is determined by the match flag `match_dot_not_newline`. Specifying this flag is equivalent to prefixing the expression with `(?-s)` and therefore causes "." not to match a newline character regardless of whether `match_not_dot_newline` is set in the match flags.
basic	b		Specifies that the grammar recognized by the regular expression engine is the same as that used by POSIX basic regular expressions in IEEE Std 1003.1-2001, Portable Operating System Interface (POSIX), Base Definitions and Headers, Section 9, Regular Expressions (FWD.1).
extended	X		Specifies that the grammar recognized by the regular expression engine is the same as that used by POSIX extended regular expressions in IEEE Std 1003.1-2001, Portable Operating System Interface (POSIX), Base Definitions and Headers, Section 9, Regular Expressions (FWD.1). Refer to the POSIX extended regular expression guide for more information. In addition some perl-style escape sequences are supported (The POSIX standard specifies that only "special" characters may be escaped, all other escape sequences result in undefined behavior).
normal	N		As *ECMAScript.*
emacs	E		Specifies that the grammar recognised is the superset of the POSIX-Basic syntax used by the emacs program.

Option	*Synonym*	*Precondition*	*Explanation*
awk	A		Specifies that the grammar recognized by the regular expression engine is the same as that used by POSIX utility awk in IEEE Std 1003.1-2001, Portable Operating System Interface (POSIX), Shells and Utilities, Section 4, awk (FWD.1). That is to say: the same as POSIX extended syntax, but with escape sequences in character classes permitted. In addition some perl-style escape sequences are supported (actually the awk syntax only requires \a \b \t \v \f \n and \r to be recognised, all other Perl-style escape sequences invoke undefined behavior according to the POSIX standard, but are in fact recognised by Boost.Regex).
grep	G		Specifies that the grammar recognized by the regular expression engine is the same as that used by POSIX utility `grep` in IEEE Std 1003.1-2001, Portable Operating System Interface (POSIX), Shells and Utilities, Section 4, Utilit\ies, grep (FWD.1). That is to say, the same as POSIX basic syntax, but with the newline character acting as an alternation character; the expression is treated as a newline separated list of alternatives.
egrep	EG		Specifies that the grammar recognized by the regular expression engine is the same as that used by POSIX utility grep when given the -E option in IEEE Std 1003.1-2001, Portable Operating System Interface (POSIX), Shells and Utilities, Section 4, Utilities, grep (FWD.1). That is to say, the same as POSIX extended syntax, but with the newline character acting as an alternation character in addition to "|".
sed	S		As *basic*.
perl	P		As *ECMAScript.*
ECMAScript	ES		Specifies that the grammar recognized by the regular expression engine uses its normal semantics: that is the same as that given in the ECMA-262, ECMAScript Language Specification, Chapter 15 part 10, RegExp (Regular Expression) Objects (FWD.1). This is functionally identical to the Perl regular expression syntax. Boost.Regex also recognizes all of the perl-

Option	*Synonym*	*Precondition*	*Explanation*
			compatible `(?...)` extensions in this mode.
JavaScript	J		As *ECMAScript.*
JScript	JS		As *ECMAScript.*
match_not_bol	nb		Specifies that the expression "^" should not be matched against the sub-sequence [first,first).
match_not_eol	nl		Specifies that the expression "$" should not be matched against the sub-sequence [last,last).
match_not_bob	no		Specifies that the expressions "\A" and "\`" should not match against the sub-sequence [first,first).
match_not_eob	ne		Specifies that the expressions "\'", "\z" and "\Z" should not match against the sub-sequence [last,last).
match_not_bow	nw		Specifies that the expressions "\<" and "\b" should not be matched against the sub-sequence [first,first).
match_not_eow	NW		Specifies that the expressions "\>" and "\b" should not be matched against the sub-sequence [last,last).
match_not_dot_newline	nn		Specifies that the expression "." does not match a newline character. This is the inverse of Perl's s/ modifier.
match_not_dot_null	f		Specifies that the expression "." does not match a character null '\0'.
match_prev_avail	a		Specifies that --first is a valid iterator position, when this flag is set then the flags match_not_bol and match_not_bow are ignored by the regular expression algorithms (RE.7) and iterators (RE.8).
match_any	y		Specifies that if more than one match is possible then any match is an acceptable result: this will still find the leftmost match, but may not find the "best" match at that position. Use this flag if you care about the speed of matching, but don't care what was matched (only whether there is one or not).
match_not_null	L		Specifies that the expression can not be matched against an empty sequence.
match_continuous	u		Specifies that the expression must match a sub-sequence that begins at first.

Option	*Synonym*	*Precondition*	*Explanation*
match_partial	r		Specifies that if no match can be found, then it is acceptable to return a match [from, last) such that from!= last, if there could exist some longer sequence of characters [from,to) of which [from,last) is a prefix, and which would result in a full match. This flag is used when matching incomplete or very long texts, see the partial matches documentation for more information.
match_not_initial_null	h		Don't match initial null.
match_all	d		Must find the whole of input even if match_any is set.
match_perl	ml		Specifies that the expression should be matched according to the Perl matching rules, irrespective of what kind of expression was compiled.
match_posix	mx		Specifies that the expression should be matched according to the POSIX leftmost-longest rule, regardless of what kind of expression was compiled. Be warned that these rules do not work well with many Perl-specific features such as non-greedy repeats.
match_nosubs	ms		Makes the expression behave as if it had no marked subexpressions, no matter how many capturing groups are actually present.
match_single_line	l		Equivalent to the inverse of Perl's m/ modifier; prevents ^ from matching after an embedded newline character (so that it only matches at the start of the text being matched), and $ from matching before an embedded newline (so that it only matches at the end of the text being matched).

7.1.3.4 Expression File Format

If pattern expressions are loaded from file this file must be of the format specified in this section. Expression File is a text file, each line specifies single expression with optionally specified Expression ID and/or Expression Options. Lines that begin with '#' character as first non-whitespace character are considered comments and isignored. Empty lines or lines with only whitespaces are ignored too.

Expression line consists of three ordered whitespace-separated fields:

First mandatory field which is a pattern expression itself. If pattern should contain whitespaces or starts with '/' or '#' characters it must be quoted with '/' characters. Pattern expression supports legacy format

which allows specification of one or more of the following options right after closing '/' quotation character:

i – equivalent to icase;

m – equivalent to !match_single_line;

s – equivalent to !match_not_dot_newline;

All legacy options are combined with default options and are processed before usual *Expression Options*. Expression must have format defined by *Expression Options*.

Examples of expressions:

```
\.gif$
/with whitespace/
/legacy/ims
```

Second optional field which defines Expression ID. If this field is omitted or contains '*' character then default ID is used. The field can be omitted if *Expression Options* are omitted too. Otherwise if you don't want to change default ID for expression but do want to change Expression Options you must use '*' character for *Expression ID* field. To avoid collisions '/' character is not allowed within *Expression ID*.

Examples of expressions with IDs:

```
\.pdf$ pdf
/with whitespace/ ws
/legacy/ims leg
```

Third optional field which defines Expression Options. Options are defined as list of comma-separated option strings. No whitespaces are allowed between options. Expression Options was described above. If options field is omitted default options are used.

Examples of expressions with IDs and options:

```
^Subject:.*rolex.$          rolex     *,!match_single_line,icase
^Subject:.*mortgage.*$ mortgage     *,!I,i
```

7.2 LDAP Configuration Guide

7.2.1 Overview

MPP can store policy group membership, ACL information and other items in an LDAP directory, a specialized database optimized for reading, browsing and searching. Directories tend to contain descriptive, attribute-based information and support sophisticated filtering capabilities.

In LDAP, directory information is in a hierarchical tree-like structure of entries. Traditionally, this structure reflected the geographic and/or organizational boundaries. An entry is a collection of attributes that has a globally unique Distinguished Name (DN) that is used to refer to the entity. Each of the entry's attributes has a type and one or more values. The types are typically mnemonic strings, like "cn" for common name, or "mail" for email address. The syntax of values depends on the attribute type.

An entry has an object class (raeMPP) that has attributes raeMPPGroupNameRecipient and raeMPPGroupSender.

An entry is referenced by its distinguished name, which is constructed by taking the name of the entry itself (called the Relative Distinguished Name or RDN) and concatenating the names of its ancestor

entries. For example, the entry for Barbara Jensen has an RDN of uid=babs and a DN of uid=babs,ou=People,dc=example,dc=com.

LDAP defines operations for interrogating and updating the directory. Operations are provided for adding and deleting an entry from the directory, changing an existing entry, and changing the name of an entry. Most of the time, LDAP is used to search for information in the directory. The LDAP search operation allows some portion of the directory to be searched for entries that match some criteria specified by a search filter. Information can be requested from each entry that matches the criteria.

7.2.2 LDAP Caching

MPP can cache LDAP query results to reduce the burden on LDAP servers for repetitive queries. Below are guidelines for configuring the LDAP cache.

1. Consider a system has many users defined in LDAP but only X of them are frequently addressed. LDAP is not frequently changed and you want to speed up mail processing. In this case:

cache size = X * 1.1 * 8 / 1024

2. Consider a Customer has X users defined in LDAP and all of them are frequently addressed. LDAP is not frequently changed and you would like to speed up mail processing. In this case:

cache size = X * 8 / 1024

For more details on LDAP, please consult documentation at sites such as www.openldap.org and the appropriate RFCs.

7.2.3 Configuring LDAP Connections

Connecting to an LDAP server requires

- Identification of the server and the client
- Authentication
- Caching

The LDAP connection or bind requires identification of the server through a network name or IP address and the TCP port. The LDAP client may need to be identified (source IP address). The bind to the server may fail due to timeouts.

Authentication is typically with a password called the bind password.

Queries may access static or dynamic data. In static mode all data are loaded at startup and at restart. In contrast, in the dynamic mode the LDAP server is queried for each sender/recipient in each email.

Each dynamic query result may be cached. Caching policy is very simple, based on a static cache size and expiration time. If the cache grows more than a maximum cache size, the oldest record is deleted. If the cached record is outdated (as determined by expiration time), this record is removed and LDAP lookup is performed again. Expiration time is checked only at the time of query.

7.2.4 MPP LDAP Schema

MPP queries an LDAP directory for policy membership and does not store complete configurations in LDAP. Storing complete configuration is not scalable for large directories in general.

MPP has a custom schema as follows:

```
# raeMpp object class
#
# Depends upon RFC 1274 (uid/dc)
#   (core.schema)

attributetype ( 1.3.6.1.4.1.21157.2.1.1
    NAME 'raeMppGroupnameRecipient'
    DESC 'Group name for e-mail`s that contains current e-mail in
Recipients field (used by RAE-INTERNET MPP)'
```

```
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.21157.2.1.2
    NAME 'raeMppGroupnameSender'
    DESC 'Group name for e-mail's that contains current e-mail in
Sender field (used by RAE-INTERNET MPP)'
    SUP raeMppGroupnameRecipient )

# raeMpp
# The raeMpp represents group names and e-mail address which are
associated
# with a person in some way.
objectclass   ( 1.3.6.1.4.1.21157.2.2.1
    NAME 'raeMpp'
    DESC 'Object class for group names for RAE-INTERNET MPP'
    AUXILIARY
    MAY (
        raeMppGroupnameRecipient $ raeMppGroupnameSender $ mail )
    )
```

LDAP schema-related parameters

- mail_attribute
- sender_group_attribute
- receiver_group_attribute

The following are the general steps

- Define Requirements
- Modify the existing schema
- Import Data

7.2.5 MPP/LDAP sample how-to

We assume that OpenLDAP and MPP 2.x are installed on the working machine.

1) Setting up OpenLDAP

==========

NOTE: Crypted passwords and LDAP/SSL are not the object of this how-to. Assuming that configuration files for OpenLDAP are installed in /etc/openldap, please make the following changes (replace "example" and "com" with your own domain parts). Copy rae-mpp.schema in to /etc/openldap/schema:

==========

```
BASE    dc=example, dc=com
URI     ldap://ldap.example.com
 /etc/openldap/slapd.conf
==========================
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/rae-mpp.schema
```

```
database        bdb
suffix          "dc=example,dc=com"
rootdn          "cn=Manager,dc=example,dc=com"
rootpw          secret
index   objectClass     eq
```

Now, you can start the slapd daemon:

```
/etc/init.d/slapd start or
/usr/lib/openldap/slapd or
/usr/lib/openldap/slapd -u USER -g GROUP
```

You must have the directory option specified in slapd.conf and that directory owned by USER and GROUP, i.e. in /etc/slapd.conf

```
directory       /var/lib/openldap-data
chown -R ldap:ldap /var/lib/openldap-data
/usr/lib/openldap/slapd -u ldap -g ldap
```

Create initial LDIF in your favorite editor and save as initial.ldif after edits are complete

```
=======================
dn: dc=example,dc=com
objectClass: dcObject
objectClass: organization
o: EXAMPLE
dc: example

dn: cn=Manager,dc=example,dc=com
objectClass: organizationalRole
cn: Manager
=============================
```

Use:

```
ldapadd -x -D "cn=Manager,dc=example,dc=com" -W -f initial.ldif
```

Password asked is: "secret" as defined in by rootpw in /etc/openldap/slapd.conf

Add some entries in another LDIF file: users.ldif

```
================================================
dn: cn=User1 Stuff, dc=example, dc=com
cn: User1 Stuff
objectClass: person
objectClass: raeMpp
mail: user1@example.com
raeMppGroupnameSender: group1
raeMppGroupnameRecipient: group2
sn: User1 FirstName LastName

dn: cn=User2 Stuff, dc=example, dc=com
cn: User2 Stuff
objectClass: person
objectClass: raeMpp
mail: user2@example.com
raeMppGroupnameSender: group1
raeMppGroupnameRecipient: group2
sn: User2 FirstName LastName

================================================
```

```
ldapadd -x -D "cn=Manager,dc=example,dc=com" -W -f users.ldif
```

Password asked is: "secret" as defined in by rootpw in /etc/openldap/slapd.conf

2) MPP configuration

From the users.ldif we will use the field "mail" to match an e-mail address

raeMPPGroupnameSender will point to the MPP group name used to scan the mail when e-mail address is the sender address

raeMPPGroupnameRecipient will point to the MPP group name used to scan the mail when e-mail address is the recipient address

In mppd.conf.xml we should have:

```
<groups>
...
<group id="group1">
 ... settings for this group here ...
</group>

<group id="group2">
 ... settings for this group here ...
</group>

 </groups>
 .....
        <ldaps>
                <ldap_use>yes</ldap_use>
                <ldap_dynamic>yes</ldap_dynamic>
                <ldap id="sample">
                        <base_dn>dc=example,dc=com</base_dn>
                        <server>192.168.0.1</server>
                        <server_timeout>30000</server_timeout>
                        <port>389</port>

<bind_dn>cn=Manager,dc=example,dc=com</bind_dn>
                        <bind_pw>secret</bind_pw>
                        <custom_filter_dynamic>
((<mail_attribute>=$ADDRESS$)(<mail_attribute>=$DOMAIN$))</custom_
filter>
                        <mail_attribute>mail</mail_attribute>

<sender_group_attribute>raeMppGroupnameSender</sender_group_attrib
ute>

<receiver_group_attribute>raeMppGroupnameRecipient</receiver_group
_attribute>
                        <use_cache>no</use_cache>
                        <cache_size>1024</cache_size>
                        <cache_ttl>30</cache_ttl>
                        <search_scope>2</search_scope>
                        <version>3</version>
                </ldap>
        </ldaps>
```

The mppd daemon can be started/restarted now. Actual ldap queries will print in the MPP log file with LOGGING set to DEBUG. Please refer to them when troubleshooting.

```


<!-----------------------------------------------------------------
--------
    These are server-wide settings. They will be used as a default
values if
    not specified in 'local' entity.
```

```
    ------------------------------------------------------------
----------->
    <common> …
    </common>

    <!-------------------------------------------------------------
------------
    This entity describes a set of MPP groups. Each group should
have its own
    'group' entity and may override a number of values from
'common' section.
    ------------------------------------------------------------
----------->
    <groups>  …
    </groups>
    <-------------------------------------------------------------
-------------
    Enter global parameters for LDAP use.
    ------------------------------------------------------------
------------>
    <ldaps>
         <!-----------------------------------------------------
-
         Enter information for a specific LDAP server, e.g.
"sample"
         -----------------------------------------------------
>
        <ldap id="sample">

        </ldap>
    </ldaps>
</mppd>
```

7.3 SNMP

7.3.1 Introduction

SNMP is a network protocol used to manage networked systems. MPP 2.x supports SNMP for performance management. For example, network managers can use SNMP to monitor MPP. MPP works with the net SNMP agent which is standard on most Unix operating systems. MPP uses the agentx protocol to speak with the snmp daemon.

7.3.2 Requirements

- net-snmp version 5.2.1 or higher are recommended.

7.3.3 Setup of NET-SNMP

This step is generally not required as most default net SNMP installations should work with MPP. If net SNMP is functional, skip to step 3.

The MPP SNMP module was completely redesigned in version 2.5 and now runs as an net SNMP agent. In order for MPP to work as described below you must use the updated MPP MIB included in the installation packages.

1) Configure and make net-snmp if the installed snmp daemon lacks SNMPv3 support.

Download & unpack latest stable net-snmp sources from http://www.net-snmp.org. Before starting, read README.<OS> file located in net-snmp base directory.

Follow the instructions from INSTALL file. The build process will default to /usr/local as the install root.

The configure script will set dynamically loadable module support on (--enable-shared on) by default if the operating system supports dynamic libraries. After the software has been made, check for support in your agent by looking at the output of the "snmpd -H" command for the "dlmod" token. If it's listed, the compiled agent supports it.

2) Configure NET-SNMP installation

Follow the instructions in README.snmpv3 located in net-snmp base directory. Don't forget to test your setup as described in the file.

Create any Net-SNMP user with read/write privileges with:

```
   net-snmp-config --create-snmpv3-user -a "user_password"
user_name
```

This command is generally located in /usr/lobal/bin or /usr/bin depending on your installation preferences.

3. Tell Net-SNMP daemon to run as master agent. Add the following line into snmpd.conf

```
master agentx
```

This file generally located in /usr/local/share/snmp or /usr/share/snmp or other depending on you installation preferences.

4. Tell Net-SNMP management application to use MPP-MIB.txt file by adding the following line into snmp.conf.

```
    mibfile /usr/local/MPP/snmp/MPP-MIB.txt.
```

5. Configure MPP. Make sure that lib/libnetsnmpengine.so is a link to lib/libnetsnmpengine_5_2_2.so. Make sure that you have new MIB at snmp/MPP-MIB.txt. In mppd.conf.xml under common->snmp set:

```
<snmp_use_statistics>yes</snmp_use_statistics>
```

Other related option can have default values. You can consider to specify:

<snmp_cache_ttl> Cache TTL in seconds

<snmp_track_senders> Whether to track senders

<snmp_track_recipients> Whether to track recipients

6. Start Net-SNMP daemon with the following command;

```
    /usr/local/sbin/snmpd  or /usr/sbin/snmpd
```

7. Re-Start MPP.

8. Test SNMP commands:

```
snmpwalk -v 3 -u user_name -l authNoPriv -a MD5 -A user_password
localhost MPP-MIB::mppSnmpMIB

snmptable -v 3 -u user_name -l authNoPriv -a MD5 -A user_password
localhost MPP-MIB::mppLibraryTable

snmpget -v 3 -u denis -l authNoPriv -a MD5 -A 12345678 localhost
MPP-MIB::mtaReceivedMessages.0
defPrivPassphrase   "Bet you can't see me"
```

7.3.4 Supported SNMP Variables

The following list represents the currently supported MIB variables. For the most current list, please consult the MPP-MIB.txt that came with the software distribution or the support website.

```
version   Version of mppd
Library_version  Version of each mppd library
System_name  Hostname of System or defined system name
Mta_name  MTA that MPP is configured for
Thread_count
Location  Configurable system location
Up_time   Time since daemon started
*Memory_utilization Amount of RAM mppd consumes. Not implemented
but will be in future builds.
Message_count Counts all messages processed by mppd
Message_rate_second Rate of messages processed by mppd.
Calculated as messages per second, updated every 30 seconds as an
average
Message_rate_hour   Average rate of messages per hour, updated
hourly
Message_volume_hour Amount in Megabytes's of email processed per
hour
Message_volume_second  Amount in Kilobytes's of email processed
per second
Message_recievers   Counts how many messages each receiver in cc,
to and bcc receives.  Flushed daily, stored for up to 1 week
Message_senders  Counts how many emails for each sender
Virus_count  Count total number of viruses
Virus_name   Count number of each virus by name
Spam_count   Count total number of emails classified as spam by
any engine
Scan_errors  Count total number of scan errors
Scan_error_count Count number of scan errors by error type
Errors Count total number of other errors
Error_count  Count errors by error name
Encrypted_msg_count Count total number of encrypted messages as
reported by scan engines
Max_mime_parts_exceeded Count total number of messages exceeding
max_mime_parts
Max_recursion_level_exceeded  Count total number of messages
exceeding max_recursion_level
Max_mime_parts_exceeded_sender   Track senders of each message
exceeding max_mime_parts
Max_recursion_level_exceeded_sender  Track senders of each message
exceeded max recursion level
```

These variables may be used with a SNMPv2 or SNMPv3 manager to monitor MPP performance. Key variables include virus_count, and spam_count.

7.4 Obsolete Commands from MPPv3

```
<email_server_in_threads_min>
<email_server_in_threads_max>
<email_server_in_threads_relax_timeout>
    This options are obsolete and removed. Now number of smtp
    thread is fixed and equal to a number of processing threads.
    Processing threads are shared with Policy Server.
```

```
<email_server_in_timeout_connect>
    This options is obsolete for postfix.

New options:

<smtp_socket>
    Additional addresses to bind to. Multiple options can be specified -
    server will bind to all corresponding ports. Port can be specified
    as name according to /etc/services. Host address can be specified
    as IPv6 address or DNS name that resolves to IPv6 address.
    Unix sockets are not supported that time.

    Value: string (URI)
    Default: inet:10025@localhost

<smtp_socket_retry_period>
    If bind to port failed then it is retried with period
    specified with this option.

    Value: integer (milliseconds)
    Default: 15000

<smtp_buffer_size_envelope_read>
    Size of received (read) buffer of a socket while reading SMTP
    envelope. Applied for all specified sockets.

    Value: integer (bytes)
    Default: 1024

<smtp_buffer_size_data_read>
    Size of received (read) buffer of a socket while reading SMTP
    data. Applied for all specified sockets.

    Value: integer (bytes)
    Default: 16384

<smtp_buffer_size_write>
    Size of send (write) buffer of a socket. Applied for
    all specified sockets.

    Value: integer (bytes)
    Default: 128
```