MessageFilter API

A Message Filtering Interface for Mailtraq®

April 2004

Revision 2.2

Introduction

The MessageFilter API allows third-party and in-house software to independently access, filter and/or modify e-mail processed by Mailtraq. The API is designed to be highly robust, simple and reliable, particularly with respect to deployment and installation.

Typical uses would include Anti-Virus scanners, Anti-Spam engines, Attachment Handling utilities and Automated Mail Processing Systems. The MessageFilter API is designed to readily accommodate all of these types of software.

Mailtraq can support an arbitrary number of filters installed simultaneously and will arbitrate the passing of e-mail from one filter to another.

The software that implements the MessageFilter API is called the “Filter”. Filters indicate their presence to Mailtraq by registering in the Windows Registry under pre-defined keys. Filters indicate that they are “active” by placing a specially named file in the input directory. Mailtraq will only pass messages to the filter if it meets both of these conditions.

Mailtraq can also be responsible for starting the filters and ensuring that they continue to operate. If a filter becomes unstable and crashes, Mailtraq can restart it.

Operation

A filter handles messages by reading them from an Input Queue and then writing to an Output Queue. The filter may choose not to place a given message in an Output Queue. The paths for these queues are registered in the Windows Registry. Multiple filters can operate in sequence, but Mailtraq will move the files from the Output Queue of one filter to the Input Queue of the next filter.

Message Format

E-Mail Messages are stored in Mailtraq Message Format (MMSG), which is essentially the entire e-mail message (including headers) prefixed with the Message Envelope. Mailtraq expects all messages appearing in the Output Queue to precisely adhere to this format, using the original filename.

See *Mailtraq Message Format (MMSG)* on page 5 for more information on the message file format.

Processed Messages

The filter may modify the message, but the filename must not be changed in order for Mailtraq to track a specific message through the routing system. Occasionally messages may be tagged for some action (for example, re-routing to a specific destination) and this action may only take place after the filters have processed the message. Mailtraq uses the filename to recognize the file after this process.

Created Messages

The Filter may place messages that it generates in the Output Queue (such as notifications to users). Thus, it is not restricted to only passing on filtered messages.

Note: Use a unique filename for each message, as it could be any length of time before Mailtraq collects the message (it may not even be running) and a new message with the same name would overwrite the existing message.

Log Messages

Log Messages can be issued to Mailtraq by placing a specially named file in the Output Queue. This file should have a name that begins with a period (e.g. ".message1234"). The file should contain a single line of text, which will be inserted into the Mailtraq log (both Console and Disk).

Note: Try to use a unique filename for each log entry to prevent the accidental overwriting of log messages with new messages before Mailtraq can collect them.

See *Mailtraq Log Line Format* on page 6 for more information.

Processes

The filter should be a single process that continuously monitors the Input Queue until it is terminated. This will allow Mailtraq to handle the startup of the process and monitor it. The filter process should not have a user interface itself, but rather a separate user interface process that can communicate with it. This will allow the filter to operate effectively in a windowless environment (such as a Windows Service).

Enabling the Filter

Mailtraq will consider the filter active only if a file named ".active" is placed in the Input Queue. Mailtraq will not delete or attempt to write this file, it simply checks to see if it exists.

If this file does not exist, Mailtraq will not put any messages in the Input Queue, but will rather continue to the next registered filter or process the mail.

Mailtraq will, however, remove and process files from the Output Queue when a filter is inactive.

Registering a Filter

During installation, the filter should create a new key underneath

```
HKEY_LOCAL_MACHINE\Software\Fastraq\Mailtraq\MessageFilters
```

The name of the key is unimportant, but should be unique to the filter. In this key, the following values should be set

Description	A textual description of the filter

InQueue	The path (ending in a backslash) where Mailtraq should place messages for the filter to process
OutQueue	The path (ending in a backslash) that Mailtraq should monitor for messages being sent or passed on by the filter.
StartProcess (Optional)	This is the path to an executable that Mailtraq should launch and monitor. Typically this will be the Filter itself. If this value does not exist, Mailtraq will not attempt to launch the executable.
ProcessRunLock (Optional)	This is the path to a lock file that the Filter will create. Mailtraq will attempt to launch the **StartProcess** if it can obtain exclusive access to this file.

Example Installation

```
Description    REG_SZ      My Message Filter
InQueue        REG_SZ      C:\Program Files\MyFilter\inqueue\
OutQueue       REG_SZ      C:\Program Files\MyFilter\outqueue\
StartProcess   REG_SZ      C:\Program Files\MyFilter\MyFilter.exe
ProcessRunLock REG_SZ      C:\Program Files\MyFilter\MyFilter.lck
```

Automatic Startup

Mailtraq can be responsible for the start up and maintaining of the filter process. This is dependent on the presence of the **StartProcess** and **ProcessRunLock** values in the registry.

Starting the Process

If the **StartProcess** value is entered into the Windows Registry, then when Mailtraq starts, it will launch the executable identified by that value. The executable will run as a child process of Mailtraq, but will not inherit handles. This means that if Mailtraq is shut down, the filter can continue to run and keeping the filter running will not prevent Mailtraq from shutting down. Running as a child process means that it will run in the same process space as Mailtraq. If Mailtraq is running as a Windows Service (under the **Localsystem** account) then the filter will run under the same account and have the same privileges.

Monitoring the Process

If the **ProcessRunLock** value is entered into the Windows Registry, then Mailtraq will monitor the file identified by that value. When the filter starts, it should create this file and hold an exclusive lock on it. When the filter process is terminated (by the filter itself or by Windows) the lock will automatically be released. Mailtraq uses this file to determine whether or not the filter is running. If Mailtraq can obtain an exclusive lock, it will start the process according to the **StartProcess** operation above.

ProcessRunLock does not need to exist in order for Mailtraq to simply execute the process when it starts.

Recommended Operation

The filtering process is a well-recognized method for message handling, and as a result experience has shown a number problems commonly appear in implementations.

Move Messages

It is good practice to move a message before operating on it. Should the message prove to be corrupt or unreadable, resulting in the filter process terminating, the procedure will not be repeated upon restart.

Do not Spin when Monitoring

A process is considered “spinning” when it is executing a tight loop and can only exit as a result of action elsewhere. Spinning is highly CPU-intensive, so avoid it. When monitoring the Input Queue, test to see if files exist, and if not perform a **sleep()** operation before repeating the test.

Lock Output Files

When writing output files, ensure that the filter process maintains an exclusive lock on the file. Failure could result in Mailtraq reading and processing an incomplete message.

Mailtraq Message Format (MMSG)

The Mailtraq Message Format consists of a single text file. The first line of the file begins with the prefix **FROM:** followed by the return-path of the message envelope (which may be blank).

This is followed by one or more lines prefixed with **RCPT:** and followed by the destination address of the message. Multiple recipients are identified by multiple **RCPT:** lines.

After this envelope, the message itself follows in standard RFC822 format.

A sample RFC822 message (with two recipients) is shown below:—

```
FROM: elric@novitraq.com
RCPT: victor@novitraq.com
RCPT: bob@enstar.net
Return-Path: <elric@novitraq.com>
Received: from KENSINGTON ([204.92.85.4]) by novitraq.com
          with SMTP (Mailtraq/2.0.0.1201) id NVTRAD0B221F4B
          for bob@enstar.net; Fri, 29 Mar 2001 11:26:20 -0500
From: "Elric Pedder" <elric@novitraq.com>
To: "Victor Lee" <victor@novitraq.com>
CC: "Bob North" <bob@enstar.net>
Subject: MessageFilter API
Date: Fri, 29 Mar 2001 11:31:00 -0500
Organization: Novitraq Inc
Message-ID: <005301c1d73f$1cc1d0e0$04555ccc@novitraq.com>
MIME-Version: 1.0
Content-Type: text/plain
In-Reply-To: <pmv8aucc93p9au8i07u7lu6llgvd0osj8k@127.0.0.1>

Please forward a revised copy of the API
Thank you,
- Elric
```

The message file may contain MIME attachments and other encoding methods. It is the responsibility of the filter to properly decode such messages. If the message is to be modified, the output message must be fully MIME compliant.

Mailtraq Log Line Format

The log files that can be placed in the OutQueue should contain one or more lines using the following BNF syntax :—

```
line ::= class-identifier space session-identifier space text
class-identifier ::= hex{8}
session-identifier ::= hex{8}
space ::= " "
hex ::= NUMERIC | "A" | "B" | "C" | "D" | "E" | "F"
text ::= CHAR*
```

Message Class

In the above syntax, the class-identifier is an eight-digit hexadecimal number indicating the log entry class. The class is calculated by adding all the relevant values from the list below :—

SMTP Server	0x00000001
SMTP Client	0x00000002
POP3 Server	0x00000004
POP3 Client	0x00000008
NNTP Server	0x00000010

NNTP Client	0x00000020
HTTP	0x00000040
Mail Router	0x00000080
Mailing List	0x00000100
Archive Action	0x00000200
Mailbox Action	0x00000400
Mail Action Summary	0x00000800
Connection	0x00001000
Dialup Activity	0x00002000
Error Message	0x00004000
Alert Message	0x00008000

Normally, all log entry classes would include 0x80 (Mail Router) as the MessageFilter operates as part of the mail routing system. If something occurs that the administrator should be aware of, the class should include the Alert value (0x8000). Thus, for normal status messages, use 0x80, and for alerts use 0x8080. If an error occurs (e.g. the message cannot be processed and attention is required) include 0x4000 (Error).

Session Number

The session number is used to group related entries together (for example, if they refer to the same message). Use a zero to indicate that the message has no session, which is normal for filters.

Text Part

The text part of the log entry can contain any 7-bit text string, which will be seen by the administrator. It is important to indicate where the message came from as no additional information will be added (other than a timestamp). You should prefix the text in a consistent manner with the name of the filter.

Examples

```
00000080 00000000 MyFilter: NVTR01 "MessageFilter API" scanned
00008080 00000000 MyFilter: NVTR02 "MessageFilter API" virus!
0000C080 00000000 MyFilter: Out of resources, shutting down
```