

Network Working Group
Document: [draft-martin-sieve-notify-01.txt](#)
Expires December 6, 2001

Tim Martin
Wolfgang Segmuller
1 June 2001

Sieve -- An extension for providing instant notifications

[<draft-ietf-sieve-notify-01.txt>](#)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Distribution of this memo is unlimited.

Abstract

Users go to great lengths to be notified as quickly as possible that they have received new mail. Most of these methods involve polling to check for new messages periodically. A push method handled by the final delivery agent gives users quicker notifications and saves server resources. This document does not specify the notification method but is expected that using existing instant messaging infrastructure such as Zephyr, ICQ, or SMS messages will be popular. This draft describes an extension to the Sieve mail filtering language that allows users to give specific preferences for notification of Sieve actions.

Expires December 6, 2001

Martin

[Page 1]

TTaablllee ooff CCoonntteennttss

Status of this Memo [1](#)

Abstract [1](#)

[1.](#) Introduction [3](#)

[1.1.](#) Conventions Used in This Document [3](#)

[2.](#) Capability Identifier [3](#)

[3.](#) Actions [3](#)

[3.1.](#) Notify action [3](#)

[3.2.](#) Denotify Action [5](#)

[4.](#) Interaction with Other Sieve Actions [6](#)

[5.](#) Security Considerations [7](#)

[6.](#) Acknowledgments [7](#)

[7.](#) Copyright [7](#)

[8.](#) References [8](#)

[9.](#) Author's Addresses [8](#)

1. Introduction

This is an extension to the Sieve language defined by [SIEVE] for providing instant notifications of sieve actions that have been preformed. It defines the new action "notify".

This document does not dictate the notification method used. Examples of possible notification methods are Zephyr and ICQ. The method shall be site-defined.

Sieve interpreters for which notifications are impractical or is not possible SHOULD ignore this extension.

Conventions for notations are as in [SIEVE] [section 1.1](#), including use of [KEYWORDS].

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

2. Capability Identifier

The capability string associated with the extension defined in this document is "notify".

3. Actions

3.1. Notify action

```
Syntax:  notify [":method" string]
          [":id" string]
          [":options" 1*(string-list / number)]
          [<":low" / ":normal" / ":high">]
          ["message:" string]
```

The Notify action specifies that a notification should be sent to the user upon successful handling of this message.

The format of the notification is implementation-defined. However,

all content specified in the notify action, including Sieve actions taken on the message, SHOULD be included. If errors occurred in another action they SHOULD be reported in the notification. In addition, if the notification method does not provide a timestamp, one SHOULD be appended to the notification. Implementations SHOULD NOT include extraneous information.

The :method tag identifies the notification method that will be used. If the :method tag is not specified, the default implementation defined notification method MUST be used. The possible values of this will be site-specific. If a method is specified that the implementation does not support, the notification MUST be ignored. An implementation treats this as a warning condition and execution of the sieve script MUST continue.

The :id tag can be used to give the notify action an unique identifier. This identifier can be used later in the script to cancel the specific notify. The string may have any value and SHOULD NOT be included in the notification.

The :options tag is used to send parameters to the notification method. This might be the phone number for an SMS message or a user's ICQ identifier.

The priority parameter specifies the importance of the notification. The priority parameter has the following values: ":high" (very important), ":normal", and ":low" (not very important). If no priority is given, a default priority of ":normal" SHOULD be assumed. Some notification methods allow users to specify their state of activity (for example "busy" or "away from keyboard"). If the notification method provides this information it SHOULD be used to selectively send notifications. If, for example, the user marks herself as "busy", an implementation SHOULD NOT send a notification for a new mailing list message with a priority of :low, however the user should be notified of a high priority action. If the notification method allows users to filter messages based upon certain parameters in the message, users should be able to filter based upon priority. If the notification method does not support priority, then this parameter MUST be ignored.

The :message tag specifies the message data to be included in the notification. The entirety of the string SHOULD be sent but implementations MAY shorten the message for technical or aesthetic reasons. Message may include the message variables defined below. If the message parameter is absent, a default message of "\$from\$: \$subject\$" will be used.

\$from\$ - the display name or address of the [RFC 822](#) from

`$env-from$` - the address of the [RFC 821](#) from

`$subject$` - the subject of the message

`$text$` - the first text/* part

`$text[n]$` - the first n bytes of the first text/* part

If there are errors sending the notification, the Sieve interpreter SHOULD ignore the notification and not retry indefinitely.

This action MUST NOT cancel the implicit keep.

Example:

```
require ["notify","fileinto"];

if header :contains "from" "boss@example.org" {
    notify :high "This is probably very important";
}

if header :contains "to" "sievemailinglist@example.org" {
    notify :low "[SIEVE] $from$: $subject$";
    fileinto "INBOX.sieve";
}
```

3.2. Denotify Action

Syntax: `denotify [MATCH-TYPE string] [<":low" / ":normal" / ":high">]`

The denotify action can be used to cancel a previous notification. If the priority, `":low" / ":normal" / ":high"`, is specified, then only cancel those notifications with the specified priority. If a MATCH-TYPE with a string is specified, then only those notifications whose `:id` tag matches the specified string using the match-type operator are canceled. The `ascii-casemap` comparator MUST be used.

If no notifications exist that match the search criteria, then the denotify has no effect. A denotify only cancels notifications that have already been requested. It is not possible to preemptively cancel a notification.

The sequence:

```
denotify;
```

```
notify;
```

will still generate a notification. The denotify does not cancel the notify.

The following table shows which notifies would get cancelled:

	# what is cancelled
denotify	# all notifications
denotify :matches "*"	# all notifications with :id tag
denotify :high	# all high priority notifications
denotify :is "foobar"	# all notifications with id "foobar"
denotify :matches "foo*" :normal	# all normal priority notifications
	# with id that starts with "foo"

Example:

```
require "notify";

notify :method "sms" :options "408-555-1212" :id "foobar";
if header :contains "from" "boss@example.org" {
    notify :method "sms" :options "408-555-1212" :id "foobar" :high
        :message "BOSS: $subject$";
}

if header :contains "to" "sievemailinglist@example.org" {
    denotify :is "foobar";
}

if header :contains "subject" "FYI:" {
    # don't need high priority notification for
    # a 'for your information'
    denotify :is "foobar" :high;
}
```

4. Interaction with Other Sieve Actions

Notifications MUST be sent in all cases, unless a reject action is also executed. Users may wish to be notified of a message being discarded, for example. The reject action is given an exception because implementations may wish to restrict users from seeing the contents of a rejected message. However, notifications MAY be modified to not include any data from the original rejected message.

The notify action MUST NOT cancel the implicit keep.

The denotify action MUST NOT affect any actions other than the notify action.

Failures of other actions MAY be reported in the notification.

5. Security Considerations

Security considerations are discussed in [SIEVE]. Additionally implementations must be careful to follow the security considerations of the specific notification methods. It is believed that this extension does not introduce any additional security concerns.

The notify action is potentially very dangerous. The path the notification takes through the network may not be secure. An error in the options string may cause the message to be transmitted to someone it was not intended for.

Just because a notification is received doesn't mean it was sent by the sieve implementation. It might be possible to forge notifications with some notification methods.

6. Acknowledgments

Thanks to Larry Greenfield, Sarah Robeson, Tim Showalter, and Barry Leiba for help with this document.

7. Copyright

Copyright (C) The Internet Society 1999. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for

copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

8. References

[ABNF] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), Internet Mail Consortium, Demon Internet Ltd, November 1997.

[SIEVE]; Showalter, T.; "Sieve: A Mail Filtering Language"; [RFC 3028](#); Mirapoint, Inc.; January 2001

9. Author's Addresses

Tim Martin
Mirapoint Inc.
909 Hermosa Court
Sunnyvale, CA 94085

Phone: (408) 720-3835
EMail: tmartin@mirapoint.com

Wolfgang Segmuller
IBM T.J. Watson Research Center
30 Saw Mill River Rd
Hawthorne, NY 10532

Phone: (914) 784-7408
Email: whs@watson.ibm.com