This document defines two new actions for the "sieve" language that add and delete e-mail header fields.

1. Introduction

Email headers are a flexible and easy to understand means of communication between email processors. This extension enables sieve scripts to interact with other components that consume or produce header fields by allowing the script to delete and add header fields.

2. Conventions used.

Conventions for notations are as in [SIEVE] section 1.1, including use of [KEYWORDS] and "Syntax:" label for the definition of action and tagged arguments syntax.

The term "header field" is used here as in [RFC-2822] to mean a logical line of an e-mail message header.
The capability string associated with extension defined in this document is "editheader".

3. Action addheader

Syntax:

```
"addheader" [":last"] <name: string> <value: string>
```

The addheader action adds a header field to the existing message header. The name MUST be a valid 7-bit US-ASCII header field name as described by [RFC-2822] "field-name" nonterminal.

If the specified field value does not match the RFC 2822 "unstructured" nonterminal or exceeds a length limit set by the implementation, the implementation MUST either flag an error or encode the field using folding white space and the encodings described in RFC 2047 or RFC 2231 to be compliant with RFC 2822.

An implementation MAY impose a length limit onto the size of the encoded header field; such a limit MUST NOT be less than 998 characters, not including the terminating CRLF supplied by the implementation.

By default, the header field is inserted at the beginning of the existing header. If the optional flag ":last" is specified, it is appended at the end.

Example:

```
/* Don’t redirect if we already redirected */
if not header :contains "X-Sieve-Filtered"
  ["<kim@job.tld>", "<kim@home.tld>"]
{
  addheader "X-Sieve-Filtered" "<kim@job.tld>";
  redirect "kim@home.tld";
}
```

4. Action deleteheader

Syntax:

```
"deleteheader"
[":index" <fieldno: number> [":last"]]
[COMPARATOR] [MATCH-TYPE]
<field-name: string>
[<value-patterns: string-list>]
```

By default, the deleteheader action deletes all occurrences of the named header field.

The field-name is mandatory and always matched as a case-insensitive us-ascii string. The value-patterns, if specified, are matched according to the match type and comparator. If none are specified, all values match.

The field-name MUST be a valid 7-bit header field name as described by the [RFC-2822] "field-name" nonterminal.

If :index <fieldno> is specified, the attempts to match a value are limited to the header field <fieldno> (beginning
at 1, the first named header field). If :last is specified, the count is backwards; 1 denotes the last named header field, 2 the second to last, and so on. The counting happens before the <value-patterns> match, if any;

deleteheader :index 2 :contains "Received" "via carrier-pigeon"
deletes the second "Received:" header field if it contains the string "via carrier-pigeon" (not the second Received: field that contains "via carrier-pigeon").

5. Interaction with Other Sieve Extensions

Tests and actions such as "exist" or "header" that examine header fields MUST examine the current state of a header as modified by any actions that have taken place so far.

As an example, the "header" test in the following fragment will always evaluate to true, regardless of whether the incoming message contained an "X-Hello" header field or not:

    addheader "X-Hello" "World";
    if header :contains "X-Hello" "World"
    {
        fileinto "international";
    }

Actions that create messages in storage or in transport to MTAs MUST store and send messages with the current set of header fields.

For the purpose of weeding out duplicates, a message modified by addheader or deleteheader MUST be considered the same as the original message. For example, in an implementation that obeys the constraint in [SIEVE] section 2.10.3 and does not deliver the same message to a folder more than once, the following code fragment

    keep;
    addheader "X-Flavor" "vanilla";
    keep;

MUST only file one message. It is up to the implementation to pick which of the redundant "fileinto" or "keep" actions is executed, and which ones are ignored.

The "implicit keep" is thought to be executed at the end of the script, after the headers have been modified. (However, a canceled "implicit keep" remains canceled.)

6. IANA Considerations

The following template specifies the IANA registration of the Sieve extension specified in this document:

To: iana@iana.org
Subject: Registration of new Sieve extension

Capability name: editheader
7. Security Considerations

Someone with write access to a user’s script storage may use this extension to generate headers that a user would otherwise be shielded from (by a gateway MTA that removes them).

A sieve filter that removes headers may unwisely destroy evidence about the path a header has taken.

Any change in a message content may interfere with digital signature mechanisms that include the header in the signed material. Since normal message delivery adds "Received:" header fields to the beginning of a message, many such schemas are impervious to headers prefixed to a message, and will work with "addheader" unless :last is used.

Any decision mechanism in a user’s filter that is based on headers is vulnerable to header spoofing. For example, if the user adds an APPROVED header or tag, a malicious sender may add that tag or header themselves. One way to guard against this is to delete or rename any such headers or stamps prior to processing the message.

8. Acknowledgments


9. Author’s Address

Jutta Degener
Sendmail, Inc.
6425 Christie Ave, 4th Floor
Emeryville, CA 94608

Email: jutta@sendmail.com

10. Discussion

This section will be removed when this document leaves the Internet-Draft stage.

This draft is intended as an extension to the Sieve mail filtering language. Sieve extensions are discussed on the MTA Filters mailing
Changes from the previous version

Changed the duplicate restrictions from "messages with different headers MUST be considered different" to their direct opposite, "messages with different headers MUST be considered the same," as requested by workgroup members on the mailing list.

Expanded mention of header signature schemes to Security Considerations.

Added IANA Considerations section.

Appendices

Appendix A. References


Appendix B. Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.