

Sieve -- Sequential Execution of Multiple Scripts
<[draft-degener-sieve-multiscript-00.txt](#)>

Status of this memo

This document is an Internet-Draft and is subject to 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/1id-abstracts.html>

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

Abstract

This document defines sieve behavior relevant when multiple scripts are executed sequentially on the same message.

1. Introduction

E-mail messages frequently are processed by multiple agents that implement nested layers of corporate policies.

For example, a provider may offer access to services that perform spam- and virus filtering; a single corporation may restrict e-mail to certain subdomains, or filter for keywords; individual divisions within a corporation may implement even more intrusive handling of e-mail messages, for example by keeping a copy of all correspondence. Amidst all of this, of course, users may still use sieve filters to presort, redirect, or notify other accounts as in the classic sieve use scenario.

In this context, it is desirable to specify an execution model for sieve scripts when executed in series. This allows each layer of the mail filtering hierarchy to use a separate sieve script to express its policies.

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.

This document defines no sieve extensions and no capability string.

3. Sequential Script Execution Model

When multiple scripts are executed for a message, they are executed in a specific order.

Within this order, this document defines that trailing scripts will be executed as long as the message is being "kept", that is, as long as either an implicit or explicit "keep" is in effect.

In other words, for every script but the last, "keep" doesn't mean "file this message into INBOX", it means "process this message with the next sieve script."

4. Locality of script actions

This document strongly limits the effects of scripts on each other.

The "require" clauses at the beginning of a script only apply to this particular script, not to following ones. Different stages in the script processing may support different "require" areas. For example, it is conceivable that "fileinto" is not supported for a stage other than a user's private script.

The "stop;" command ends the execution of its single containing script, not of scripts in general.

After one script terminates, the next script is executed if an implicit or explicit "keep" is in effect. (To end all script execution, a script should execute "discard; stop;".)

For sieve engines that implement the "variables" extension,

variable state is not carried over between scripts.

For sieve engines that implement the "notify" extension, the "denotify" action in one script can only cancel "notify" actions from that same script.

However, if a sequence of script executions results in the same message sent to the same recipient, or filed to the same destination folder, more than once, an implementation MAY only produce a single message, even if the commands executed stem from multiple scripts.

If a sieve implementation enforces the incompatibility of "reject" with other actions (a SHOULD in [[SIEVE](#)]), it MUST only enforce it within one script; an action in a preceding script MUST be compatible with a "reject" in a later script.

5. Script Errors

When an error occurs during processing of any of the scripts, all message processing stops, and the message is treated as if the final script had executed a "keep;".

6. Security Considerations

A script executed before another script can prevent the trailing script from running (by executing "discard; stop;" or by encountering an error.) This is deliberate.

Corporate filtering of employee e-mail may violate the privacy expectations of employees. However, since these instances are the ones running the software that handles employee e-mail to begin with, they already have the technical capability to do this.

7. Acknowledgments

Thanks to Eric Allman, Will Lee, Dowson Tong, and Chris Markle for comments.

8. Authors' Addresses

Jutta Degener
Sendmail, Inc.
6425 Christie Ave, 4th Floor
Emeryville, CA 94608
Email: jutta@sendmail.com

Rand Wacker
Sendmail, Inc.
6425 Christie Ave, 4th Floor
Emeryville, CA 94608
Email: rand@sendmail.com

9. 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 list at <ietf-mta-filters@imc.org>. Subscription requests can be sent to <ietf-mta-filters-request@imc.org> (send an email message with the word "subscribe" in the body).

More information on the mailing list along with a WWW archive of back messages is available at <<http://www.imc.org/ietf-mta-filters/>>.

9.1 Comparison to [draft-daboo-sieve-include-00](#)

The "include" sieve extension describes a mechanism for naming and combining multiple scripts. This document doesn't do that; how the sequence of scripts to be executed on a message is determined is left up to the environment and likely out of control of the script owner.

The "include" sieve extension creates a hierarchical tree of nested scripts; this extension describes sequential, not hierarchical execution.

The "include" sieve extension defines the semantics of "stop" to stop all sieve execution, not just that of the innermost containing script. It adds a new "return" command to explicitly end execution of a single script. This document specifies that "stop" just stops a single script, and uses the redefined meaning of "keep" to regulate the flow of messages through scripts.

9.2 "require" keyword

This draft started out with a "require" keyword, "multiscript", but since what it describes lies outside the domain of strict sieve language behavior, I'm not sure it needs one.

Appendix A. References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.
- [SIEVE] Showalter, T., "Sieve: A Mail Filtering Language", [RFC 3028](#), January 2001.

Appendix B. Full Copyright Statement

Copyright (C) The Internet Society 2002,2003. 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.