

Sieve Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 6, 2010

A. Melnikov  
Isode Limited  
July 5, 2009

**Sieve Extension: Externally Stored Lists**  
**draft-melnikov-sieve-external-lists-02**

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire on January 6, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

Sieve scripting language can be used for implementing of whitelisting, blacklisting and personal distribution lists. Currently this requires that all members of such lists be hardcoded

in the script itself. Whenever a member of such list is added or deleted, the script needs to be updated and possibly uploaded to a mail server.

This document defines a Sieve extension for accessing externally stored mailing lists, i.e. list whose members are stored externally to the script, for example in LDAP ([RFC 4510](#)), ACAP ([RFC 2244](#)) or a relational database.

#### ToDo

- o Need a way to advertise supported URI schemas in ManageSieve and ihave.

#### Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">1.1.</a>	Conventions used in this document . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Extlists extension . . . . .	<a href="#">3</a>
<a href="#">2.1.</a>	Capability Identifier . . . . .	<a href="#">3</a>
<a href="#">2.2.</a>	:list tagged argument to the 'address', 'header' and 'envelope' tests . . . . .	<a href="#">3</a>
<a href="#">2.3.</a>	:list tagged argument to the 'redirect' action . . . . .	<a href="#">4</a>
<a href="#">2.4.</a>	Syntax of an externally stored list name . . . . .	<a href="#">4</a>
<a href="#">2.5.</a>	Examples . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Security Considerations . . . . .	<a href="#">5</a>
<a href="#">4.</a>	IANA Considerations . . . . .	<a href="#">6</a>
<a href="#">5.</a>	Acknowledgements . . . . .	<a href="#">6</a>
<a href="#">6.</a>	References . . . . .	<a href="#">6</a>
<a href="#">6.1.</a>	Normative References . . . . .	<a href="#">6</a>
<a href="#">6.2.</a>	Informative References . . . . .	<a href="#">7</a>
	Author's Address . . . . .	<a href="#">7</a>

## **1. Introduction**

This document specifies an extension to the Sieve language defined by [\[Sieve\]](#) for checking membership in an externally stored list or for sending messages to a list of recipients stored externally to the Sieve script.

This extension adds a new tagged argument to the "header" and "envelope" tests, and to the "redirect" action [\[Sieve\]](#).

### **1.1. Conventions used in this document**

Conventions for notations are as in [\[Sieve\]](#) [section 1.1](#), including the use of [\[ABNF\]](#).

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 [\[Kwds\]](#).

## **2. Extlists extension**

### **2.1. Capability Identifier**

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

### **2.2. `:list` tagged argument to the 'address', 'header' and 'envelope' tests**

Usage: address [COMPARATOR] [ADDRESS-PART] [MATCH-TYPE] `:list`  
      <header-list: string-list>  
      <key-list: string-list>

Usage: header [COMPARATOR] [MATCH-TYPE] `:list`  
      <header-names: string-list>  
      <key-list: string-list>

Usage: envelope [COMPARATOR] [ADDRESS-PART] [MATCH-TYPE] `:list`  
      <envelope-part: string-list>  
      <key-list: string-list>

The new `:list` tagged argument, if present, changes interpretation of the "key-list" parameter to the 'address'/'header'/'envelope' test to become a list of names of externally stored lists. E.g. the "header" test with the `:list` parameter evaluates to true if the value of any of the named headers, ignoring leading and trailing whitespace, matches any member of one or more externally stored lists

of values with names specified in key-list.

See [Section 2.4](#) for the detailed description of syntax used for naming externally stored lists.

### **2.3. :list tagged argument to the 'redirect' action**

Usage: `redirect :list <ext-list-name: string>`

The "redirect" action with the ":list" argument is used to send the message to one or more email address stored in the externally stored list 'ext-list-name'. This variant of the redirect command can be used to implement a personal distribution list.

See [Section 2.4](#) for the detailed description of syntax used for naming externally stored lists.

### **2.4. Syntax of an externally stored list name**

A name of an externally stored list is always an absolute URI. Implementations might find URL such as [[LDAP](#)], [[CardDAV](#)] URL), or [[TAG-URI](#)] to be useful for naming external lists.

The "tag" URI scheme can be used to represent opaque, but user friendlier identifiers. Resolution of such identifiers is going to be implementation specific and it can help in hiding the complexity of an implementation from end users. For example, an implementation can provide a web interface for managing lists of users stored in LDAP. Requiring users to know generic LDAP URL syntax is not going to be very practical, due to its complexity. However such implementation can use a fixed tag URI prefix such as "tag:example.com,<date>:" (where <date> can be, for example, a date generated once on installation of the web interface and left untouched upon upgrades) instead and the prefix doesn't even need to be shown to end users.

### **2.5. Examples**

Example 1:

```
require ["extlists"];

# Submission from list members is sent to all members
if allof (envelope :detail :list "to"
          "tag:example.com,2009-05-28:mylist",
          header :contains :list "from"
          "tag:example.com,2009-05-28:mylist") {
  redirect :list "tag:example.com,2009-05-28:mylist";
}
```



### 3. Security Considerations

Security considerations related to the "envelope"/"header" tests and "redirect" action discussed in [[Sieve](#)] also apply to this document.

A failure to retrieve data due to the server storing the external list membership being down or otherwise inaccessible may alter the result of Sieve processing. So implementations SHOULD treat a temporary failure to retrieve or verify external list membership in the same manner as a temporary failure to retrieve a Sieve script. For example, if the Sieve script is stored in the Lightweight Directory Access Protocol (LDAP) and the script can't be retrieved when a message is processed, then the agent performing Sieve processing can, for example, assume that the script doesn't exist or delay message delivery until the script can be retrieved successfully. External list memberships should be treated as if they are a part of the script itself, so a temporary failure to retrieve them should be handled in the same way as a temporary failure to retrieve the Sieve script itself.

Protocols/APIs used to retrieve/verify external list membership MUST provide at least the same level of confidentiality as protocols/APIs used to retrieve Sieve scripts. For example, if Sieve scripts are retrieved using LDAP secured with Transport Layer Security (TLS) encryption, then the protocol used to retrieve external list membership must use a comparable mechanism for providing connection confidentiality. In particular, the protocol used to retrieve external list membership must not be lacking encryption.

Implementations of this extensions should keep in mind that matching values against an externally stored list can be IO and/or CPU intensive. This can be used to deny service to the mailserver and/or to servers providing access to externally stored mailing lists. A naive implementation, such as the one that tries to retrieve content of the whole list to perform matching can make this worse. But note that many protocols that can be used for accessing externally stored lists support flexible searching facilities that can be used to minimize network traffic and load on the directory service. For example LDAP allows for search filters.

Many organizations support external lists with thousands of recipients. In order to avoid mailbombs, when redirecting a message to an externally stored mailing list, implementations SHOULD enforce limits on the number of recipients and/or on domains to which such recipients belong.



#### 4. IANA Considerations

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

To: iana@iana.org

Subject: Registration of new Sieve extension

Capability name: extlists

Description: adds the ':list' tagged argument to 'address', 'header' and 'envelope' tests, and to the 'redirect' action. The ':list' argument changes address/header/envelope test to match values against values stored in one or more externally stored list. The ':list' argument to the redirect action changes the redirect action to forward the message to email addresses stored in the externally stored list.

RFC number: this RFC

Contact address:

The Sieve discussion list <ietf-mta-filters@imc.org>

This information should be added to the list of sieve extensions given on <http://www.iana.org/assignments/sieve-extensions>.

#### 5. Acknowledgements

Thanks to Alexandros Vellis, Barry Leiba, Nigel Swinson, Kjetil Torgrim Homme, Dave Cridland, Cyrus Daboo, Pete Resnick for ideas, comments and suggestions.

#### 6. References

##### 6.1. Normative References

- [ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 5234](#), January 2008.
- [Kwds] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.
- [Sieve] Guenther, P. and T. Showalter, "Sieve: An Email Filtering Language", [RFC 5228](#), January 2008.
- [URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.



## **6.2. Informative References**

- [ACAP] Newman, C. and J. Myers, "ACAP -- Application Configuration Access Protocol", [RFC 2244](#), November 1997.
- [CardDAV] Daboo, C., "vCard Extensions to WebDAV (CardDAV)", work in progress, [draft-daboo-carddav](#), May 2007.
- [LDAP] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", [RFC 4510](#), June 2006.
- [TAG-URI] Kindberg, T. and S. Hawke, "The 'tag' URI Scheme", [RFC 4151](#), October 2005.

### Author's Address

Alexey Melnikov  
Isode Limited  
5 Castle Business Village  
36 Station Road  
Hampton, Middlesex TW12 2BX  
UK

Email: [Alexey.Melnikov@isode.com](mailto:Alexey.Melnikov@isode.com)