

## Sieve Extension: Relational Tests

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

### Abstract

This document describes the RELATIONAL extension to the Sieve mail filtering language defined in RFC 3028. This extension extends existing conditional tests in Sieve to allow relational operators. In addition to testing their content, it also allows for testing of the number of entities in header and envelope fields.

## 1 Introduction

Sieve [SIEVE] is a language for filtering e-mail messages at the time of final delivery. It is designed to be implementable on either a mail client or mail server. It is meant to be extensible, simple, and independent of access protocol, mail architecture, and operating system. It is suitable for running on a mail server where users may not be allowed to execute arbitrary programs, such as on black box Internet Messages Access Protocol (IMAP) servers, as it has no variables, loops, nor the ability to shell out to external programs.

The RELATIONAL extension provides relational operators on the address, envelope, and header tests. This extension also provides a way of counting the entities in a message header or address field.

With this extension, the sieve script may now determine if a field is greater than or less than a value instead of just equivalent. One use is for the x-priority field: move messages with a priority greater than 3 to the "work on later" folder. Mail could also be sorted by the from address. Those userids that start with 'a'-'m' go to one folder, and the rest go to another folder.

The sieve script can also determine the number of fields in the header, or the number of addresses in a recipient field. For example: are there more than 5 addresses in the to and cc fields.

## 2 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 BCP 14, RFC 2119.

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

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

## 3 Comparators

This document does not define any comparators or exempt any comparators from the require clause. Any comparator used, other than "i;octet" and "i;ascii-casemap", MUST be declared a require clause as defined in [SIEVE].

The "i;ascii-numeric" comparator, as defined in [ACAP], MUST be supported for any implementation of this extension. The comparator "i;ascii-numeric" MUST support at least 32 bit unsigned integers.

Larger integers MAY be supported. Note: the "i;ascii-numeric" comparator does not support negative numbers.

## 4 Match Type

This document defines two new match types. They are the VALUE match type and the COUNT match type.

The syntax is:

MATCH-TYPE =/ COUNT / VALUE

COUNT = ":count" relational-match

VALUE = ":value" relational-match

relational-match = DQUOTE ( "gt" / "ge" / "lt"  
/ "le" / "eq" / "ne" ) DQUOTE

#### 4.1 Match Type Value

The VALUE match type does a relational comparison between strings.

The VALUE match type may be used with any comparator which returns sort information.

Leading and trailing white space MUST be removed from the value of the message for the comparison. White space is defined as

SP / HTAB / CRLF

A value from the message is considered the left side of the relation. A value from the test expression, the key-list for address, envelope, and header tests, is the right side of the relation.

If there are multiple values on either side or both sides, the test is considered true, if any pair is true.

#### 4.2 Match Type Count

The COUNT match type first determines the number of the specified entities in the message and does a relational comparison of the number of entities to the values specified in the test expression.

The COUNT match type SHOULD only be used with numeric comparators.

The Address Test counts the number of recipients in the specified fields. Group names are ignored.

The Envelope Test counts the number of recipients in the specified envelope parts. The envelope "to" will always have only one entry, which is the address of the user for whom the sieve script is running. There is no way a sieve script can determine if the message was actually sent to someone else using this test. The envelope "from" will be 0 if the MAIL FROM is blank, or 1 if MAIL FROM is not blank.

The Header Test counts the total number of instances of the specified fields. This does not count individual addresses in the "to", "cc", and other recipient fields.

In all cases, if more than one field name is specified, the counts for all specified fields are added together to obtain the number for comparison. Thus, specifying ["to", "cc"] in an address COUNT test, comparing the total number of "to" and "cc" addresses; if separate counts are desired, they must be done in two comparisons, perhaps joined by "allof" or "anyof".

## 5 Security Considerations

Security considerations are discussed in [SIEVE].

An implementation **MUST** ensure that the test for envelope "to" only reflects the delivery to the current user. It **MUST** not be possible for a user to determine if this message was delivered to someone else using this test.

## 6 Example

Using the message:

```
received: ...
received: ...
subject: example
to: foo@example.com.invalid, baz@example.com.invalid
cc: qux@example.com.invalid
```

The test:

```
address :count "ge" :comparator "i;ascii-numeric" ["to", "cc"]
["3"]
```

would be true and the test

```
anyof ( address :count "ge" :comparator "i;ascii-numeric"
          ["to"] ["3"],
        address :count "ge" :comparator "i;ascii-numeric"
          ["cc"] ["3"] )
```

would be false.

To check the number of received fields in the header, the following test may be used:

```
header :count "ge" :comparator "i;ascii-numeric"
["received"] ["3"]
```

This would return false. But

```
header :count "ge" :comparator "i;ascii-numeric"
["received", "subject"] ["3"]
```

would return true.

The test:

```
header :count "ge" :comparator "i;ascii-numeric"
["to", "cc"] ["3"]
```

will always return false on an RFC 2822 compliant message [RFC2822], since a message can have at most one "to" field and at most one "cc" field. This test counts the number of fields, not the number of addresses.

## 7 Extended Example

```
require ["relational", "comparator-i;ascii-numeric"];

if header :value "lt" :comparator "i;ascii-numeric"
["x-priority"] ["3"]
{
    fileinto "Priority";
}

elseif address :count "gt" :comparator "i;ascii-numeric"
["to"] ["5"]
{
    # everything with more than 5 recipients in the "to" field
    # is considered SPAM
    fileinto "SPAM";
}

elseif address :value "gt" :all :comparator "i;ascii-casemap"
["from"] ["M"]
{
    fileinto "From N-Z";
} else {
    fileinto "From A-M";
}

if allof ( address :count "eq" :comparator "i;ascii-numeric"
["to", "cc"] ["1"] ,
address :all :comparator "i;ascii-casemap"
["to", "cc"] ["me@foo.example.com.invalid"]
{
    fileinto "Only me";
}
```

## 8 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: RELATIONAL  
Capability keyword: relational  
Capability arguments: N/A  
Standards Track/IESG-approved experimental RFC number: this RFC  
Person and email address to contact for further information:  
Wolfgang Segmuller  
IBM T.J. Watson Research Center  
30 Saw Mill River Rd  
Hawthorne, NY 10532

Email: whs@watson.ibm.com

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

## 9 References

### 9.1 Normative References

- [SIEVE] Showalter, T., "Sieve: A Mail Filtering Language", RFC 3028, January 2001.
- [Keywords] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [ABNF] Crocker, D., "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

### 9.2 Non-Normative References

- [ACAP] Newman, C. and J. G. Myers, "ACAP -- Application Configuration Access Protocol", RFC 2244, November 1997.

## 10 Author's Address

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

EMail: whs@watson.ibm.com

## 11 Full Copyright Statement

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

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.



