

Sieve Email Filtering:
Delivery Status Notifications and Deliver-By Extensions

Abstract

This document describes the "envelope-dsn", "redirect-dsn", "envelope-deliverby", and "redirect-deliverby" extensions to the Sieve email filtering language. The "envelope-dsn" and "envelope-deliverby" extensions provide access to additional envelope information provided by the delivery status notification (DSN) and Deliver-By SMTP extensions, respectively. The "redirect-dsn" and "redirect-deliverby" extensions extend Sieve's redirect action to provide control over delivery status notification and Deliver-By parameters, respectively.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6009>.

Copyright Notice

Copyright (c) 2010 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 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	3
3. Capability Identifiers	4
4. Envelope-dsn Extension	4
4.1. Examples	5
5. Envelope-deliverby Extension	6
5.1. Examples	7
6. redirect-dsn Extension	9
6.1. MAIL FROM Address Selection	9
6.2. Example	9
7. redirect-deliverby Extension	10
7.1. MAIL FROM Address Selection	11
7.2. Example	11
8. Security Considerations	11
9. IANA Considerations	12
10. References	13
10.1. Normative References	13
10.2. Informative References	14
Appendix A. Acknowledgements	15

1. Introduction

Sieve [RFC5228] is a language for filtering email messages at or around the time of final delivery. It is designed to be implementable on either a mail client or mail server. 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 Message Access Protocol [RFC3501] servers, as it has no user-controlled loops or the ability to run external programs.

The base Sieve specification defines the envelope extension and test to access information in the message envelope. Only information available in regular SMTP [RFC5321] is provided; additional information added to the SMTP envelope by SMTP extensions cannot be accessed.

The "envelope-dsn" extension extends the envelope test to allow access to the additional envelope fields defined by the SMTP extension for delivery status notifications (DSNs) specified in RFC 3461 [RFC3461]. The "envelope-deliverby" extension extends the envelope test to allow access to the additional envelope fields defined by the Deliver-By SMTP extension defined in [RFC2852].

The base Sieve specification also defines the redirect action, which sends the message to a different address. Redirect only allows specification of the new recipient address. The "redirect-dsn" extension extends redirect to allow specification of some fields defined by the delivery status notification SMTP extension. "redirect-deliverby" in turn provides the ability to set a time limit for delivery as specified in RFC 2852 [RFC2852].

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 [RFC2119].

The terms used to describe the various components of the Sieve language are taken from Section 1.1 of [RFC5228]. The nature and handling of Sieve errors are described in Section 2.10.6 of [RFC5228].

This document uses the ABNF notation specified in [RFC5234], and refers to the notify-esmtp-value ABNF production defined in Section 4.1 of [RFC3461].

3. Capability Identifiers

The capability strings associated with the extensions defined in this document are "envelope-dsn", "redirect-dsn", "envelope-deliverby", and "redirect-deliverby".

4. Envelope-dsn Extension

The "envelope-dsn" extension does not define any new tests or actions; rather, it adds four values to the list of possible (case-insensitive) envelope-part strings defined in [Section 5.4 of \[RFC5228\]](#):

notify - Match the list of notification conditions, or NOTIFY values, associated with the TO address used in the SMTP RCPT TO command that resulted in this message getting delivered to this user. More than one notification condition can be in effect at once; each condition that is in effect is tested separately, and any match causes the test to succeed. The syntax and semantics of the NOTIFY parameter are defined in [Section 4.1 of RFC 3461 \[RFC3461\]](#). Currently, the possible notification condition values are "NEVER", "SUCCESS", "FAILURE", and "DELAY". Note that the value "NEVER" is never combined with any other value.

orcpt - Match the original recipient, or ORCPT, value associated with the TO address used in the SMTP RCPT TO command that resulted in this message getting delivered to this user, with xtext encoding removed. The syntax and semantics of the ORCPT parameter are defined in [Section 4.2 of RFC 3461 \[RFC3461\]](#).

ret - Match the return of content, or RET, value given in the SMTP MAIL FROM command. The syntax and semantics of the RET parameter are defined in [Section 4.3 of RFC 3461 \[RFC3461\]](#). The possible return of content values are "FULL" and "HDRS".

envid - Match the envelope identifier, or ENVID, value in decoded form given in the SMTP MAIL FROM command. The syntax and semantics of the ENVID parameter are defined in [Section 4.4 of RFC 3461 \[RFC3461\]](#).

The envelope test fails unconditionally for each of these envelope-part strings if the specified envelope parameter does not exist for the current message or recipient.

The envelope test's ADDRESS-PART argument assumes the string being tested has the syntax of an email address. None of the new envelope parts defined here have address syntax; accordingly, it is an error to specify an ADDRESS-PART argument in conjunction with these new envelope parts.

The "relational" extension [RFC5231] adds a match type called ":count". The count of an envelope test with an envelope-part of "orcpt", "ret", and "envid" is 1 if the corresponding SMTP parameter is present and 0 otherwise. The count of an envelope test with an envelope-part of "notify" is equal to the number of notification conditions specified and 0 if the NOTIFY parameter is not present.

4.1. Examples

The fact that the NOTIFY envelope parameter is multivalued and the notify envelope-part turns this into a list of values makes it easy to check to see if a given value is present without having to worry about other values:

```
require ["envelope", "envelope-dsn"];

# Check whether SUCCESS notifications were requested,
# irrespective of any other requests that were made
if envelope "notify" "SUCCESS"
{
    # do whatever
}
```

Checking to see if a given request is the only one present is a little trickier, however:

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

# Check whether only FAILURE notifications were requested
if allof ( envelope "notify" "FAILURE",
            envelope :comparator "i;ascii-numeric"
                      :count "eq" "notify" "1"
          )
{
    # do whatever
}
```

The orcpt envelope-part always contains an address type indicator prefix in addition to an address, which must be taken into account in any tests:

```
require ["envelope", "envelope-dsn"];

# See if the orcpt is an RFC822 address in the example.com
# domain
if envelope :matches "orcpt" "rfc822;*@example.com"
{
    # do whatever
}
```

5. Envelope-deliverby Extension

The "envelope-deliverby" extension does not define any new tests or actions; rather, it adds four values to the list of possible (case-insensitive) envelope-part strings defined in [Section 5.4 of \[RFC5228\]](#) and an optional :zone tagged argument. This updates the usage description for envelope to:

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

These new envelope parts correspond to the new MAIL FROM parameters defined in [Section 4 of \[RFC2852\]](#). They are:

bytimeabsolute - Match the current value of the initial integer part of the Deliver-By extension's BY parameter on the SMTP MAIL FROM command, converted into an absolute time represented in restricted ISO 8601 format. The restricted ISO 8601 format is specified by the date-time ABNF production given in [\[RFC3339\]](#), [Section 5.6](#), with the added restrictions that the letters "T" and "Z" MUST be in upper case, and a time zone offset of zero MUST be represented by "Z" and not "+00:00".

bytimerelative - Match the current value of the initial integer part of the Deliver-By extension's BY parameter specified in the SMTP MAIL FROM command.

bymode - Match a string computed from the by-mode part of the Deliver-By extension's BY parameter. The possible values are "notify" and "return", which correspond to the BY parameter mode specifier characters "N" and "R", respectively.

bytrace - Match the trace modifier computed from the by-trace modifier on the Deliver-By extension's BY parameter. The possible values are "trace" and "" (the empty string). These values correspond to the presence or absence of the by-trace "T" modifier, respectively.

The envelope test fails unconditionally for each of these envelope-part strings if the BY SMTP MAIL FROM parameter does not exist for the current message or recipient.

The new :zone argument specifies a time zone offset string that any bytimeabsolute value is to be shifted to prior to testing. :zone has no effect on envelope-parts other than bytimeabsolute. The value of the time zone offset string MUST be an offset relative to UTC with the following syntax:

```
time-zone = ( "+" / "-" ) 4DIGIT
```

The "+" or "-" indicates whether the time-of-day is ahead of (i.e., east of) or behind (i.e., west of) UTC. The first two digits indicate the number of hours difference from Universal Time, and the last two digits indicate the number of minutes difference from Universal Time. Note that this agrees with the [RFC5322] format for time zone offsets, not the ISO 8601 format. The local time zone MUST be used for bytimeabsolute if the :zone argument is omitted.

The envelope test's ADDRESS-PART argument assumes the string being tested has the syntax of an email address. None of the new envelope parts defined here have address syntax; accordingly, it is an error to specify an ADDRESS-PART argument in conjunction with these new envelope parts.

The "relational" extension [RFC5231] adds a match type called ":count". The count of an envelope test with an envelope-part of "bytime", "bymode", and "bytrace" is 1 if the BY parameter is present and 0 otherwise.

It is important to note that the Deliver-By by-time is decremented as the message passes through the transport infrastructure. Accordingly, it is not possible to tell what the message originator set the value to; only the amount of time remaining at the moment the sieve is run can be determined. Additionally, note that bytimerelative values can be negative, making it necessary to either perform additional checks or else use a comparator that, unlike i;ascii-numeric, is capable of handling signed integers.

5.1. Examples

As noted above, this extension does not provide access to the originator's initial by-time setting for the simple reason that this information is not part of the envelope. It can, however, be used to check and see if the message was delivered within the allotted time. Note the additional check to see if the value is negative:

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

# Check to see if this message didn't make it in the time allotted by
# the originator.
if anyof (envelope :contains "bytimerelative" "-",
        envelope :value "eq" :comparator "i;ascii-numeric"
        "bytimerelative" "0")
{
    # do whatever
}
```

This operation can be done more simply if the date [RFC5260] and variables [RFC5229] extensions are available:

```
require ["envelope", "envelope-deliverby", "relational", "date",
        "variables"];

# Check to see if this message didn't make it in the time allotted by
# the originator.
if currentdate :matches "iso8601" "*" {
    set "cdate" "${0}";
    if envelope :value "ge" "bytimeabsolute" "${cdate}" {
        # do whatever
    }
}
```

Note that there is no need to force the use of a particular time zone since both currentdate and the bytimeabsolute value are required to default to the local time zone. A similar check could be written using :zone if the action taken depends on having the by-time represented in a particular zone:

```
require ["envelope", "envelope-deliverby", "relational", "date",
        "variables"];

# If the message didn't make it in time, file it according to when it
# should have been received
if envelope :matches :zone "+0000" "bytimeabsolute" "*T*:*:*" {
    set "bdate" "${0}";
    set "bhour" "${2}";
    if currentdate :zone "+0000" :value "lt" "iso8601" "${bdate}")
        fileinto "missed-${bhour}";
    }
}
```


6. redirect-dsn Extension

The "redirect-dsn" extension does not define any new tests or actions; rather, it adds two new arguments, NOTIFY and RET, to the redirect action defined in [Section 4.2 of \[RFC5228\]](#). This updates the usage description for redirect to:

```
Usage:  redirect [:notify "value"] [:ret "FULL"|"HDRS"]
        <address: string>
```

The syntax for the NOTIFY and RET arguments are:

```
NOTIFY = ":notify" notify-value
notify-value = DQUOTE ("NEVER" / notify-esmtp-list) DQUOTE
notify-esmtp-list = notify-list-element *("," notify-list-element)
```

```
RET = ":ret" ret-value
ret-value = DQUOTE ("FULL" / "HDRS") DQUOTE
```

The notify-list-element ABNF production is defined in [Section 4.1 of \[RFC3461\]](#).

When these arguments are specified, they set the corresponding NOTIFY ESMTP RCPT TO and RET ESMTP MAIL FROM parameters, respectively. These arguments are only honored if the delivery status notification (DSN) ESMTP extension is available. When the DSN extension is not available, these arguments MUST be ignored and MUST NOT cause an error.

6.1. MAIL FROM Address Selection

[RFC 5228](#) does not require that any particular envelope sender address be associated with redirected messages. However, the redirect-dsn extension isn't terribly useful if the place where the delivery status notifications are sent isn't known. Accordingly, when either :notify or :ret is specified and the envelope sender address isn't empty, implementations MUST set the envelope sender address to the address of the sieve owner.

6.2. Example

One possible use of :notify on redirect is to combine the copy extension [\[RFC3894\]](#) with the ability to suppress nondelivery notifications to generate a private copy of selected messages with no side effects or error notifications:

```
require ["copy", "redirect-dsn"];

# Make a private copy of messages from user@example.com
if address "from" "user@example.com"
{
    redirect :copy :notify "NEVER" "elsewhere@example.com";
}
```

7. redirect-deliverby Extension

The "redirect-deliverby" extension does not define any new tests or actions; rather, it adds three new arguments, BYTIME, BYMODE, and BYTRACE, to the redirect action defined in [Section 4.2 of \[RFC5228\]](#). This updates the usage description for redirect to:

```
Usage:  redirect [:bytimerelative <rlimit: number> /
           :bytimeabsolute <alimit:string>
           [:bymode "notify"|"return"] [:bytrace]]
           <address: string>
```

:bytimerelative specifies the number of seconds within which the message should be delivered. This parameter does not allow specification of negative values; it should not be necessary to specify such values in this context. :bytimeabsolute specifies an absolute time limit on delivery. The limit in this case is specified in the restricted ISO 8601 format specified by the date-time ABNF production given in [\[RFC3339\]](#).

:bymode specifies whether a notification should be sent or the message simply returned if the time limit is exceeded. The default is "return" if :bymode is not specified. :bytrace, if specified, activates message tracing.

The semantics of delivery time limits and these parameters are specified and discussed at length in [\[RFC2852\]](#).

It is an error to specify either :bymode or :bytrace without either :bytimeabsolute or :bytimerelative.

When these arguments are specified, they are used to construct the corresponding BY ESMTPL MAIL FROM parameter. The :bytimeabsolute or :bytimerelative value becomes the by-time, the :bymode becomes the by-mode value, and :bytrace sets the by-trace modifier. If the Deliver-By extension is unavailable, the handling of the redirected message MUST conform to the semantics specified in [\[RFC2852\]](#), [Section 4.1.4](#) for relaying to a server that does not support the Deliver-By SMTP extension.

7.1. MAIL FROM Address Selection

RFC 5228 does not require that any particular envelope sender address be associated with redirected messages. However, the `redirect-deliverby` extension, like the `redirect-dsn` extension, isn't terribly useful if the place where any delivery status notifications are sent isn't known. Accordingly, when either `:bytimeabsolute` or `:bytimerelative` is specified and the envelope sender address isn't empty, implementations MUST set the envelope sender address to the address of the sieve owner.

7.2. Example

The obvious use of `"redirect-deliverby"` is to specify a limit on delivery attempts for a redirected message:

```
require ["copy", "redirect-deliverby"];

# Send a copy to my cell phone, time out after 10 minutes
if address "from" "user@example.com"
{
    redirect :copy :bytimerelative 600 "cellphone@example.com";
}
```

Limits on delivery after a particular time of day may also be constructed:

```
require ["copy", "redirect-deliverby", "date", "variables",
        "relational", "comparator-i;ascii-numeric"];

# Send a copy to my cell phone to be delivered before 10PM
if currentdate :value "lt"
    :comparator "i;ascii-numeric" "hour" "22"
{
    if currentdate :matches "date" "*" {set "date" "${0}";}
    if currentdate :matches "zone" "*" {set "zone" "${0}";}
    redirect :copy :bytimeabsolute "${date}T20:00:00${zone}"
        :bymode "return" "cellphone@example.com";
}
```

8. Security Considerations

The `envelope-dsn` and `envelope-deliverby` extensions provide access to additional message envelope information. This is not believed to raise any additional security issues beyond those for the Sieve `"envelope"` test.

The redirect-dsn extension allows specification of the delivery status notification's NOTIFY parameter, which can cause the generation of notification messages that might otherwise not be generated, especially if notification in the event of successful delivery is required. Sites that limit the ability to request success notifications will also need to restrict the ability to request them using the redirect-dsn extension.

Similarly, the redirect-deliverby extension is used to control how long the transport infrastructure will continue to attempt to deliver a message before giving up, which could result in the generation of additional notification messages. While the underlying Deliver-By extension does have a minimum by-time limit, sites may wish to impose additional limits on the minimum by-time allowed in a redirect action.

All of the security considerations given in the base Sieve specification also apply to this extension.

9. 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 extensions

Capability name: envelope-dsn
Description: The "envelope-dsn" extension extends the envelope test to allow checking of information associated with the DSN ESMTPE extension defined in [RFC 3461](#).
RFC number: [RFC 6009](#)
Contact address: Sieve discussion list <sieve@ietf.org>

Capability name: envelope-deliverby
Description: The "envelope-deliverby" extension extends the envelope test to allow checking of information associated with the Deliver-By ESMTPE extension defined in [RFC 2852](#).
RFC number: [RFC 6009](#)
Contact address: Sieve discussion list <sieve@ietf.org>

Capability name: redirect-dsn

Description: The "redirect-dsn" extension extends the redirect action to allow specification of the NOTIFY and RET ESMTP parameters associated with the DSN SMTP extension defined in [RFC 3461](#).

RFC number: [RFC 6009](#)

Contact address: Sieve discussion list <sieve@ietf.org>

Capability name: redirect-deliverby

Description: The "redirect-deliverby" extension extends the redirect action to allow specification of the BY ESMTP parameter associated with the Deliver-By SMTP extension defined in [RFC 2852](#).

RFC number: [RFC 6009](#)

Contact address: Sieve discussion list <sieve@ietf.org>

This information has been added to the list of Sieve extensions available from <http://www.iana.org>.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2852] Newman, D., "Deliver By SMTP Service Extension", [RFC 2852](#), June 2000.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", [RFC 3339](#), July 2002.
- [RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", [RFC 3461](#), January 2003.
- [RFC5228] Guenther, P. and T. Showalter, "Sieve: An Email Filtering Language", [RFC 5228](#), January 2008.
- [RFC5231] Segmuller, W. and B. Leiba, "Sieve Email Filtering: Relational Extension", [RFC 5231](#), January 2008.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", [RFC 5321](#), October 2008.

- [RFC5322] Resnick, P., Ed., "Internet Message Format", [RFC 5322](#), October 2008.

10.2. Informative References

- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", [RFC 3501](#), March 2003.
- [RFC3894] Degener, J., "Sieve Extension: Copying Without Side Effects", [RFC 3894](#), October 2004.
- [RFC5229] Homme, K., "Sieve Email Filtering: Variables Extension", [RFC 5229](#), January 2008.
- [RFC5260] Freed, N., "Sieve Email Filtering: Date and Index Extensions", [RFC 5260](#), July 2008.

Appendix A. Acknowledgements

Cyrus Daboo, Derek Diget, Philip Guenther, Arnt Gulbrandsen, Tero Kivinen, Barry Leiba, Andrew McKeon, Alexey Melnikov, Chris Newman, Aaron Stone, and Alexandros Vellis provided helpful suggestions and corrections.

Author's Address

Ned Freed
Oracle
800 Royal Oaks
Monrovia, CA 91016-6347
USA

EMail: ned.freed@mrochek.com