Sieve Notification Mechanism: xmpp
draft-ietf-sieve-notify-xmpp-09

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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/1id-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 August 22, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

This document describes a profile of the Sieve extension for notifications, to allow notifications to be sent over the Extensible Messaging and Presence Protocol (XMPP), also known as Jabber.
1. Introduction

1.1. Overview

The [NOTIFY] extension to the [SIEVE] mail filtering language is a framework for providing notifications by employing URIs to specify the notification mechanism. This document defines how xmpp URIs (see [XMPP-URI]) are used to generate notifications via the Extensible Messaging and Presence Protocol [XMPP], which is widely implemented in Jabber instant messaging technologies.

1.2. Terminology

This document inherits terminology from [NOTIFY], [SIEVE], and [XMPP].

The capitalized key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [TERMS].

2. Definition

2.1. Notify parameter "method"

The "method" parameter MUST be a URI that conforms to the xmpp URI scheme (as specified in [XMPP-URI]) and that identifies an XMPP account associated with the email inbox. The URI MAY include the resource identifier of an XMPP address and/or the query component portion of an XMPP URI, but SHOULD NOT include an authority component or fragment identifier component. The processing application MUST extract an XMPP address from the URI in accordance with the processing rules specified in [XMPP-URI]. The resulting XMPP address MUST be encapsulated in XMPP syntax as the value of the XMPP 'to' attribute.

2.2. Test notify_method_capability

In response to a notify_method_capability test for the "online" notification-capability, an implementation SHOULD return a value of "yes" if it has knowledge of an active presence session (see [XMPP-IM]) for the specified XMPP notification-uri; otherwise it SHOULD return a value of "maybe" (since typical XMPP systems may not allow a Sieve engine to gain knowledge about the presence of XMPP entities).
2.3. Notify tag ":from"

If included, the ":from" tag MUST be an electronic address that conforms to the "Mailbox" rule defined in [RFC2821]. The value of the ":from" tag MAY be included in the human-readable XML character data of the XMPP notification; alternatively or in addition, it MAY be transformed into formal XMPP syntax, in which case it MUST be encapsulated as the value of an XMPP Stanza Headers and Internet Metadata [SHIM] header named "Resent-From".

2.4. Notify tag ":importance"

The ":importance" tag has no special meaning for this notification mechanism, and this specification puts no restriction on its use. The value of the ":importance" tag MAY be transformed into XMPP syntax (in addition to or instead of including appropriate text in the XML character data of the XMPP <body/> element); if so, it SHOULD be encapsulated as the value of an XMPP Stanza Headers and Internet Metadata [SHIM] header named "Urgency", where the XML character of that header is "high" if the value of the ":importance" tag is "1", "medium" if the value of the ":importance" tag is "2", and "low" if the value of the ":importance" tag is "3".

2.5. Notify tag ":message"

If the ":message" tag is included, that string MUST be transformed into the XML character data of an XMPP <body/> element (where the string is generated according to the guidelines specified in Section 3.6 of [NOTIFY]).

2.6. Notify tag ":options"

The ":options" tag has no special meaning for this notification mechanism. Any handling of this tag is the responsibility of an implementation.

2.7. XMPP syntax

The xmpp mechanism results in the sending of an XMPP message to notify a recipient about an email message. The general XMPP syntax is as follows:

- The notification MUST be an XMPP <message/> stanza.
- The value of the XMPP 'from' attribute SHOULD be the XMPP address of the notification service associated with the Sieve engine or the XMPP address of the entity to be notified. The value of the XMPP 'from' attribute MUST NOT be generated from the Sieve ":from" tag.
o The value of the XMPP 'to' attribute MUST be the XMPP address specified in the XMPP URI contained in the "method" notify parameter.

o The value of the XMPP 'type' attribute MUST be 'headline' or 'normal'.

o The XMPP <message/> stanza MUST include a <body/> child element. If the "message" tag is included in the Sieve script, that string MUST be used as the XML character data of the <body/> element. If not and if the XMPP URI contained in the "method" notify parameter specified a "body" parameter in the query component, that value SHOULD be used. Otherwise the XML character data SHOULD be some configurable text indicating that the message is a Sieve notification.

o The XMPP <message/> stanza MAY include a <subject/> child element. If the XMPP URI contained in the "method" notify parameter specified a "subject" parameter in the query component, that value SHOULD be used as the XML character data of the <subject/> element. Otherwise the XML character data SHOULD be some configurable text indicating that the message is a Sieve notification.

o The XMPP <message/> stanza SHOULD include a URI for the recipient to use as a hint in locating the message, encapsulated as the XML character data of a <url/> child element of an <x/> element qualified by the 'jabber:x:oob' namespace as specified in [OOB]. If included, the URI SHOULD be an Internet Message Access Protocol [IMAP] URL that specifies the location of the message as defined in [IMAP-URL], but MAY be another URI type that can specify or hint at the location of an email message, such as a URI for an HTTP resource [HTTP] or a POP3 mailbox [POP-URL] at which the message can be accessed. It is not expected that an XMPP user agent shall directly handle such a URI, but instead that it shall invoke an appropriate helper application to handle the URI.

o The XMPP <message/> stanza MAY include an XMPP Stanza Headers and Internet Metadata [SHIM] header named "Resent-From". If the Sieve script included a ":from" tag, the "Resent-From" value MUST be the value of the ":from" tag; otherwise the "Resent-From" value SHOULD be the envelope recipient address of the original email message that triggered the notification.

3. Examples

In the following examples, the sender of the email has an address of <mailto:juliet@example.org>, the entity to be notified has an email address of <mailto:romeo@example.com> and an XMPP address of romeo@im.example.com (resulting in an XMPP URI of <xmpp:romeo@im.example.com>), and the notification service associated with the Sieve engine has an XMPP address of notify.example.com.
3.1. Basic action

The following is a basic Sieve notify action with only a method. The XML character data of the XMPP <body/> and <subject/> elements are therefore generated by the Sieve engine based on configuration. In addition, the Sieve engine includes a URI pointing to the message.

Basic action (Sieve syntax)

```plaintext
notify "xmpp:romeo@im.example.com"
```

The resulting XMPP <message/> stanza might be as follows.

Basic action (XMPP syntax)

```xml
<message from='notify.example.com'
to='romeo@im.example.com'
type='headline'
xml:lang='en'>
  <subject>SIEVE</subject>
  <body>&lt;juliet@example.com&gt; You got mail.&lt;/body&gt;
  <x xmlns='jabber:x:oob'>
    <url>
      imap://romeo@example.com/INBOX;UIDVALIDITY=385759043;/UID=18
    </url>
  </x>
</message>
```

3.2. Action with body

The following action contains a "body" parameter in the query component of the XMPP URI but no ":message" tag in the Sieve script. As a result, the XML character data of the XMPP <body/> element in the XMPP notification is taken from the XMPP URI. In addition, the Sieve engine includes a URI pointing to the message.

Action with body (Sieve syntax)

```plaintext
notify "xmpp:romeo@im.example.com?message
;body=Wherefore%20art%20thou%3F"
```

The resulting XMPP <message/> stanza might be as follows.
Action with body (XMPP syntax)

```xml
<message from='notify.example.com'
to='romeo@im.example.com'
type='headline'
xml:lang='en'>
  <subject>SIEVE</subject>
  <body>Wherefore art thou?</body>
  <x xmlns='jabber:x:oob'>
    <url>
      imap://romeo@example.com/INBOX;UIDVALIDITY=385759044;/UID=19
    </url>
  </x>
</message>
```

3.3. Action with body, importance, message, and subject

The following action specifies an "importance" tag and a "message" tag in the Sieve script, as well as a "body" parameter and a "subject" parameter in the query component of the XMPP URI. As a result, the "message" tag from the Sieve script overrides the "body" parameter from the XMPP URI when generating the XML character data of the XMPP <body/> element. In addition, the Sieve engine includes a URI pointing to the message.

Action with body, importance, message, and subject (Sieve syntax)

```
notify :importance "1"
  :message "Contact Juliet immediately!"
  "xmpp:romeo@im.example.com?message
    ;body=You%27re%20in%20trouble
    ;subject=ALERT%21"
```

The resulting XMPP <message/> stanza might be as follows.
Action with body, importance, message, and subject (XMPP syntax)

<message from='notify.example.com'
to='romeo@im.example.com'
type='headline'
xm:lang='en'>
<subject>ALERT!</subject>
<body>Contact Juliet immediately!</body>
<br />headers xmlns='http://jabber.org/protocol/shim'>
<header name='Urgency'>high</header>
</headers>
<x xmlns='jabber:x:oob'>
<ul>
  <url>
    imap://romeo@example.com/INBOX;UIDVALIDITY=385759045;/UID=20
  </url>
</x>
</message>

3.4. Action with from, message, importance, body, and subject

The following action specifies a ":from" tag, an ":importance" tag, and a ":message" tag in the Sieve script, as well as a "body" parameter and a "subject" parameter in the query component of the XMPP URI. As a result, the ":message" tag from the Sieve script overrides the "body" parameter from the XMPP URI when generating the XML character data of the XMPP <body/> element. In addition, the Sieve engine includes a URI pointing to the message, as well as an XMPP Stanza Headers and Internet Metadata [SHIM] header named "Resent-From" (which encapsulates the value of the ":from" tag).

Action with from, body, importance, message, and subject (Sieve syntax)

notify :from "romeo.my.romeo@example.com"
 :importance "1"
 :message "Contact Juliet immediately!"
 "xmpp:romeo@im.example.com?message
 ;body=You%27re%20in%20trouble
 ;subject=ALERT%21"

The resulting XMPP <message/> stanza might be as follows.
Action with body, from, importance, message, and subject (XMPP syntax)

<message from='notify.example.com' 
  to='romeo@im.example.com' 
  type='headline' 
  xml:lang='en'>
  <subject>ALERT!</subject>
  <body>Contact Juliet immediately!</body>
  <headers xmlns='http://jabber.org/protocol/shim'>
    <header name='Resent-From'>romeo.my.romeo@example.com</header>
    <header name='Urgency'>high</header>
  </headers>
  <x xmlns='jabber:x:oob'>
    <url>
      imap://romeo@example.com/INBOX;UIDVALIDITY=385759045;/UID=21
    </url>
  </x>
</message>

4. Requirements Conformance

Section 3.8 of [NOTIFY] specifies a set of requirements for Sieve notification methods. The conformance of the xmpp notification mechanism is provided here.

1. An implementation of the xmpp notification method SHOULD NOT modify the final notification text (e.g., to limit the length); however, a given deployment MAY do so (e.g., if recipients pay per character or byte for XMPP messages). Modification of characters themselves should not be necessary, since XMPP character data is encoded in [UTF-8].

2. An implementation MAY ignore parameters specified in the ":from", ":importance", and ":options" tags.

3. There is no recommended default message for an implementation to include if the ":message" argument is not specified.

4. A notification sent via the xmpp notification method MAY include a timestamp in the textual message.

5. The value of the XMPP 'from' attribute MUST be the XMPP address of the notification service associated with the Sieve engine. The value of the Sieve "from" tag MAY be transformed into the value of an XMPP Stanza Headers and Internet Metadata [SHIM] header named "Reply-To".

6. The value of the XMPP 'to' attribute MUST be the XMPP address specified in the XMPP URI contained in the "method" parameter.
7. In accordance with [XMPP-URI], an implementation MUST ignore any URI action or parameter it does not understand (i.e., the URI MUST be processed as if the action or parameter were not present). It is RECOMMENDED to support the XMPP "message" query type (see [QUERIES]) and the associated "body" and "subject" parameters, which parameters SHOULD be mapped to the XMPP <body/> and <subject/> child elements of the XMPP <message/> stanza, respectively. However, if included then the Sieve notify ":message" parameter MUST be mapped to the XMPP <body/> element, overriding the "body" parameter (if any) included in the XMPP URI.

8. An implementation MUST NOT include any other extraneous information not specified in parameters to the notify action.

9. In response to a notify_method_capability test for the "online" notification-capability, an implementation SHOULD return a value of "yes" if it has knowledge of an active presence session (see [XMPP-IM]) for the specified XMPP notification-uri but only if the entity that requested the test is authorized to know the presence of the associated XMPP entity (e.g., via explicit presence subscription as specified in [XMPP-IM]); otherwise it SHOULD return a value of "maybe" (since typical XMPP systems may not allow a Sieve engine to gain knowledge about the presence of XMPP entities).

10. An implementation SHOULD NOT attempt to retry delivery of a notification if it receives an XMPP error of type "auth" or "cancel", MAY attempt to retry delivery if it receives an XMPP error of type "wait", and MAY attempt to retry delivery if it receives an XMPP error of "modify" but only if it makes appropriate modifications to the notification (see [XMPP]); in any case the number of retries SHOULD be limited to a configurable number no less than 3 and no more than 10. An implementation MAY throttle notifications if the number of notifications within a given time period becomes excessive according to local service policy. Duplicate suppression (if any) is a matter of implementation and is not specified herein.

5. Internationalization Considerations

Although an XMPP address may contain nearly any [UNICODE] character, the value of the "method" parameter MUST be a Uniform Resource Identifier (see [URI]) rather than an Internationalized Resource Identifier (see [IRI]). The rules specified in [XMPP-URI] MUST be followed when generating XMPP URIs.

In accordance with Section 13 of RFC 3920, all data sent over XMPP MUST be encoded in [UTF-8].
6. Security Considerations

Depending on the information included, sending a notification can be comparable to forwarding mail to the notification recipient. Care must be taken when forwarding mail automatically, to ensure that confidential information is not sent into an insecure environment. In particular, implementations MUST conform to the security considerations given in [NOTIFY], [SIEVE], and [XMPP].

[NOTIFY] specifies that a notification method MUST provide mechanisms for avoiding notification loops. One type of notification loop can be caused by message forwarding; however, such loops are prevented because XMPP does not support forwarding of messages from one XMPP address to another. Another type of notification loop can be caused by auto-replies to XMPP messages received by the XMPP notification service associated with the Sieve engine; therefore such a service MUST NOT auto-reply to XMPP messages it receives.

A common use case might be for a user to create a script that enables the Sieve engine to act differently if the user is currently available at a particular type of service (e.g., send notifications to the user’s XMPP address if the user has an active session at an XMPP service). Whether the user is currently available can be determined by means of a notify_method_capability test for the "online" notification-capability. In XMPP, information about current network availability is called "presence" (see also [MODEL]). Since [XMPP-IM] requires that a user must approve a presence subscription before an entity can gain access to the user’s presence information, a limited but reasonably safe implementation might be for the Sieve engine to request a subscription to the user’s presence. The user would then need to approve that subscription request so that the Sieve engine can act appropriately depending on whether the user is online or offline. However, the Sieve engine MUST NOT use the user’s presence information when processing scripts on behalf of a script owner other than the user, unless the Sieve engine has explicit knowledge (e.g., via integration with an XMPP server’s presence authorization rules) that the script owner is authorized to know the user’s presence. While it would be possible to design a more advanced approach to delegation of presence authorization, any such approach is left to future standards work.

7. IANA Considerations

The following template provides the IANA registration of the Sieve notification mechanism specified in this document:
To: iana@iana.org
Subject: Registration of new Sieve notification mechanism
Mechanism name: xmpp
Mechanism URI: draft-saintandre-rfc4622bis
Mechanism-specific options: none
Standards Track/IESG-approved experimental RFC number: this RFC
Person and email address to contact for further information:
Peter Saint-Andre <stpeter@jabber.org>

This information should be added to the list of Sieve notification mechanisms maintained at <http://www.iana.org/assignments/sieve-notification>.

8.  References

8.1.  Normative References

[NOTIFY] Melnikov, A., Leiba, B., Segmuller, W., and T. Martin,
"Sieve Email Filtering: Notifications",
draft-ietf-sieve-notify-12 (work in progress),
December 2007.

[OOB] Saint-Andre, P., "Out of Band Data", XSF XEP 0066,
August 2006.

[QUERIES] Saint-Andre, P., "XMPP URI Scheme Query Components", XSF
XEP 0147, September 2006.

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821,
April 2001.

[SHIM] Saint-Andre, P. and J. Hildebrand, "Stanza Headers and
Internet Metadata", XSF XEP 0131, July 2006.

[SIEVE] Showalter, T. and P. Guenther, "Sieve: An Email Filtering
Language", draft-ietf-sieve-3028bis-13 (work in progress),
October 2007.

[TERMS] Bradner, S., "Key words for use in RFCs to Indicate

[XMPP-URI] Saint-Andre, P., "Internationalized Resource Identifiers
(IRIs) and Uniform Resource Identifiers (URIs) for the
Extensible Messaging and Presence Protocol (XMPP)",
draft-saintandre-rfc4622bis-01 (work in progress),
June 2007.
8.2. Informative References


Authors’ Addresses

Peter Saint-Andre
XMPP Standards Foundation

Email: stpeter@jabber.org
URI: https://stpeter.im/

Alexey Melnikov
Isode Limited

Email: Alexey.Melnikov@isode.com