Multiple-Recipient MESSAGE Requests in the Session Initiation Protocol (SIP)
draft-ietf-sip-uri-list-message-01.txt

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 July 12, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document specifies a mechanism that allows a SIP User Agent Client (UAC) to request a SIP URI-list (Uniform Resource Identifier list) service to send a SIP MESSAGE request to a set of destinations. The client sends a SIP MESSAGE request that includes the payload along with the URI-list to the MESSAGE URI-list service, which sends a similar MESSAGE request to each of the URIs included in the list.
1. Introduction

RFC 3261 (SIP) [5] is extended by RFC 3428 [8] to carry instant messages in MESSAGE requests. One of the aspects of MESSAGE requests according to RFC 3428 [8] is the lack of support for sending the same request to multiple recipients and replying to all recipients of a SIP MESSAGE request. This memo addresses these functions.

A first requirement can be expressed as:

REQ-1: It MUST be possible for a user to send an instant message request to an ad-hoc group, where the identities of the recipients are carried in the message itself.

One possibility to fulfill the above requirement is to establish a session of instant messages with an instant messaging conference server. While this option seems to be reasonable in many cases, in other situations the sending user just wants to send a small page-mode instant message to an ad-hoc group without the burden of setting up a session. This document focuses on sending a page-mode instant message to a number of intended recipients.

To meet the requirement with a page-mode instant message, we allow SIP MESSAGE requests carry recipient-list bodies, i.e., URI-lists in body parts whose Content-Disposition (RFC 2183) [2] is ‘recipient-list’, as specified in the Framework and Security Considerations for SIP URI-List Services [11]. A SIP MESSAGE URI-list service, which is a specialized application service, receives the request and sends a similar MESSAGE request to each of the URIs in the list. Each of these MESSAGE requests contains a copy of the body included in the original MESSAGE request.

A second requirement addresses the "Reply-to-All" functionality:

REQ-2: It MUST be possible for the recipient of a group instant message to send a message to all other participants that received the same group instant message (i.e., Reply-To-All).

To meet this requirement, we provide a mechanism whereby the MESSAGE URI-list service also includes a URI-list in body parts whose Content-Disposition (RFC 2183) [2] is ‘recipient-list-history’, as specified in the Extensible Markup Language (XML) Format Extension for Representing Copy Control Attributes in Resource Lists [12]. The ‘recipient-list-history’ body is sent along with the instant message payload in each of the instant messages sent to the recipients.

The User Agent Client (UAC) that sends a MESSAGE request to a MESSAGE URI-list service needs to be configured with the SIP URI of the
service that provides the functionality. Discovering and provisioning of this URI to the UAC is outside the scope of this document.

2. Terminology

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 [1] and indicate requirement levels for compliant implementations.

MESSAGE URI-list service: SIP application service that receives a MESSAGE request with a URI-list and sends a similar MESSAGE request to each URI in the list. In this context, similar indicates that some SIP header fields can change, but the MESSAGE URI-list service will not change the instant message payload. MESSAGE URI-list services behave effectively as specialised B2BUAs (Back-To-Back-User-Agents). A server providing MESSAGE URI-list services can also offer URI-list services for other methods, although this functionality is outside the scope of this document. In this document we only discuss MESSAGE URI-list services.

Incoming MESSAGE request: A SIP MESSAGE request that a UAC creates and addresses to a MESSAGE URI-list service. Besides the regular instant message payload, an incoming MESSAGE request contains a URI-list.

Outgoing MESSAGE request: A SIP MESSAGE request that a MESSAGE URI-list service creates and addresses to a UAS (User Agent Server). It contains the regular instant message payload.

Intended recipient: The intended final recipient of the request to be generated by MESSAGE URI-list service.

3. Overview

A UAC creates a MESSAGE request that contains a multipart body including a list of URIs (intended recipients) and an instant message. The list of URIs is formatted according to the XML Formats for Representing Resource List [10] and extended with the attributes defined in the XML Format Extension for Representing Copy Control Attributes in Resource Lists [12]. The UAC sends this MESSAGE request to the MESSAGE URI-list service. On reception of this incoming MESSAGE request, the MESSAGE URI-list service creates a MESSAGE request per intended recipient (listed in the URI-list) and
copies the instant message payload to each of those MESSAGES. The
MESSAGE URI-list service also manipulates the XML resource list
according to the procedures indicated in the XML Format Extension for
Representing Copy Control Attributes in Resource Lists [12], and
attaches the result to each of the MESSAGE requests, along with the
instant message payload. Then the MESSAGE URI-list service sends
each of the created outgoing MESSAGE request to the respective
receiver.

The MESSAGE URI-list mechanism allows a sender to specify multiple
targets for a MESSAGE request by including an XML resource list
according to the XML Format for Representing Resource Lists [10] in
the body of the MESSAGE request extended with the attributes defined
in the XML Format Extension for Representing Copy Control Attributes
in Resource Lists [12]. This resource list, whose Content-Disposition (RFC 2183) [2] is ‘recipient-list’, as specified in the
Framework and Security Considerations for SIP URI-List Services [11],
includes the URIs of the targets. Each target URI may also be marked
to indicate in what role the URI-list service will place the target
(e.g., "to", "cc", or "bcc"), and whether the target URI is expected
to be anonymized or not, according to the procedures described in the
XML Format Extension for Representing Copy Control Attributes in
Resource Lists [12]. When the MESSAGE URI-list server expands the
MESSAGE request to each recipient, it includes (along with the
instant message payload) a new URI-list (based on the received one),
whose Content-Disposition (RFC 2183) [2] is ‘recipient-list-history’,
as specified in the XML Format Extension for Representing Copy
Control Attributes in Resource Lists [12]. This new URI-list
includes the list of non-anonymous "to" and "cc" targets, allowing
recipients to both get knowledge of other recipients and reply to
them.

4. URI-List document

As described in the Framework and Security Considerations for SIP
URI-List Services [11], specifications of individual URI-list
services, like the MESSAGE URI-list service described here, need to
specify a default format for ‘recipient-list’ bodies used within the
particular service.

The default format for ‘recipient-list’ bodies for MESSAGE URI-list
services is the XML Formats for Representing Resource Lists [10]
extended with the XML Format Extension for Representing Copy Control
Attributes in Resource Lists [12]. UACs and MESSAGE URI-list
services handling ‘recipient-list’ bodies MUST support both of these
formats and MAY support other formats.
As described in the XML Format Extension for Representing Copy Control Attributes in Resource Lists [12], each URI can be tagged with a 'copyControl' attribute set to either "to", "cc", or "bcc", indicating the role in which the recipient will get the MESSAGE request. Additionally, URIs can be tagged with the 'anonymize' attribute to prevent that the MESSAGE URI-list server discloses the target URI in a URI-list.

Additionally, the XML Format Extension for Representing Copy Control Attributes in Resource Lists [12] defines a 'recipient-list-history' body that contains the list of intended recipients. The default format for 'recipient-list-history' bodies for MESSAGE URI-list services is also the XML Formats for Representing Resource Lists [10] extended with the XML Format Extension for Representing Copy Control Attributes in Resource Lists [12]. MESSAGE URI-list services MUST support both of these formats; UASes MAY support these formats. MESSAGE URI-list servers and UASes MAY support other formats.

Nevertheless, the XML Formats for Representing Resource Lists [10] document provides features, such as hierarchical lists and the ability to include entries by reference relative to the XCAP root URI, that are not needed by the MESSAGE URI-list service defined in this document, which only needs to transfer a flat list of URIs between a UA (User Agent) and the MESSAGE URI-list server.

5. Option-tag

This document defines the 'recipient-list-message' option-tag for use in the Require and Supported SIP header fields.

This option-tag is used to ensure that a server can process the 'recipient-list' body used in a MESSAGE request. It also provides a mechanism to discover the capability of the server in responses to OPTIONS requests.

Section 6 provides normative procedures for the usage of this option tag.

6. Procedures at the User Agent Client

A UAC that wants to create a multiple-recipient MESSAGE request creates a MESSAGE request that MUST be formatted according to RFC 3428 [8] Section 4. The UAC populates the Request-URI with the SIP or SIPS URI of the MESSAGE URI-list service. In addition to the regular instant message body, the UAC adds a recipient-list body whose Content-Disposition type is 'recipient-list', specified in the
Framework and Security Considerations for SIP URI-list Services [11]. This body contains a URI-list with the recipients of the MESSAGE. Target URIs in this body MAY also be tagged with the ‘copyControl’ and ‘anonymize’ attributes specified in the XML Format Extension for Representing Copy Control Attributes in Resource Lists [12]. The UAC MUST also include the ‘recipient-list-message’ option-tag, defined in Section 5, in a Require header field.

UACs generating MESSAGE request that carry recipient-list bodies, as described in previous sections, MUST include this option-tag in a Require header field. UAs that are able to receive and process MESSAGEs with a recipient-list body, as described in previous sections, SHOULD include this option-tag in a Supported header field when responding to OPTIONS requests.

Multiple-recipient MESSAGE requests contain a multipart body that contains the body carrying the list and the actual instant message payload. In some cases, the MESSAGE request can contain bodies other than the text and the list bodies (e.g., when the request is protected with S/MIME as per RFC 3851 [9]).

Typically, the MESSAGE URI-list service will copy all the significant header fields in the outgoing MESSAGE request. However, there might be cases where the SIP UA wants the MESSAGE URI-list service to add a particular header field with a particular value, even if the header field wasn’t present in the MESSAGE request sent by the UAC. In this case, the UAC MAY use the "?" mechanism described in Section 19.1.1 of RFC 3261 [5] to encode extra information in any URI in the list. However, the UAC MUST NOT use the special "body" hname (see Section 19.1.1 of RFC 3261 [5]) to encode a body, since the body is present in the MESSAGE request itself.

The following is an example of a URI that uses the "?" mechanism:

```
sip:bob@example.com?Accept-Contact=*%3bmobility%3d%22mobile%22
```

The previous URI requests the MESSAGE URI-list service to add the following header field to a MESSAGE request to be sent to bob@example.com:

```
Accept-Contact: *;mobility="mobile"
```

The XML Format for Representing Resource Lists [10] document provides features, such as hierarchical lists and the ability to include entries by reference relative to the XCAP root URI. However, these features are not needed by the multiple MESSAGE URI-List service defined in this document. Therefore, when using the default resource list document, UAs SHOULD use flat lists (i.e., no hierarchical
lists) and SHOULD NOT use <entry-ref> elements.

7. Procedures at the MESSAGE URI-List Service

On reception of a MESSAGE request with a URI-list, the MESSAGE URI-list service answers to the UAC with a 202 (Accepted) response.

Note that the status code in the response to the MESSAGE does not provide any information about whether or not the MESSAGEs generated by the URI-list service were successfully delivered to the URIs in the list. That is, a 202 (Accepted) response means that the MESSAGE URI-list service has received the MESSAGE and that it will try to send a similar MESSAGE to the URIs in the list. Designing a mechanism to inform a client about the delivery status of an instant message is outside the scope of this document.

Since the MESSAGE URI-List service does not use hierarchical lists nor lists that include entries by reference to the XCAP root URI, a MESSAGE URI-list server receiving a URI-list with more information than what has just been described MAY discard all the extra information.

7.1. Determining the intended recipient

On reception of a MESSAGE request with a URI-list, a MESSAGE URI-list service determines the list of intended recipients by inspecting the URI-list contained in the body. In case two of those URIs are equivalent (section 19.1.4 of RFC 3261 [5] defines equivalent URIs), the MESSAGE URI-list SHOULD consider a single intended recipient rather than sending multiple copies of the MESSAGE to the same destination.

7.2. Creating an outgoing MESSAGE request

Since the MESSAGE URI-list behaves as a UAC for outgoing MESSAGE requests, for each of the intended recipients, the MESSAGE URI-list service creates a new MESSAGE request according to the procedures described in Section 4 of RFC 3428 [8] and the following procedures:

- A MESSAGE URI-list service MUST include a From header field whose value is the same as the From header field included in the incoming MESSAGE request, subject to the privacy requirements (see RFC 3323 [6] and RFC 3325 [7]) expressed in the incoming MESSAGE request.
Note that this does not apply to the "tag" parameter.

Failing to copy the From header field of the sender would prevent the recipient to get a hint of the sender’s identity. Note also that this requirement does not intend to contradict requirements for additional services running on the same physical node. Specifically, a privacy service (see RFC 3323 [6]) can be co-located with the MESSAGE URI-list service, in which case, the privacy service has precedence over the MESSAGE URI-list service.

- A MESSAGE URI-list service SHOULD generate a new To header field value set to the intended recipient’s URI. According to the procedures of RFC 3261 [5] Section 8.1.1.1, this value is also expected to be equal to the Request-URI of the outgoing MESSAGE request.

  The MESSAGE URI-list service behaves as a User Agent Client, thus, the To header field should be populated with the recipient’s URI.

- A MESSAGE URI-list service SHOULD create a new Call-ID header field value.

  A Call-ID header field might contain addressing information that the sender wants to remain private. Since there is no need to keep the same Call-ID on both sides of the MESSAGE URI-list service, and since the MESSAGE URI-list service behaves as a User Agent Client, it is recommended to create a new Call-ID header field value according to the regular SIP procedures.

- If a P-Asserted-Identity header field was present in the incoming MESSAGE request and the request was received from a trusted source, as specified in RFC 3325 [7], and the first hop of the outgoing MESSAGE request is also trusted, a MESSAGE URI-list service MUST include a P-Asserted-Identity header field in the outgoing MESSAGE request with the same received value. However, if the first hop of the outgoing MESSAGE request is not trusted and the incoming MESSAGE request included a Privacy header field with a value different than ‘none’, the MESSAGE URI-list service MUST NOT include a P-Asserted-Identity header field in the outgoing MESSAGE request.

- If a MESSAGE URI-list service is able to assert the identity of a user (e.g., using HTTP Digest authentication scheme as per RFC 2617 [3], S/MIME as per RFC 3851 [9], etc.) and the service implements a mechanism where it can map that authentication scheme to a user’s SIP or SIPS URI, and subject to the privacy requirements expressed in the incoming MESSAGE request (see RFC
the MESSAGE URI-list MAY insert a P-Asserted-Identity header with the value of the user’s asserted URI.

- If the incoming MESSAGE request contains an Authorization or Proxy-Authorization header fields whose realm is set to the MESSAGE URI-list server’s realm, then the MESSAGE URI-list service SHOULD NOT copy it to the outgoing MESSAGE request; otherwise (i.e., if the Authorization or Proxy-Authorization header field of incoming MESSAGE request contains a different realm), the MESSAGE URI-list service MUST copy the value to the respective header field of the outgoing MESSAGE request.

- A MESSAGE URI-list service SHOULD create a separate count for the CSeq header field of the outgoing MESSAGE request.

- A MESSAGE URI-list service SHOULD initialize the value of the Max-Forward header field of the outgoing MESSAGE request.

- A MESSAGE URI-list service MUST include its own value in the Via header field.

- A MESSAGE URI-list service SHOULD include any other header field expressed with the "?" mechanism described in Section 19.1.1 of RFC 3261 [5] and encoded in the intended recipient URI of the URI-list.

- A MESSAGE URI-list service SHOULD preserve to the outgoing MESSAGE request any other header field not explicitly indicated in the above paragraphs.

### 7.3. Composing bodies in the outgoing MESSAGE request

When creating the body of each of the outgoing MESSAGE requests, the MESSAGE URI-list service tries to keep the relevant bodies of the incoming MESSAGE request and copies them to the outgoing MESSAGE request. The following guidelines are provided:

- A MESSAGE request received at a MESSAGE URI-list service can contain one or more security bodies (e.g., S/MIME, RFC 3851 [9]) encrypted with the public key of the MESSAGE URI-list service. These bodies are deemed to be read by the URI-list service rather than the recipient of the outgoing MESSAGE request (which will not be able to decrypt them). Therefore, a MESSAGE URI-list service MUST NOT copy any security body (such as an S/MIME as per RFC 3851 [9] encrypted body) addressed to the MESSAGE URI-list service to the outgoing MESSAGE request. This includes bodies encrypted with the public key of the URI-list service.

- The incoming MESSAGE request typically contains a recipient-list body or reference, as indicated in the Framework and Security Considerations for SIP URI-List Services [11] with the actual list of recipients. If this URI-list includes resources tagged with the 'copyControl' attribute set to a value of "to" or "cc", the URI-list service SHOULD include a URI-list in each of the outgoing MESSAGE requests. This list SHOULD be formatted according to the

- If the MESSAGE URI-list service includes a URI-list in an outgoing MESSAGE request, it MUST include a Content-Disposition header field as per RFC 2183 [2] with the value set to ‘recipient-list-history’ and a ‘handling’ parameter as per RFC 3204 [4] set to "optional".
- If a MESSAGE URI-list service includes a URI-list in an outgoing MESSAGE request, it SHOULD use S/MIME (RFC 3851) [9] to encrypt the URI-list with the public key of the receiver.
- The MESSAGE URI-list service SHOULD copy all the remaining message bodies (e.g., text messages, images, etc.) of the incoming MESSAGE request to the outgoing MESSAGE request.
- If there is only one body left, the MESSAGE URI-list service MUST remove the multipart/mixed wrapper in the outgoing MESSAGE request.

The rest of the MESSAGE request corresponding to a given URI in the URI-list MUST be created following the rules in Section 19.1.5 "Forming Requests from a URI" of RFC 3261 [5]. In particular, Section 19.1.5 of RFC 3261 [5] states:

"An implementation SHOULD treat the presence of any headers or body parts in the URI as a desire to include them in the message, and choose to honor the request on a per-component basis."

SIP allows to append a "method" parameter to a URI. Therefore, it is legitimate that an the ‘uri’ attribute of the <entry> element in the XML resource list contains a ‘method’ parameter. MESSAGE URI-list services MUST generate only MESSAGE requests, regardless of the ‘method’ parameter that the URIs in the list indicate. Effectively, MESSAGE URI-list services MUST ignore the ‘method’ parameter in each of the URIs present in the URI-list.

8. Procedures at the UAS

A UAS (in this specification, also known as intended recipient UAS) that receives a MESSAGE request from the URI-list service behaves as specified in RFC 3428 [8] Section 7.

If the UAS supports this specification and the MESSAGE request contains a body with a Content-Disposition header field as per RFC 2183 [2] set to ‘recipient-list-history’, then the UAS will be able
to determine who are the other intended recipients of the MESSAGE request. This allows the user to create a reply request (e.g., MESSAGE, INVITE) to the sender and the rest of the recipients included in the URI-list.

9. Examples

Figure 1 shows an example of operation. A SIP UAC issuer sends a MESSAGE request. The MESSAGE URI-list service answers with a 202 (Accepted) response and sends a MESSAGE request to each of the intended recipients.

```
+--------+        +---------+      +--------+ +--------+ +--------+
|SIP UAC |        | MESSAGE |      |intended| |intended| |intended|
| issuer  |        | URI-list|      | recip.  | | recip.  | | recip.  |
| service |        | service |      |   1     | |   2     | |   n     |
+--------+        +---------+      +--------+ +--------+ +--------+

F1. MESSAGE
F2. 202 Accepted
F3. MESSAGE
F4. MESSAGE
F5. MESSAGE
F6. 200 OK
F7. 200 OK
F8. 200 OK

Figure 1: Example of operation
```

The MESSAGE request F1 (shown in Figure 2) contains a multipart/mixed body that is composed of two bodies: a text/plain body containing the instant message payload and an application/resource-lists+xml body containing the list of recipients.
MESSAGE sip:list-service.example.com SIP/2.0
Via: SIP/2.0/TCP uac.example.com
;branch=z9hG4bKhjhs8ass83
Max-Forwards: 70
To: MESSAGE URI-list Service <sip:list-service.example.com>
From: Alice <sip:alice@example.com>;tag=32331
Call-ID: d432fa84b4c76e66710
CSeq: 1 MESSAGE
Require: recipient-list-message
Content-Type: multipart/mixed;boundary="boundary1"
Content-Length: 501

--boundary1
Content-Type: text/plain

Hello World!

--boundary1
Content-Type: application/resource-lists+xml
Content-Disposition: recipient-list

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
 xmlns:cp="urn:ietf:params:xml:ns:copycontrol">
 <list>
  <entry uri="sip:bill@example.com" cp:copyControl="to" />
  <entry uri="sip:randy@example.net" cp:copyControl="to"
 cp:anonymize="true"/>
  <entry uri="sip:eddy@example.com" cp:copyControl="to"
 cp:anonymize="true"/>
  <entry uri="sip:joe@example.org" cp:copyControl="cc" />
  <entry uri="sip:carol@example.net" cp:copyControl="cc"
 cp:anonymize="true"/>
  <entry uri="sip:ted@example.net" cp:copyControl="bcc" />
  <entry uri="sip:andy@example.com" cp:copyControl="bcc" />
 </list>
</resource-lists>

--boundary1--

Figure 2: MESSAGE request received at the MESSAGE URI-list server

The MESSAGE requests F3, F4 and F5 are similar in nature. All those
MESSAGE requests contain a multipart/mixed body which is composed of
two other bodies: a text/plain body containing the instant message
payload and an application/resource-lists+xml containing the list of
recipients. Unlike the text/plain body the application/
resource-lists+xml body is not equal to the application/
resource-lists+xml included in the incoming MESSAGE request F1,
because the URI-list service has anonymized those URIs tagged with the 'anonymize' attribute and has removed those URIs tagged with a "bcc" 'copyControl' attribute. Figure 3 shows an example of the message F3.

MESSAGE sip:bill@example.com SIP/2.0
Via: SIP/2.0/TCP list-service.example.com
;branch=z9hG4bKhjhs8as34sc
Max-Forwards: 70
To: <sip:bill@example.com>
From: Alice <sip:alice@example.com>;tag=210342
Call-ID: 39s02sds120d9sj21
CSeq: 1 MESSAGE
Content-Type: multipart/mixed;boundary="boundary1"
Content-Length: 501

--boundary1
Content-Type: text/plain
Hello World!

--boundary1
Content-Type: application/resource-lists+xml
Content-Disposition: recipient-list-history; handling=optional

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
xmlns:cp="urn:ietf:params:xml:ns:copycontrol">
  <list>
    <entry uri="sip:bill@example.com" cp:copyControl="to" />
    <entry uri="sip:anonymous@anonymous.invalid" cp:copyControl="to"
      cp:count="2"/>
    <entry uri="sip:joe@example.org" cp:copyControl="cc" />
    <entry uri="sip:anonymous@anonymous.invalid" cp:copyControl="cc"
      cp:count="1"/>
  </list>
</resource-lists>
--boundary1--

Figure 3: MESSAGE request sent by the MESSAGE URI-list server

10. Security Considerations

The Framework and Security Considerations for SIP URI-List Services [11] discusses issues related to SIP URI-list services. Implementations of MESSAGE URI-list services MUST follow the security-related rules in the Framework and Security Considerations...

If the contents of the instant message needs to be kept private, the user agent client SHOULD use S/MIME as per RFC 3851 [9] to prevent a third party from viewing this information. In this case, the user agent client SHOULD encrypt the instant message body with a content encryption key. Then, for each receiver in the list, the UAC SHOULD encrypt the content encryption key with the public key of the receiver, and attach it to the MESSAGE request.

11. IANA Considerations

This document defines the SIP option tag 'recipient-list-message'

The following row shall be added to the "Option Tags" section of the SIP Parameter Registry:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>recipient-list-message</td>
<td>The body contains a list of URIs that indicates the recipients of the SIP MESSAGE request</td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

Table 1: Registration of the 'recipient-list-message' Option-Tag in SIP

Note to IANA and the RFC editor: replace RFCXXXX above with the RFC number of this specification.

12. Acknowledgements

Duncan Mills supported the idea of having 1 to n MESSAGEs. Ben Campbell, Paul Kyzivat, Cullen Jennings, Jonathan Rosenberg, Dean Willis, and Keith Drage provided helpful comments.

13. Normative References

Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, August 1997.


Authors’ Addresses

Miguel A. Garcia-Martin
Nokia
P.O.Box 407
NOKIA GROUP, FIN 00045
Finland

Email: miguel.an.garcia@nokia.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com