Remote Call Control in SIP using the REFER method and the
session-oriented dialog package
draft-mahy-sip-remote-cc-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with
all provisions of Section 10 of RFC2026.

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

This Internet-Draft will expire on March 31, 2004.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document describes how to use the SIP REFER method and the
dialog package to manipulate conversations, dialogs, and sessions on
remote User Agents. This functionality is most useful for
collections of loosely coupled User Agents that wish to present a
coordinated user experience. It does not require a Third-Party Call
Control controller to be involved in any of the manipulated dialogs.
Table of Contents

1. Conventions ............................................. 3
2. Introduction ............................................ 3
3. Examples of Remote Call Control Operations ............. 4
4. User Agent Behavior ................................. 8
4.1 Organizing requests within dialogs .................. 8
4.2 Addressing the relevant parties ..................... 10
4.3 Selecting an existing dialog context for the triggered request ............................................. 11
4.4 Accessing Local Services Remotely .................. 11
4.5 Authorizing remote call control requests ............ 12
5. More complex examples ............................... 13
6. Handling DTMF ........................................ 15
7. Formal Syntax .......................................... 15
8. Security Considerations ............................... 16
9. IANA Considerations .................................. 16
10. Acknowledgments ...................................... 16
    Normative References ................................ 16
    Informational References ............................. 17
    Authors’ Addresses ................................... 18
    Intellectual Property and Copyright Statements .... 19
1. Conventions

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 RFC-2119 [2].

To simplify discussions related to the REFER method and its extensions, three new terms will be used:
- REFER-Issuer: the UA issuing the REFER request. Sometimes this document will also use the term "controller".
- REFER-Recipient: the UA receiving the REFER request
- REFER-Target: the UA designated in the Refer-To URI

2. Introduction

The SIP [1] core protocol describes how User Agents originate and terminate sessions. The SIP call control framework [11] also describes how User Agents involved in these sessions can manipulate conversations based on the sessions to provide functionality such as transfer, pickup, and barge-in. Third-Party Call Control [13] goes on to describe how a controller can setup dialogs with a number of participants in order to manipulate sessions among the participants.

Remote call control is the manipulation of conversations and session-oriented dialogs by a UA that is not directly involved in any of the relevant conversations, dialogs, or sessions. This manipulation generally involves sending REFER [4] requests to a UA which is directly involved, using information obtained via the dialog package [5]. (Although many are familiar with REFER only as used to implement call transfer [12], the authors of the REFER method never intended this limitation. In fact the REFER method was created when the SIP working group realized that a generic request to ask another UA to do something on your behalf was much more powerful than just doing transfers.)

Unlike the Third-Party Call Control (3pcc) model which requires its controller to act as a B2BUA and maintain dialog state for all relevant dialogs, all the SIP entities involved in remote call control using REFER are just regular SIP User Agents. For convenience we can still describe the SIP entity that sends requests to manipulate remote sessions "the controller", but this is just a logical role. A UA that acts as a controller for one request can terminate and originate its own sessions, and even receive remote call control requests as other requests.

Some readers may question if remote call control is an appropriate use of SIP, instead possibly something more appropriate for MGCP [16] or Megaco [17]. The authors believe that remote call control
is an appropriate and natural extension of SIP. Manipulating sessions and dialogs is certainly consistent with core functionality of SIP. This usage of SIP is much different from an MGCP or Megaco master/slave approach. For example, multiple UAs can send remote call control requests. All remote call control requests can be refused based on local authorization policy or if the request doesn’t make sense. Finally, each UA is still fully responsible and authoritative for their own dialog and session state. In other words, each UA still has the last word on its sessions and dialogs, even if asked to perform manipulations on that state by another entity. This seems completely appropriate with the design of SIP. In fact these requirements and goals are well documented in the SIP Call Control Framework.

Remote call control is especially useful for collections of loosely coupled User Agents which would like to present a coordinated user experience. Among other things, this allows User Agents which handle orthogonal media types but which would like to be present in a single conversation to add and remove each other from the conversation as needed. This is especially appropriate when coordinating conversations among organizers, general purpose computers, and special purpose communications appliances like telephones, Internet televisions, in-room video systems, electronic whiteboards, and gaming devices.

For example using remote call control, an Instant Messaging client could initiate a multiplayer gaming session and an audio session to a chat conversation. Likewise a telephone could add an electronic whiteboard session to a voice conversation. Finally, a computer or organizer could cause a nearby phone to dial from numbers or URIs in a document, email, or address book; allow users to answer or deflect incoming calls without removing hands from the computer keyboard; place calls on hold; and join other sessions on the phone or otherwise.

3. Examples of Remote Call Control Operations

This entire section provide non-normative examples of functionality where a computer or PDA manipulates a telephone. The behavior for remote call control with other types of devices is similar, but describing similar manipulations for other media or device types would naturally use a different set of vocabulary.

In the requests labeled with 1 and 2, Alice’s PC or PDA sets up a subscription to the dialog package from Alice’s phone (messages shown in a later section). All of the subsequent NOTIFY messages are notifications about changes in the dialog state at Alice’s phone. In message 3, Alice’s PC or PDA asks her phone to "call Bob" (message
4), which eventually results in an early dialog (5) with one of Bob’s Contacts. [Note Well: Parts of the flow marked in parentheses (including messages 6, 7, 9, and 10) show alternative outcomes in the call flow.]

<table>
<thead>
<tr>
<th>Alice’s PC or PDA</th>
<th>Alice’s Phone</th>
<th>Bob</th>
<th>Cathy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call-ID: 123</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---SUBSCRIBE/200--&gt;</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---REFER/202------</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td>---INVITE------</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(---REFER/202------</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(---CANCEL/200-----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( &lt;--NOTIFY/200-----</td>
<td>---487/ACK-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(---REFER/202------</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(---BYE/200--------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( &lt;--NOTIFY/200-----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--------------INVITE/180-----</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(---REFER/202------</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( &lt;--NOTIFY/200-----</td>
<td>---302/ACK-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(---REFER/202------</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( &lt;--NOTIFY/200-----</td>
<td>---486/ACK-----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---REFER/202------</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Messages 3, 4, and 5 follow. The norefersub option-tag on each REFER suppresses the implicit subscription which would normally follow the REFER (the notifications in the call flow diagram are for the dialog package subscription in messages 1 and 2).

Via and Max-Forward headers and session descriptions are omitted for brevity and clarity. In some cases, display names are added for simplify the task of the reader following the examples. Note that URIs in SIP cannot wrap lines. Due to RFC formatting conventions, this draft splits URIs across lines where the URI would exceed 72 characters. A backslash character marks where this line folding has taken place. Finally, some of the URIs shown here are not escaped properly to aid in readability. In message 9 the @ in the Refer-To URI should be escaped.

Message 3:

REFER sip:reg2@10.1.1.3 SIP/2.0
To: "Alice’s phone" <sip:reg2@10.1.1.3>;tag=def
From: "Alice’s PC or PDA" <sip:alice1@10.1.1.2>;tag=abc
Call-ID: 123
CSeq: 2 REFER
Require: remotecc
Supported: norefersub
Refer-To: "Bob" <sip:bob@example.net>

Message 4:

INVITE sip:bob@example.net SIP/2.0
To: "Bob" <sip:bob@example.net>
From: "Alice" <sip:alice@example.com>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Alice’s Phone" <sip:reg2@10.1.1.3>
Content-Type: application/sdp
Content-Length: xxx

Message 5:

SIP/2.0 180 Ringing
To: "Bob" <sip:bob@example.net>;tag=uvw
From: "Alice’s phone" <sip:reg2@10.1.1.3>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Bob’s Contact" <sip:line1@192.168.0.5>
The rest of the REFER messages in this example are identical to the REFER in Message 3 except for the Refer-To header, CSeq header, and Via branch id (not shown). Message fragment 6 and 7 show the Refer-To headers which Alice’s PC or PDA would send to cause Alice’s phone to terminate the session which Message 4 attempted to originate. The extra parameters in the Refer-To header are used to explicitly match a specific dialog. They are more fully described in a later section.

Header for Message 6:

Refer-To: "Bob’s Contact" <sip:line1@192.168.0.5;method=CANCEL>
;call-id=456;remote-tag=;local-tag=xyz

Header for Message 7:

Refer-To: "Bob’s Contact" <sip:line1@192.168.0.5;method=BYE>
;call-id=456;remote-tag=uvw;local-tag=xyz

Message 8 is an invitation received by Alice’s phone from Cathy. Refer-To headers for Messages 9, 10, and 11 are shown below which would cause Alice’s phone to redirect, reject, or accept the invitation. They use the response URI parameter defined in [6]. Note that some specialized User Agents might not be capable of accepting an invitation autonomously. For example, a SIP user agent which connect to an analog telephone cannot physically force the phone to go offhook.
Message 8:

INVITE sip:reg2@10.1.1.3 SIP/2.0
To: "Alice" <sip:alice@example.com>
From: "Cathy" <sip:cathy@example.net>;tag=ijk
Call-ID: 789
CSeq: 1 INVITE
Contact: "Cathy’s Contact" <sip:cathy-pc.example.net>
Content-Type: application/sdp
Content-Length: xxx

Header for 9:

Refer-To: <sip:cathy-pc.example.net;method=INVITE;response=302 \ 
?Contact=sip:doug@example.com> 
;call-id=789;remote-tag=ijk;local-tag=lmn

Header for 10:

Refer-To: <sip:cathy-pc.example.net;method=INVITE;response=486> 
;call-id=789;remote-tag=ijk;local-tag=lmn

Header for 11:

Refer-To: <sip:cathy-pc.example.net;method=INVITE;response=200> 
;call-id=789;remote-tag=ijk;local-tag=lmn

4. User Agent Behavior

4.1 Organizing requests within dialogs

REFER messages used for call transfer usually arrive within an existing dialog which was created with the INVITE method. In general, REFER messages can be sent within an existing dialog, or they can start a new dialog (the dialog used by the implicit subscription they create). In many use cases of remote call control, receiving notifications about the status of a REFER request are superfluous, as the Refer-Issuer typically maintains a long duration subscription to the dialog package. This situation can be addressed by including the norefersub option-tag, defined in section 7 of [6]. When the norefersub option tag is present, a REFER request which would have created a new subscription and dialog becomes a standalone transaction instead. Each such standalone REFER transaction MUST use a new (unique) Call-ID header field value. The following three use cases are suggested:
1. In the most common usage, the controller maintains a long duration subscription to the dialog package, and sends REFER requests within that dialog. Each REFER is sent within the context of the dialog created for the subscription to the dialog package, and should include the norefersub option-tag in a Supported header field value.

2. Occasionally the dialog package is only supported via a dialog state agent separate from the Refer-Receiver, in which case the controller maintains a long duration subscription to the dialog package to a dialog state agent, and the controller sends these individual REFER requests as standalone requests each with a different (unique) Call-ID header field value, which could also include the norefersub option-tag in a Supported header field value.

3. In some cases, the controller does not maintain a dialog package subscription for the Refer-Receiver. This might be the case for a "webdialer" or other application which associates with other UAs on an adhoc and intermittent basis. An initial REFER request is sent to start a new dialog, which is followed by notifications for the refer event type (the norefersub option-tag SHOULD NOT be used in this case). These notifications could contain message/sipfrag or application/dialog-info+xml notification bodies as described in Section 4 of [6].

   OPEN ISSUE: Should we restrict usage to one of these three models?
   OPEN ISSUE: Are there other models possible?

Message 1:

```
SUBSCRIBE sip:reg2@10.1.1.3 SIP/2.0
To: "Alice’s phone" <sip:reg2@10.1.1.3>
From: "Alice’s PC or PDA" <sip:alice1@10.1.1.2>;tag=abc
Call-ID: 123
CSeq: 1 SUBSCRIBE
Event: dialog
Contact: <sip:alice1@10.1.1.2>
```

Message 2:

```
NOTIFY sip:reg2@10.1.1.3 SIP/2.0
To: "Alice’s PC or PDA" <sip:alice1@10.1.1.2>;tag=abc
From: "Alice’s phone" <sip:reg2@10.1.1.3>;tag=def
Call-ID: 123
CSeq: 1 NOTIFY
Event: dialog
Contact: <sip:reg2@10.1.1.3>
Subscription-State: active;expires=3600
Content-Type: application/dialog-info+xml
Content-Length: xxx
```
4.2 Addressing the relevant parties

REFER requests contain a number of URIs which need to address the appropriate parties. A list of the relevant fields include the Request-URI, To header URI, From header URI, Contact header URI, Refer-To header URI, and the Referred-By header URI. This section defines the semantics of each field.

In most cases, remote call control seeks to manipulate dialogs or sessions on a specific UA. For this reason, the Request URI of the REFER request SHOULD be a valid GRUU for a single UA (a Contact URI). Contact URIs for a UA can be discovered by subscribing to the registration package for the relevant AORs.

In the rare exceptions when the controller does not care which specific UA it manipulates, an AOR MAY be used instead. When an AOR is used, the REFER request can include appropriate caller-preferences to encourage selection of an appropriate Contact. The norefersub option-tag MUST NOT be used when the REFER Request-URI is an AOR, as the REFER Request could fork and cause incorrect behavior. While, the controller can discourage a proxy from forking remote call control request by using the Request-Disposition: no-fork header field, insuring that no proxy forks requires the use of the callerpref option-tag in a Proxy-Require header field value. Because any proxy in the chain of this request which did not support caller preferences would cause the request to fail, use of Proxy-Require is NOT RECOMMENDED.

For remote call control requests to operate as expected, the Refer-Issuer needs to be confident that the Refer-Receiver supports the extensions and conventions described here. Otherwise, the triggered request might have completely different semantics from the request which was indicated in the Refer-To header. (Most implementations ignore unknown URI and header parameters). For example a REFER intended to cause the Refer-Receiver to send a 486 Busy Here response for an existing dialog, might instead trigger a new INVITE to the sender of the original INVITE. Implementations which send remote call control requests MUST include the remotecc option-tag in a Require header field value in each REFER request. (Note that support for this option-tag also implies support for the response URI parameter in a Refer-To header.)

The To header field in the REFER request should contain the same URI as in the Request-URI, and the From identifies the AOR of the controller. The Refer-To is set to whatever URI would normally be inserted by a user of the Refer-Receiver (if operated autonomously). A REFER triggering a standalone request or dialog starting request, could send to either an AOR or a Contact address, but typically to an
A REFER request triggering a request which is in a dialog MUST always place a Contact URI in the Refer-To header.

When set, the Referred-By header field SHOULD be the same URI as the URI in the Contact address of the REFER. If included by the Refer-Issuer, it SHOULD be protected with a signed authenticated identity body as recommended in the Referred-By specification.

### 4.3 Selecting an existing dialog context for the triggered request

Many uses of remote call control require that the Refer-Receiver generate a new request or response in the context of an existing dialog, which was not necessarily initiated by the controller. For example, the controller might want the Refer-Receiver to send a BYE, CANCEL, or response to an INVITE in the context of a dialog created with INVITE. For subscriptions, the controller might want the Refer-Receiver to unsubscribe (send a SUBSCRIBE with an Expires header field of 0).

To select the appropriate dialog from which to source the request, this document proposes a few new (header) parameters to the Refer-To header (the call-id, remote-tag, and local-tag parameters). Explicit header parameters were selected because they can apply to non SIP URIs. For example, the following URI, loads a "How To" website in the context of an existing dialog (presumably one created with an INVITE). When the associated dialog completes, the content may be hidden or dismissed with the context with which it was associated.

Refer-To: <http://support.example.com/howto.html>; call-id=xyz; remote-tag=123; local-tag=456

When describing the context of a subscription, the event and event-id parameters are also used. These correspond to the event type and the event-id parameter in the Event header (if present).

Explicit matching of target dialogs and subscriptions was intentionally selected instead of including the appropriate values in embedded Call-ID, To, From, and Event headers. Among other benefits, this reduces the length of the URI portion of the Refer-To header and simplifies URI encoding requirements dramatically.

### 4.4 Accessing Local Services Remotely

It may be desirable to have a URI convention for contacts on some UAs which gives autoanswer behavior. This allows for the development of several services if properly authorized (for example, an intercom service). In the context of remote call control, this URI could be placed in the Request-URI of a REFER requesting a new dialog (as in
Message 3 in the examples) instead of Alice’s regular Contact address (at issue is if non-autoanswer behavior if desirable on devices which are capable of autoanswer). There are several possible syntactic choices for an autoanswer URI based on Alice’s registration. (We cannot restrict ourselves to only one autoanswer URI on Alice’s phone, since multiple registrations may exhibit different authorization, alerting, and other behavior.) Three of these choices are listed below.

(option1) sip:reg2+autoanswer@10.1.1.3
(option2) sip:reg2;autoanswer@10.1.1.3
(option3) sip:reg2@10.1.1.3;autoanswer

Hold is a very common local service on phones. Unfortunately hold has two semantics. Hold is often used to describe a primitive operation of setting all media lines in a session to inactive. (We will call this Simple Hold). Hold also describes a server running on a phone which can cause different behavior, for example, music on hold, tone on hold, or simple hold. It is desirable to be able to access the "Hold" service on Alice’s phone via a URI. In the absense of any other convention, we will use the following URI: sip:service.hold@10.1.1.3 . Using this "service URI" on the phone, Alice’s PC or PDA can ask the phone to place a specific session on or off hold as shown below.

REFER sip:reg2@10.1.1.3 SIP/2.0
Refer-To: <sip:service.hold@10.1.1.3>
   ;call-id=789;remote-tag=ijk;local-tag=lmn

REFER sip:reg2@10.1.1.3 SIP/2.0
Refer-To: <sip:service.unhold@10.1.1.3>
   ;call-id=789;remote-tag=ijk;local-tag=lmn

The SIP conferencing framework [14] and the cc-conferencing describe the Conference Factory as a service which assigns new conference URIs. The conference factory is itself accessible via a URI. In some cases, this conference factory is colocated with a phone or other single-user UA. In the absense of any other convention, we use the following URI to reference the conference factory service on Alice’s phone: sip:service.conf-factory@10.1.1.3 .

OPEN ISSUE: How are service URIs registered?
OPEN ISSUE: Is this a complete set of services that are needed for remote call control?

4.5 Authorizing remote call control requests

To be written.
5. More complex examples

The following example shows how a controller can cause the Refer-Receiver to complete a transfer between two existing calls.

Alice’s PC or PDA Alice’s Phone Call-ID: 456 Call-ID: 789

A ---REFER/202------>
B ---REFER/202------>
C <---NOTIFY/200------> ---INVITE------>
d/Replaces

D <---NOTIFY/200------>
E <---NOTIFY/200------>
F <---NOTIFY/200------>

Message A: (CHECK TAGS!) (character escaping)

Refer-To: "Bob’s Contact" <sip:line1@192.168.0.5;method=REFER \ ?Refer-To=<sip::cathy-pc.example.net?Replaces=789;to-tag=ijk;from-tag=lmn>> ;call-id=456;remote-tag=uvw;local-tag=xyz

Messages C and D are notifications to the refer package. Messages E and F are notifications to the dialog package.

The following example shows how a controller can cause the Refer-Receiver to join two existing sessions using a SIP conference server (labeled "Focus" in the call flow diagram). Note that < > ; and = characters in these URIs need to be escaped.

Alice’s PC or PDA Focus Alice’s Phone Call-ID: 456 Call-ID: 789

Mahy & Levin Expires March 31, 2004
Message G:

Refer-To: <sip:service.conf-factory@10.1.1.3>
; call-id=456; remote-tag=uvw; local-tag=xyz

Message H:

SIP/2.0 200 OK
To: "Conference Factory" <sip:service.conf-factory@10.1.1.3>; tag=ccc
From: "Alice" <sip:alice@example.com>; tag=bbb
Call-ID: aaa
CSeq 1 INVITE
Contact: "Conference #3" <sip:conf3@10.1.1.3>; isFocus
Content-Type: application/sdp
Content-Length: xxx

Message I:

REFER sip:conf3@10.1.1.3 SIP/2.0
Refer-To: <sip:cathy-pc.example.net \?
?Replaces=789;to-tag=ijk;from-tag=lmn>
Referred-By: "Alice’s PC or PDA" <sip:alice1@10.1.1.2>; cid="333@444"
Message J:

REFER sip:conf3@10.1.1.3 SIP/2.0
Refer-To: "Bob’s Contact" <sip:line1@192.168.0.5 \ 
  ?Replaces=456;to-tag=uvw;from-tag=xyz>
Referred-By: "Alice's PC or PDA" <sip:alice1@10.1.1.2>;cid="111@222"

6. Handling DTMF

Occasionally it is useful for one UA to collect digits on behalf of a User Agent which can actually send them using RFC2833 [19]. One of the options is that the collecting UA send KPML [20] responses to the UA capable of turning these keypad events into DTMF media. The method used for sending markup responses is under discussion currently, but one proposal is a new method called FEEDBACK as part of [21]. For example, it is possible that the KPML response tag include new parameters as shown below to identify the dialog for which the keypad markup is intended.

FEEDBACK sip:reg2@10.1.1.3 SIP/2.0
To: "Alice’s phone" <sip:reg2@10.1.1.3>;tag=def
From: "Alice’s PC or PDA" <sip:alice1@10.1.1.2>;tag=abc
Call-ID: 123
CSeq: 27 FEEDBACK
Content-Type: application/kpml+xml
Content-Length: xxx

<?xml version="1.0">
<kpml version="1.0">
  <response digits="9999#" call-id="456"
    remote-tag="uvw" local-tag="xyz"/>
</kpml>

OPEN ISSUE: Need further investigation of mechanism to carry DTMF.
Can this be generalized?

7. Formal Syntax

The following syntax specification extends the Refer-To header described in RFC3515 using the augmented Backus-Naur Form (BNF) as described in RFC-2234 [3].
Refer-To = ("Refer-To" / "r") HCOLON ( name.addr / addr-spec ) *
(SEMI referto-params)

referto-params = callid-param / rtag-param / ltag-param /
    event-param / eventid-param / generic-param

callid-param = "call-id" EQUAL ( token / LDQOT callid RDQUOT )
rtag-param = "remote-tag" EQUAL token
ltag-param = "local-tag" EQUAL token
event-param = "event" EQUAL event-type
eventid-param = "event-id" EQUAL token

8. Security Considerations

The functionality described in this document allows an authorized
party to manipulate SIP sessions and dialogs in arbitrary ways.
Implementations need to take reasonable precautions to insure
authenticity of remote call control request, which MUST be sent using
either hop-by-hop TLS [10] via a SIPS URI, or individually signed
using SMIME [9]. Signing remote call control requests with SMIME is
RECOMMENDED. In addition, UAs which support remote call control
SHOULD sign Referred-By headers in remote call control requests in an
appropriate authenticated identity body. UAs which support remote
call control MUST implement SIPS, SHOULD implement SMIME signing and
verification, and SHOULD implement separate signing of Referred-By
headers in an appropriate authenticated identity body.

9. IANA Considerations

Need to register the remotecc option-tag, the Refer-To header
parameters, and the SIP URI parameters

10. Acknowledgments

Many thanks to Sean Olson, Robert Sparks, and Alan Johnston.

Normative References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,
    Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP:

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


Informational References


Authors’ Addresses

Rohan Mahy
Cisco Systems, Inc.
5617 Scotts Valley Drive, Suite 200
Scotts Valley, CA 95066
USA
EMail: rohan@cisco.com

Orit Levin
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA
EMail: oritl@microsoft.com
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). 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 assignees.

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.

Acknowledgment

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