Remote Call Control in SIP using the REFER method and the
session-oriented dialog package
draft-mahy-sip-remote-cc-01.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 August 1, 2004.

Copyright Notice

Copyright (C) The Internet Society (2004). 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. Terminology ................................................. 3
2. Introduction ............................................. 3
3. Remote control operations ............................... 4
4. Implementing these operations ......................... 6
5. Examples of Remote Call Control Operations SIP Call Flows 8
   5.1 Make Call Operation .................................. 8
   5.2 Answer Call Operation ................................ 10
   5.3 Clear Connection ..................................... 12
   5.4 Deflect Call .......................................... 14
   5.5 Single Step Transfer Call ............................ 16
   5.6 Complete Transfer Between Sessions ................ 18
   5.7 Hold Call ........................................... 18
   5.8 Retrieve Call ......................................... 20
   5.9 Conference Call ...................................... 21
   5.10 Single Step Conference Call ....................... 23
   5.11 Set Do Not Disturb .................................. 25
   5.12 Set Forwarding ...................................... 26
   5.13 Alternate Call ....................................... 28
   5.14 Consultation Call .................................... 29
6. Examples of implementing remote call control operations  
  with Refer-To URI .......................................... 30
   6.1 Make Call Operation .................................. 30
   6.2 Answer Call Operation ................................ 30
   6.3 Clear Connection ..................................... 30
   6.4 Deflect Call .......................................... 31
   6.5 Complete Transfer Between Calls ................... 31
7. User Agent Behavior ......................................... 31
   7.1 Organizing requests within dialogs .................. 31
   7.2 Addressing the relevant parties .................... 32
   7.3 Selecting an existing dialog context for the triggered 
     request ................................................. 33
8. Authorizing remote call control requests ................. 34
9. Security Considerations ................................... 34
10. IANA Considerations ...................................... 35
11. Acknowledgments .......................................... 35
    Normative References .................................... 35
    Informational References ............................... 36
    Authors’ Addresses ..................................... 37
    Intellectual Property and Copyright Statements .... 38
1. 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 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 [13] 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 [15] 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 [14], 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.) The Extensions to the REFER mechanism [6] describes the use of REFER for that purpose.

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
Some readers may question if remote call control is an appropriate use of SIP, instead possibly something more appropriate for MGCP [19] or Megaco [20]. 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.

Remote call control can also be used in two directions. A computer could remote control a nearby phone and make it dial a SIP URI, but the SIP phone could then also remote control the computer into terminating the session upon the user hanging up the phone.

3. Remote control operations

Remote call control can be used to request a variety of operations.
Commonly used operations include the following:

Make Session - Initiate a new session.

Clear Session - Terminate a session.

Answer - Successfully respond to a session invitation.

Deflect Session - Redirect a session invitation.

Reject Session - Reject a session invitation.

Single Step Transfer Session - Transfer a session to another UA in a single step. The transferring device is no longer involved with the session after single step transfer is completed. This is described as a "Blind Transfer" in [14].

Complete Transfer Between Sessions - Transfer the remote UA of one existing session to communicate directly with the remote UA of another existing session. Once the transfer completes, the remote controlled UA is no longer involved with either session.

Hold Session - Holds a call at the holding UA. Note that this operation would cause whatever call control would occur locally when this operation is selected (for example a simple hold which makes the call inactive, or a service such as music on hold using a remote stream.

Retrieve Session - Retrieves a held call at the retrieving device.

Merge Sessions - Conferences together two existing sessions at a UA.

Single Step Conference Call - Initiate another session and merge it to an existing session into a new conference.

Alternate Sessions - Place an existing session on hold, and retrieves a previously held session. This operation is a combination of the Hold Call and Retrieve Call operations.

Consultation Session - Places an existing session on hold at the UA and initiates a new session from the UA. This operation is a combination of the Hold Call and Make Call operations.

Set Do Not Disturb - Will cause the remote controlled UA to reject further session invitations with a proper response indicating that it is not available. This operation does not require the participation of the controller for subsequent session
invitations. The target may cause this operation via local processing or for example by updating presence [17] status which is consumed by systems performing call routing.

Set Forwarding - Will cause the remote controlled UA to redirect further session invitations to another URI. This operation does not require the participation of the controller for subsequent session invitations. The target may cause this operation via local processing or for example by manipulating SIP registrations.

4. Implementing these operations

In order to convey requests for remote call control operations, there are several syntactic approaches possible. The most obvious is to use the existing Refer-To URI syntax. However, escaping long URIs is error-prone and obfuscates the intent of a request. Another option mentioned as a REFER extension is carrying the Refer-To target as a message/sipfrag [12] body. However, encoding remote call control operations which deal with with more that one session in a single URI are still cumbersome. Also, both these approaches rely on implicit behavior or undefined URI conventions. This document uses this approach for operations which only require a straightforward encoding.

Alternatively, the Refer-To URI could be a Universal Resource Name (URN) [21] which could describe a particular operation such as Hold or Retrieve. Combined with the dialog-identifiers of an existing session conveyed as parameters of the Refer-To header, this would permit explicit operations which do not need additional parameters or handle more than a single session. For example, the following could represent a Hold operation of a session with the Call-ID "123":

Refer-To: <urn:ietf:params:sip:remotecc:hold>;call-id=123;remote-tag=aaa;local-tag=bbb

Note however that the most interesting remote call control operations (such as Complete Transfer Between Sessions and Merge) operate on more than one session and may require additional parameters. These are still abstract operations, but they operate on more than one target. Using an explicit description of these parameters in a new MIME body is an ideal way to provide this additional functionality, and the only approach which works with all the sample remote call control operations in this document.

An additional benefit of a remote call control body is that certain details of these operations can be abstracted. For example, a Clear Session operation can cause either a CANCEL, BYE or appropriate
response to be sent depending on context. A Hold operation can result in whatever user-visible functionality occurs when a Hold is selected locally (for example a simple hold, tone-on-hold, music-on-hold, animated cartoon characters, etc.). A Merge Sessions operation can use whatever conference resource would be used by the UA itself (a local conferencing focus, a discovered focus, or an administratively configured focus).

This document therefore describes a MIME body for remote call control operations conveyed in the body of a REFER request. Remote call control operations using a remote call control MIME type body are operations that are typically more abstract or complex information than can be practically be achieved with a message/sipfrag body or a Refer-to URI.

This document makes frequent use of the REFER extensions defined in [6] to carry out these operation. In particular, we frequently reference bodies in the Refer-To header using a Content-ID URI (cid:).

While a remote call control MIME body is not defined in this document, we use the MIME type application/remotecc in our examples.

The following is an example of a REFER with a Remote Call Control operation with such a body:

REFER sip:reg2@10.1.1.3 SIP/2.0
Via: SIP/2.0/TCP issuer.example.com;branch=z9hG4bK-a-1
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@issuer.example.com
CSeq: 2 REFER
Max-Forwards: 70
Contact: sip:alice1@10.1.1.2
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: <cid:1239103912039@issuer.example.com>;call-id=
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

| Remote Call Control Body |

The application/dialog-info+xml package can be used to provide information about the status of dialogs. The examples in this specification assume that the dialog event package is sufficient to
provide the necessary feedback for remote call control operations.

5. Examples of Remote Call Control Operations SIP Call Flows

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.

The following sub-sections provide an example for every operation described in the previous section.

The following notes are applicable to all the call flows in the subsections below:

It is assumed that Alice’s PC or PDA has subscribed to Alice’s Phone dialog package. All of the NOTIFY messages are notifications about changes in the dialog state at Alice’s phone. No additional remote call control event packages are shown, but it is not precluded that one be defined later.

As specified in [6], there is no implicit subscription on all REFER messages between Alice’s PDA or PC and Alice’s Phone with the extended REFER mechanism.

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.

5.1 Make Call Operation

In message 1, Alice’s PC or PDA asks her phone to "call Bob" (message 2), which eventually results in an early dialog (3) with one of Bob’s Contacts. Bob sends a ringing indication (4) which triggers Alice’s phone to send a notification (5) of "early" to Alice’s PC or PDA. Then Bob answers the phone (6) which triggers Alice’s phone to send a notification (7) of "confirmed" to Alice’s PC or PDA.

Alice’s PC or PDA          Alice’s Phone                  Bob
In this first example, in Message 1a, traditional Refer-To encoding is used. Message 1b shows how to request this same operation with an embedded remote call control MIME body.

Message 1a:

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
Accept: application/dialog-info+xml
Refer-To: "Bob" <sip:bob@example.net;method=INVITE>

Message 1b:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

------------------------
| Remote Call Control Body |
| MakeCall |
| From: sip:reg2@10.1.1.3 |
| To: bob@example.net |
| other parameters |
------------------------
Message 2:

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 3:

NOTIFY indicates "trying".

Message 4:

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>

Message 5:

NOTIFY will indicates "early".

Message 6:

SIP/2.0 200 OK
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>
Content-Type: application/sdp
Content-Length: xxx

Message 7:

NOTIFY indicates "confirmed".

5.2 Answer Call Operation

In message 1, Bob makes a call to Alice’s Phone. A notification (2)
of "trying" is sent to Alice. Alice’s phone automatically sends a "ringing" (3) to Bob. Another notification (4) of "early" is then sent to Alice’s PC. Alice then instructs (5) her PDA to tell the phone to answer the call (6). Alice’s phone sends a notification (7) of "confirmed" to Alice’s PDA.

Message 1:

INVITE sip:alice@example.com SIP/2.0
To: "Alice" <sip:alice@example.com>
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Bob’s Contact" <sip:line1@192.168.0.5>
Content-Type: application/sdp
Content-Length: xxx

Message 2:

NOTIFY indicates "trying".

Message 3:

SIP/2.0 180 Ringing
To: "Alice" <sip:alice@example.com>;tag=uvw
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Alice’s Phone" <sip:reg2@10.1.1.3>

Message 4:
NOTIFY indicates "early".

Message 5:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

<table>
<thead>
<tr>
<th>Remote Call Control Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>answercall</td>
</tr>
<tr>
<td>Call=</td>
</tr>
<tr>
<td>sip:line1@192.168.0.5</td>
</tr>
<tr>
<td>call-id:456</td>
</tr>
<tr>
<td>remote-tag=uvw</td>
</tr>
<tr>
<td>local-tag=xyz</td>
</tr>
<tr>
<td>other parameters</td>
</tr>
</tbody>
</table>

Message 6:

SIP/2.0 200 OK
To: "Alice" <sip:alice@example.com>;tag=uvw
From: "Bob" <sip:bob@example.net>;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 7:

NOTIFY indicates "confirmed".

5.3 Clear Connection

Alice’s Phone and Bob’s contact are currently in an established dialog. In message 1, Alice’s PC or PDA asks her phone to "clear the connection" with Bob’s phone. (message 2).
Alice’s PC or PDA | Alice’s Phone | Bob
|----------------|-------------|-------
| Call-ID: 123   | Call-ID: 456|

1  ---REFER/202------->
2  ------BYE/200----->
3  <---NOTIFY/200-----

Message 1:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

--------------------------------
Remote Call Control Body
clearconnection
Call= sip:line1@192.168.0.5
call-id:456
remote-tag=uvw
local-tag=xyz
other parameters
--------------------------------

Message 2:

BYE is sent to Bob’s contact.

Message 3:

NOTIFY indicates "trying".

Message 3:

NOTIFY will indicates "terminated".
5.4 Deflect Call

In message 1, Bob makes a call to Alice’s Phone. A notification (2) of "trying" is sent to Alice. Alice’s phone automatically sends a "ringing" (3) to Bob. Another notification (4) of "early" is then sent to Alice’s PC. Alice then instructs (5) her PDA to tell the phone to deflect the call (6) to Cathy. Alice’s phone sends a notification (7) of "terminated" to Alice’s PDA. Bob’s will attempt the call to Cathy (8).

<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>Call-ID: 456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---INVITE-------- 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---NOTIFY/200----- 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 ------180--------&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---NOTIFY/200----- 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 ---REFER/202------&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 ------302/ACK-----&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---NOTIFY/200----- 7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8 ----INVITE--------&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Message 1:

INVITE sip:alice@example.com SIP/2.0
To: "Alice" <sip:alice@example.com>
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Bob’s Contact" <sip:line1@192.168.0.5>
Content-Type: application/sdp
Content-Length: xxx

Message 2:

NOTIFY indicates "trying".

Message 3:

SIP/2.0 180 Ringing
To: "Alice" <sip:alice@example.com>;tag=uvw
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Alice’s Phone" <sip:reg2@10.1.1.3>

Message 4:

NOTIFY indicates "early".

Message 5:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

--------------------------------
| Remote Call Control Body     |
| deflectcall                  |
| Call=                       |
|   sip:line1@192.168.0.5     |
|   call-id:456               |
|   remote-tag=uvw            |
|   local-tag=xyz             |
| other parameters            |
--------------------------------

Message 6:

SIP/2.0 302 Moved Temporarily
To: "Alice" <sip:alice@example.com>;tag=uvw
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Cathy" <sip:cathy@example.net>

Message 7:

NOTIFY indicates "rejected".

Message 8:

INVITE sip:cathy@example.net SIP/2.0
To: "Cathy" <sip:cathy@example.net>
5.5 Single Step Transfer Call

Alice’s Phone and Bob’s contact are currently in an established dialog. In message 1, Alice’s PC or PDA requests that a request be made to transfer the call to Cathy. Alice’s phone sends a request (2) to Bob’s contact to transfer the call to Cathy (3). Call from Bob’s contact to Cathy rings (4), is answered (5). Bob’s contact sends a notification (6) to Alice’s phone because of the REFER implicit subscription. Alice’s phone then terminates the session with Bob’s contact (7) and sends a notification of "terminated" to Alice’s PC or PDA.

Message 1:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...
--------------------------------
| Remote Call Control Body     |
| transfer                    |
|   FirstCall=                |
|     sip:line1@192.168.0.5   |
|     call-id:456             |
|     remote-tag=uvw           |
|     local-tag=xyz           |
|   SecondCall=               |
|     sip:cathy@example.net   |
|   other parameters          |
--------------------------------

Message 2:

REFER 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 REFER
Refer-To: "Cathy" <sip:cathy@example.net;method=INVITE>

Message 3:

INVITE sip:cathy@example.net SIP/2.0
To: "Cathy" <sip:cathy@example.net>
From: "Bob" <sip:bob@example.net>;tag=pqr
Call-ID: 789
CSeq: 1 INVITE
Contact: "Bob’s Contact" <sip:line1@192.168.0.5>
Content-Type: application/sdp
Content-Length: xxx

Messages 4 & 5:

180, 200, ACK when call is set up with Cathy.

Message 6:

NOTIFY will include the sigfrag as per the REFER implicit subscription.

Message 7:

Bob’s contact clears the call.
5.6 Complete Transfer Between Sessions

TBD

5.7 Hold Call

In message 1, Alice’s PC or PDA asks her phone to put on hold the already established dialog with Bob. Alice’s phone sends a re-INVITE to Bob’s contact to put the media stream on hold. Note that a call hold is different concept than held media. In fact, a user can be placed on hold, and be provided with music on hold. A held call is a logical state which could be useful for a number of things such as monitoring the amount of time a user stays in a queue. This diagram does not illustrate any event package to illustrate that a can can be held.

<table>
<thead>
<tr>
<th>Alice’s PC or PDA</th>
<th>Alice’s Phone</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call-ID: 123</td>
<td>Call-ID: 456</td>
<td></td>
</tr>
<tr>
<td>---REFER/202------&gt;</td>
<td>&lt;=Estab. dialog==&gt;</td>
<td></td>
</tr>
<tr>
<td>2 ---INVITE--------&gt; 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---NOTIFY/200------&gt; &lt;----200/ACK------- 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---NOTIFY/200------&gt; 5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Message 1:

Accept: application/dialog-info+xml
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: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

--------------------------------
<table>
<thead>
<tr>
<th>Remote Call Control Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>hold</td>
</tr>
<tr>
<td>call=</td>
</tr>
<tr>
<td>sip:line1@192.168.0.5</td>
</tr>
<tr>
<td>call-id:456</td>
</tr>
<tr>
<td>remote-tag=uvw</td>
</tr>
<tr>
<td>local-tag=xyz</td>
</tr>
<tr>
<td>other parameters</td>
</tr>
</tbody>
</table>

--------------------------------

Message 2:

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

SDP to indicate held media for example.

Message 3:

NOTIFY indicates "trying".

Message 4:

SIP/2.0 200 OK
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>
Content-Type: application/sdp
Content-Length: xxx

Message 5:

NOTIFY indicates "confirmed".
5.8 Retrieve Call

In message 1, Alice’s PC or PDA asks her phone to retrieve an held call with Bob. Alice’s phone sends a re-INVITE to Bob’s contact to resume the media stream which was already on hold. Note that a call hold is different concept than held media. In fact, a user can be placed on hold, and be provided with music on hold. A held call is a logical state which could be useful for a number of things such as monitoring the amount of time a user stays in a queue. This diagram does not illustrate any event package to illustrate that a call can be held.

<table>
<thead>
<tr>
<th>Alice’s PC or PDA</th>
<th>Alice’s Phone</th>
<th>Bob Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call-ID: 123</td>
<td>Call-ID: 456</td>
<td></td>
</tr>
<tr>
<td>1 ---REFER/202----&gt;</td>
<td>&lt;=Estab. dialog==&gt;</td>
<td></td>
</tr>
<tr>
<td>2 ---INVITE--------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;=NOTIFY/200------- 3</td>
<td>&lt;----200/ACK------ 4</td>
<td></td>
</tr>
<tr>
<td>&lt;=NOTIFY/200------- 5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Message 1:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remoteccc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

<table>
<thead>
<tr>
<th>Remote Call Control Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieve</td>
</tr>
<tr>
<td>Call=</td>
</tr>
<tr>
<td>sip:line1@192.168.0.5</td>
</tr>
<tr>
<td>call-id:456</td>
</tr>
<tr>
<td>remote-tag=uvw</td>
</tr>
<tr>
<td>local-tag=xyz</td>
</tr>
</tbody>
</table>
Message 2:

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
    SDP to indicate re-established media.

Message 3:

NOTIFY indicates "trying".

Message 4:

SIP/2.0 200 OK
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>
Content-Type: application/sdp
Content-Length: xxx

Message 5:

NOTIFY indicates "confirmed".

5.9 Conference Call

Alice’s Phone and Bob’s contact are currently in an established
dialog. Alice’s Phone and Cathy’s contact are also currently in an
established dialog. In message 1, Alice’s PC or PDA requests that a
call be established between the two calls (i.e., a conference
between Alice’s Phone, Bob’s contact and Cathy’s contact. Alice’s
phone establishes a call with a conference bridge (2-5). Alice’s phone
sends a request (6) to Bob’s contact to transfer the call to the same
conference bridge (7). Alice’s phone is notified (implicit REFER
subscription) of the successful transfer to the conference bridge (8)
and clears the call with Bob (9). Alice’s phone sends a request (10)
to Cathy’s contact to transfer the call to the same conference bridge (11). Alice’s phone is notified (implicit REFER subscription) of the successful transfer to the conference bridge (12) and clears the call with Cathy (13). The call flow does not show an event package for the successful remote conference invocation.

Message 1:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

Message 1:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

| Remote Call Control Body |
5.10 Single Step Conference Call

A single step conference call is the same operation as a conference call, except that one of the legs is a SIP URI instead of an established dialog. Alice’s Phone and Bob’s contact are currently in an established dialog. In message 1, Alice’s PC or PDA requests that a conference be established between the existing call with Bob’s contact and with Cathy (i.e., a conference between Alice’s Phone, Bob’s contact and Cathy). Alice’s phone establishes a call with a conference bridge (2-5). Alice’s phone sends a request (6) to Bob’s contact to transfer the call to the same conference bridge (7). Alice’s phone is notified (implicit REFER subscription) of the successful transfer to the conference bridge (8) and clears the call with Bob (9). Alice’s phone sends a request (10) to Cathy’s contact to transfer the call to the same conference bridge (11). Alice’s phone is notified (implicit REFER subscription) of the successful transfer to the conference bridge (12). The call flow does not show an event package for the successful remote single step conference invocation.
Message 1:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

------------------------
Remote Call Control Body
| conference
|   FirstCall=
|     sip:line1@192.168.0.5
|     call-id:456
|     remote-tag=uvw
|     local-tag=xyz
| SecondCall=
|     sip:cathy@example.net
| other parameters
------------------------
Message 3:

NOTIFY indicates "trying".

Message 5:

NOTIFY indicates "confirmed".

5.11 Set Do Not Disturb

In message 1, Alice sends a request so that her phone will be in "Do not disturb" or "Make set busy" mode. Any subsequent invitation (2) sent to Alice’s phone will result in the session being rejected with response 480 "Temporarily not available" (or 486 "Busy Here", or any other appropriate code) without any interaction from Alice’s PC or PDA.

Alice’s                                                                 Alice’s                                                                 Bob
PC or PDA                                                             Phone

<table>
<thead>
<tr>
<th>Call-ID: 123</th>
<th>Call-ID: 456</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  ---REFER/202-------</td>
<td>2 ---INVITE--------</td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td>3</td>
</tr>
<tr>
<td>4  &lt;---480/ACK----- &gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------</td>
<td>5</td>
</tr>
</tbody>
</table>

Message 1:

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
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...
Message 2:

INVITE sip:alice@example.com SIP/2.0
To: "Alice" <sip:alice@example.com>
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Bob’s Contact" <sip:line1@192.168.0.5>
Content-Type: application/sdp
Content-Length: xxx

Message 3:

NOTIFY indicates "trying".

Message 4:

SIP/2.0 480 Temporarily unavailable
To: "Alice" <sip:alice@example.com>;tag=uvw
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Alice’s Phone" <sip:reg2@10.1.1.3>

Message 5:

NOTIFY indicates "rejected".

5.12 Set Forwarding

In message 1, Alice sends a request so that her phone will be "call forwarded" to Cathy. Any subsequent invitation (2) sent to Alice’s phone will result in the session being forewared with response 302 "Move temporarily" without any interaction from Alice’s PC or PDA.

Alice’s PC or PDA  Alice’s Phone  Bob  Cathy
Message 1:

REFER sip:reg2@10.1.1.3 SIP/2.0
To: "Alice’s phone" <sip:reg2@10.1.1.3>;tag=d
From: "Alice’s PC or PDA" <sip:alice1@10.1.1.2>;tag=a
Call-ID: 123
CSeq: 2 REFER
Accept: application/dialog-info+xml
Require: extended-refer
Refer-To: cid:1239103912039@issuer.example.com
Content-Type: application/remotecc
Content-Id: <1239103912039@issuer.example.com>
Content-Length: ...

--------------------------------
| Remote Call Control Body     |
| setforwarding               |
|   destination               |
|     sip:cathy@example.net    |
|     forwardingtype=always    |
|   other parameters          |

Message 2:

INVITE sip:alice@example.com SIP/2.0
To: "Alice" <sip:alice@example.com>
From: "Bob" <sip:bob@example.net>;tag=x
Call-ID: 456
CSeq: 1 INVITE
Contact: "Bob’s Contact" <sip:line1@192.168.0.5>
Content-Type: application/sdp
Content-Length: xxx

Message 3:
NOTIFY indicates "trying".

Message 4:

SIP/2.0 302 Moved temporarily
To: "Alice" <sip:alice@example.com>;tag=uvw
From: "Bob" <sip:bob@example.net>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Cathy" <sip:cathy@example.net>

Message 5:

NOTIFY indicates "rejected".

Message 6:

INVITE sip:cathy@example.net SIP/2.0
To: "Cathy" <sip:cathy@example.net>
From: "Bob" <sip:bob@example.net>;tag=pqr
Call-ID: 789
CSeq: 1 INVITE
Contact: "Bob’s Contact" <sip:line1@192.168.0.5>
Content-Type: application/sdp
Content-Length: xxx

5.13 Alternate Call

Alternate call is not really an operation by itself. It is a hold operation followed by a retrieve operation. This section is included only to illustrate how those two operations can be combined to provide an Alternate Call service. In message 1, Alice’s PC or PDA asks her phone to put on hold the already established dialog with Bob. Alice’s phone sends a re-INVITE to Bob’s contact (2) to put the media stream on hold. Alice’s PC or PDA then asks her phone (6) to retrieve her previously held call with Cathy (7).

```
Alice’s PC or PDA  Alice’s Phone  Bob                Cathy
Call-ID: 123       Call-ID: 456       Call-ID: 789
10---REFER/202------> 10---INVITE------> 10---ESTAB. dialog--->
10---Call on hold-----------------> 10---Call on hold-----------------> 10---Call on hold----------------->
```
5.14 Consultation Call

Consultation call is not really an operation by itself. It is a hold operation followed by a make call operation. This section is included only to illustrate how those two operations can be combined to provide a Consultation Call service. In message 1, Alice’s PC or PDA asks her phone to put on hold the already established dialog with Bob. Alice’s phone. Alice’s phone sends a re-INVITE to Bob’s contact (2) to put the media stream on hold. Alice’s PC or PDA then asks her phone (6) to make a call to Cathy (7).

<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>Call-ID: 456</td>
<td>Call-ID: 789</td>
</tr>
<tr>
<td>1 ---REFER/202------&gt;</td>
<td></td>
<td>===Establish dialog===</td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------&gt;</td>
<td>2 ---INVITE--------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 &lt;----200/ACK-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 &lt;---200/ACK----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 ---REFER/202------&gt;</td>
<td>7 ---------------INVITE----------------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;--NOTIFY/200------&gt;</td>
<td>8 &lt;---200/ACK-----------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 &lt;---200/ACK------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 &lt;---200/ACK------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11 &lt;---200/ACK------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12 &lt;---200/ACK------------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. Examples of implementing remote call control operations with Refer-To URI

This section provided examples of how to implement some of the simple operations of the previous sections without using a REFER MIME body and relying instead of the Refer-To URI. All the call flows are assumed to be the same as per the previous section. Only the changed REFER message is shown.

6.1 Make Call Operation

This example is already discussed in a previous section.

6.2 Answer Call Operation

Message 1:

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
Accept: application/dialog-info+xml
Require: response-refer
Refer-To: "Bob's Contact" <sip:line1@192.168.0.5;method=INVITE;response=200?Call-ID=456&To=alice%40example.com;tag=uvw&From=bob%40example.net;tag=xyz>

6.3 Clear Connection

Message 1:

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
Accept: application/dialog-info+xml
Require: response-refer
Refer-To: "Bob's contact" <sip:line1@192.168.0.5;method=BYE?Call-ID=456&To=alice%40example.com;tag=uvw&From=bob%40example.net;tag=xyz>
6.4 Deflect Call

Message 5:

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
Accept: application/dialog-info+xml
Require: response REFER
Refer-To: "Bob’s Contact" <sip:line1@192.168.0.5;method=INVITE;response=302?
Call-ID=456&To=alice%40example.com;tag=uvw&From=bob%40example.net;tag=xyz>

6.5 Complete Transfer Between Calls

7. User Agent Behavior

7.1 Organizing requests within dialogs

REFER messages used for call transfer always 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 is complicated by the possible presence of 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 could 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 typically 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].

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 phone" <sip:reg2@10.1.1.3>;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

7.2 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 attempts to clarify what needs to be placed in 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 MUST be a valid Globally Routable Unique URI (GRUU) [9] for a single UA (a Contact URI). Contact URIs for a UA can be discovered by subscribing to the Registration Package [22] for the relevant AORs.

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 refer-response 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 appear in the triggered request if the request were initiated autonomously by the Refer-Receiver. A REFER triggering a standalone request or dialog starting request, could send to either an AOR or a Contact address, but typically to an AOR. 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 [7] 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 [8] as recommended in the Referred-By specification.

7.3 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. 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.

OPEN ISSUE: These parameter extensions should be incorporated in the REFER extensions draft.

8. Authorizing remote call control requests

User Agents MUST authorize all remote call control requests. Requests from user agents which can authenticate themselves (using Digest authentication, mutual TLS authentication, or S/MIME) as representing the AOR on the target UA SHOULD be authorized unless local policy directs otherwise. In addition, some user agents may need introduction using one-time credentials which have additional authorization restrictions. For example, an electronic whiteboard in a conference room could authorize participants only if they had scheduled a meeting in the corresponding conference room for the current time. [More explanation needed.]

9. 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 [11] via a SIPS URI, or individually signed using SMIME [10]. 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.

10. IANA Considerations

No action by IANA is required.

11. Acknowledgments

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

Normative References


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

Francois Audet
Nortel Networks
4655 Great America Parkway
Santa Clara, CA  95054
USA
EMail: audet@nortelnetworks.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 (2004). 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.

Acknowledgement

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