The Refer Method
draft-sparks-sip-refer-split-00

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 http://www.ietf.org/shadow.html.

This Internet-Draft will expire on October 25, 2002.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This document defines the REFER method. This SIP extension requests that the recipient REFER to a resource provided in the request. This can be used to enable many applications, including Call Transfer.
# Table of Contents

1. Overview ........................................... 3
2. Changes from draft-ietf-sip-refer-02 .................. 3
3. The REFER Method ................................... 3
4. Header Field Support for the REFER Method .......... 4
5. Message Body Inclusion .............................. 6
6. Behavior of SIP User Agents ......................... 6
   6.1 Accessing the referred-to resource ............... 6
   6.2 UA Responses within the REFER transaction ...... 6
   6.3 Reporting on the results of the reference ...... 7
   6.3.1 Using NOTIFY ................................ 7
   6.3.2 The body of the NOTIFY ....................... 7
   6.3.3 Multiple REFER requests in a dialog .......... 8
6.4 Behavior of SIP Registrars/Redirect Servers ...... 8
6.5 Behavior of SIP Proxies ........................... 9
7. Examples ........................................... 9
   7.1 Prototypical REFER callflow ..................... 9
   7.2 Multiple REFERs in a dialog .................... 10
8. Security Considerations ............................ 12
   8.1 Circumventing privacy ........................... 12
   8.2 Circumventing security .......................... 13
   8.3 Limiting the breach ............................. 13
9. Historic Material .................................. 13
10. IANA Considerations ................................. 14
11. Acknowledgments ................................... 14
References ............................................ 14
Author's Address ...................................... 15
Full Copyright Statement .............................. 16
1. Overview

This document defines the REFER method. This SIP [1] extension requests that the recipient REFER to a resource provided in the request. This can be used to enable many applications, including Call Transfer.

2. Changes from draft-ietf-sip-refer-02

This section will be removed before the draft is forwarded to the IESG

- Removed Referred-By to a separate draft.
- Repaired syntax errors (particularly escaping) in example URIs
- Changed syntax definitions to use RFC2234 ABNF
- Incorporated other last call feedback
- Replaced the header tables to match RFC3261
- Changed syntax of Refer-To to allow name-addr (not backwards compatible)
- Redo IANA section to align with Events/Presence
- Replaced the cseq= Event header parameter with the sip-events id= parameter when handling multiple subscriptions within a dialog

3. The REFER Method

REFER is a SIP method as defined by RFC3261 [1]. The REFER method indicates that the recipient (identified by the Request-URI) should be interpreted as if it appeared in Table 3 of RFC 3261.
The Refer-To header field MAY be encrypted as part of end-end encryption.

Examples

Refer-To: sip:alice@atlanta.com

Refer-To: sip:bob@biloxi.com?Accept-Contact;sip:bobsdesk.
   biloxi.com&amp;Call-ID%3D55432%40alicepc.atlanta.com

Refer-To: sip:dave@denver.com?Replaces%3D12345%40192.168.118.3%3B
to-tag%3D12345%3Bfrom-tag%3D5FFE-3994

Refer-To: sip:carol@cleveland.com;method=SUBSCRIBE

Refer-To: http://www.ietf.org

Long headers field values are line-wrapped here for clarity only.

4. Header Field Support for the REFER Method

This table adds a column to tables 4 and 5 in [1], describing header field presence in a REFER method. See [1] for a key for the symbols used. A row for the Refer-To: request-header should be inferred, each mandatory for REFER. Refer-To is not applicable for any other methods. The enc and e-e columns in [1] apply to the REFER method unmodified.
<table>
<thead>
<tr>
<th>Header</th>
<th>Where</th>
<th>REFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
<td>R</td>
<td>o</td>
</tr>
<tr>
<td>Accept</td>
<td>2xx</td>
<td>-</td>
</tr>
<tr>
<td>Accept</td>
<td>415</td>
<td>c</td>
</tr>
<tr>
<td>Accept-Encoding</td>
<td>R</td>
<td>o</td>
</tr>
<tr>
<td>Accept-Encoding</td>
<td>2xx</td>
<td>-</td>
</tr>
<tr>
<td>Accept-Encoding</td>
<td>415</td>
<td>c</td>
</tr>
<tr>
<td>Accept-Language</td>
<td>R</td>
<td>o</td>
</tr>
<tr>
<td>Accept-Language</td>
<td>2xx</td>
<td>-</td>
</tr>
<tr>
<td>Accept-Language</td>
<td>415</td>
<td>c</td>
</tr>
<tr>
<td>Alert-Info</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Allow</td>
<td>Rr</td>
<td>o</td>
</tr>
<tr>
<td>Allow</td>
<td>405</td>
<td>m</td>
</tr>
<tr>
<td>Authentication-Info</td>
<td>2xx</td>
<td>o</td>
</tr>
<tr>
<td>Authorization</td>
<td>R</td>
<td>o</td>
</tr>
<tr>
<td>Call-ID</td>
<td>c</td>
<td>m</td>
</tr>
<tr>
<td>Call-Info</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Contact</td>
<td>R</td>
<td>m</td>
</tr>
<tr>
<td>Contact</td>
<td>1xx</td>
<td>-</td>
</tr>
</tbody>
</table>

Sparks
Expires: October 25, 2002
5. Message Body Inclusion

A REFER method MAY contain a body. This specification assigns no meaning to such a body. A receiving agent may choose to process the body according to its Content-Type.

6. Behavior of SIP User Agents

6.1 Accessing the referred-to resource

A UA accepting a well-formed REFER request SHOULD request approval from the user to proceed (this request could be satisfied with an interactive query or through accessing configured policy). If approval is granted, the UA MUST contact the resource identified by
the URI in the Refer-To: header field value. Note that if the URI is a SIP URI, it could contain header fields such as Call-Id that may be used to form the resulting request.

The resource identified by the Refer-To: URI is contacted using the normal mechanisms for that URI type. For example, if the URI is a SIP URI indicating an INVITE should be generated (using a method=INVITE URI parameter for example), the UA would issue a new INVITE using all of the normal rules for sending an INVITE defined in [1].

6.2 UA Responses within the REFER transaction

If the approval sought above for a well formed REFER request is immediately denied, the UA MAY decline the request.

An agent responding to a REFER Method MUST return a 400 Bad Request if the request contained zero or more than one Refer-To header field values.

An agent (including proxies generating local responses) MAY return a 100 Trying or any appropriate 400-600 class response as prescribed by [1].

Care should be taken when implementing the logic that determines whether or not to accept the REFER request. A UA not capable of accessing non-SIP URIs SHOULD NOT accept REFER requests to them.

If no final response has been generated according to the rules above, the UA MUST return a 202 Accepted response before the REFER transaction expires.

6.3 Reporting on the results of the reference

6.3.1 Using NOTIFY

If a REFER request is accepted (with a 202 Accepted response), the UA receiving the REFER SHOULD notify the agent sending the REFER of the status of the reference. This is done using the NOTIFY mechanism
defined in [2] as if the REFER had established a subscription. In particular:

- Each NOTIFY should reflect the To:, From:, and Call-ID header fields from the REFER as if they had arrived in a SUBSCRIBE.

- Each NOTIFY MUST contain an event header field of Event: refer

- Each NOTIFY MUST contain a body of type "message/sipfrag" [3].

- Analogous to the case for SUBSCRIBE described in [2], the agent that issued the REFER MUST be prepared to receive a NOTIFY before the REFER transaction completes.

6.3.2 The body of the NOTIFY

Each NOTIFY MUST contain a body of type "message/sipfrag" [3]. The body of a NOTIFY MUST begin with a SIP Response Status-Line as defined in [1]. The response class in this status line indicates the status of the referred action. The body MAY contain other SIP header fields to provide information about the outcome of the referenced action.

A minimal, but complete, implementation can respond with a single NOTIFY containing either the body:

SIP/2.0 200 OK

if the reference was successful, the body:

SIP/2.0 503 Service Unavailable

if the reference failed, or the body:
SIP/2.0 603 Declined

if the REFER request was accepted before approval to follow the reference could be obtained and that approval was subsequently denied.

An implementation MAY include more of a SIP message in that body to convey more information. Warning header field values received in responses to the referred action are good candidates. In fact, if the reference was to a SIP URI, the entire response to the referenced action could be returned (perhaps to assist with debugging). However, doing so could have grave security repercussions (see Section 8). Implementers must carefully consider what they choose to include.

Note that if the reference was to a non-SIP URI, status in any NOTIFYs to the referrer must still be in the form of SIP Response Status-Lines. The minimal implementation discussed above is sufficient to provide a basic indication of success or failure. For example, if a client receives a REFER to a HTTP URL, and is successful in accessing the resource, its NOTIFY to the referrer can contain the message/sipfrag body of "SIP/2.0 200 OK". If the notifier wishes to return additional non-SIP protocol specific information about the status of the request, it may place it in the body of the sipfrag message.

6.3.3 Multiple REFER requests in a dialog

A REFER creates an implicit subscription sharing the dialog identifiers in the REFER request. If more than one REFER is issued in the same dialog (a second attempt at transferring a call for example), the dialog identifiers do not provide enough information to associate the resulting NOTIFYs with the proper REFER.

Thus, for the second and subsequent REFER requests a UA receives in a given dialog, it MUST include an id parameter[2] in the Event header field of each NOTIFY containing the sequence number (the number from the CSeq header field value) of the REFER this NOTIFY is associated with. This id parameter MAY be included in NOTIFYs to the first REFER a UA receives in a given dialog.

6.4 Behavior of SIP Registrars/Redirect Servers

Registrars and Redirect Servers SHOULD return a 603 to a REFER request, unless they are also playing some other SIP role.
6.5 Behavior of SIP Proxies

SIP Proxies do not require modification to support the REFER method. Specifically, as required by [1], a proxy should process a REFER request the same way it processes an OPTIONS request.

7. Examples

7.1 Prototypical REFER callflow

Here are examples of what the four messages between Agent A and Agent B might look like if the reference to (whatever) that Agent B makes is successful. The details of this flow indicate this particular REFER occurs outside a session (there is no To: tag in the REFER request). If the REFER occurs inside a session, there would be a non-empty To: tag in the request.

Message One (F1)

REFER sip:b@agentland SIP/2.0
Via: SIP/2.0/UDP agenta.agentland;branch=2293940223
To: <sip:b@agentland>
From: <sip:a@agentland>;tag=193402342
Call-ID: 898234234@agenta.agentland
CSeq: 93809823 REFER
Max-Forwards: 70
Refer-To: (whatever URI)
Contact: sip:a@agentland
Content-Length: 0

Message Two (F2)

SIP/2.0 202 Accepted
Via: SIP/2.0/UDP agenta.agentland;branch=2293940223
To: <sip:b@agentland>;tag=4992881234
From: <sip:a@agentland>;tag=193402342
Call-ID: 898234234@agenta.agentland
CSeq: 93809823 REFER
Contact: sip:b@agentland
Content-Length: 0

Message Three (F3)

NOTIFY sip:a@agentland SIP/2.0
Via: SIP/2.0/UDP agentb.agentland;branch=9922ef992-25
To: <sip:a@agentland>;tag=193402342
From: <sip:b@agentland>;tag=4992881234
Call-ID: 898234234@agenta.agentland
CSeq: 1993402 NOTIFY
Max-Forwards: 70
Event: refer
Contact: sip:b@agentland
Content-Type: message/sipfrag;version=2.0
Content-Length: 16

SIP/2.0 200 OK

Message Four (F4)

SIP/2.0 200 OK
Via: SIP/2.0/UDP agentb.agentland;branch=9922ef992-25
To: <sip:a@agentland>;tag=193402342
From: <sip:b@agentland>;tag=4992881234
Call-ID: 898234234@agenta.agentland
CSeq: 1993402 NOTIFY
Contact: sip:a@agentland
Content-Length: 0
7.2 Multiple REFERs in a dialog

Message One above brings an implicit subscription dialog into existence. Suppose Agent A issued a second REFER inside that dialog:

```
F5 REFER
F6 202 Accepted

F7 NOTIFY
F8 200 OK
```

Message Five (F5)

```
REFER sip:b@agentland SIP/2.0
Via: SIP/2.0/UDP agenta.agentland;branch=9390399231
To: <sip:b@agentland>;tag=4992881234
From: <sip:a@agentland>;tag=193402342
Call-ID: 898234234@agenta.agentland
CSeq: 93809824 REFER
Max-Forwards: 70
Refer-To: (some different URI)
Contact: sip:a@agentland
Content-Length: 0
```

Message Six (F6)
SIP/2.0 202 Accepted
Via: SIP/2.0/UDP agenta.agentland;branch=9390399231
To: <sip:b@agentland>;tag=4992881234
From: <sip:a@agentland>;tag=193402342
Call-ID: 898234234@agenta.agentland
CSeq: 93809824 REFER
Contact: sip:b@agentland
Content-Length: 0

Message Seven (F7)

Sparks                  Expires October 25, 2002               
Internet-Draft              The Refer Method                  April 2002

NOTIFY sip:a@agentland SIP/2.0
Via: SIP/2.0/UDP agentb.agentland;branch=2994a93eb-fe
To: <sip:a@agentland>;tag=193402342
From: <sip:b@agentland>;tag=4992881234
Call-ID: 898234234@agenta.agentland
CSeq: 1993403 NOTIFY
Max-Forwards: 70
Event: refer;id=93809824
Contact: sip:b@agentland
Content-Type: message/sipfrag;version=2.0
Content-Length: 16

SIP/2.0 200 OK

Message Eight (F8)

SIP/2.0 200 OK
Via: SIP/2.0/UDP agentb.agentland;branch=2994a93eb-fe
To: <sip:a@agentland>;tag=193402342
From: <sip:b@agentland>;tag=4992881234
Call-ID: 898234234@agenta.agentland
CSeq: 1993403 NOTIFY
Contact: sip:a@agentland
Content-Length: 0
8. Security Considerations

The security considerations of [1] apply to the REFER method.

This mechanism relies on providing contact information for the referred-to resource to the party being referred. Care should be taken to provide a suitably restricted URI if the referred to resource should be protected.

Using message/sipfrag bodies to return the progress and results of a REFER request is extremely powerful. Careless use of that capability will compromise security and privacy. Here are a couple of simple, somewhat contrived, examples to demonstrate the potential for harm.

8.1 Circumventing privacy

Suppose Alice has a user-agent that accepts REFER requests to SIP INVITE URIs, and NOTIFYs the referrer of the progress of the INVITE by copying each response to the INVITE into the body of a NOTIFY.

Suppose further that Carol has a reason to avoid Mallory and has configured her system at her proxy to only accept calls from a certain set of people she trusts (including Alice), so that Mallory doesn't learn when she's around, or what user agent she's actually using.

Mallory can send a REFER to Alice, with a Refer-To: indicating Carol. If Alice can reach Carol, the 200 OK Carol sends gets returned to Mallory in a NOTIFY, letting him know not only that Carol is around, but also the IP address of the agent she's using.

8.2 Circumventing security

Suppose Alice, with the same user agent as above, is working at a company that is working on the greatest SIP device ever invented - the SIP FOO. The company has been working for months building the device and the marketing materials, carefully keeping the idea, even the name of the idea secret (since a FOO is one of those things that anybody could do if they'd just had the idea first). FOO is up and
running, and anyone at the company can use it, but it's not available outside the company firewall.

Mallory has heard rumor that Alice's company is onto something big, and has even managed to get his hands on a URI that he suspects might have something to do with it. He sends a REFER to ALICE with the mysterious URI and as Alice connects to the FOO, Mallory gets NOTIFYs with bodies containing

Server: FOO/v0.9.7

8.3 Limiting the breach

For each of these cases, and in general, returning a carefully selected subset of the information available about the progress of the reference through the NOTIFYs mitigates risk. The minimal implementation described in Section 6.3.2 exposes the least information about what the agent operating on the REFER request has done, and is least likely to be a useful tool for malicious users.

9. Historic Material

This method was initially motivated by the call-transfer application. Starting as TRANSFER, and later generalizing to REFER, this method improved on the BYE/Also concept of the expired draft-ietf-sip-cc-01 by disassociating transfers from the processing of BYE. These changes facilitate recovery of failed transfers and clarify state management in the participating entities.

Early versions of this work required the agent responding to REFER to wait until the referred action completed before sending a final response to the REFER. That final response reflected the success or failure of the referred action. This was infeasible due to the transaction timeout rules defined for non-INVITE requests in [1]. A REFER must always receive an immediate (within the lifetime of a non-INVITE transaction) final response.

10. IANA Considerations

(Note to RFC Editor: Please fill in all occurrences of XXXX in this
This document defines a new SIP method name (REFER), a new SIP header name with a compact form (Refer-To and r respectively), and an event package (refer).

SIP Method names are not currently registered with IANA.

The following information should be added to the header sub-registry under [62x646] http://www.iana.org/assignments/sip-parameters. [62x646].

Header Name: Refer-To

Compact Form: r

Reference: RFC XXXX

This specification registers an event package, based on the registration procedures defined in [2]. The following is the information required for such a registration:

Package Name: refer

Package or Package-Template: This is a package.

Published Specification: RFC XXXX

Person to Contact: Robert Sparks, rsparks@dynamicsoft.com

11. Acknowledgments

This draft is a collaborative product of the SIP working group.

References


Author's Address

Robert J. Sparks
dynamicsoft
5100 Tennyson Parkway
Suite 1200
Plano, TX  75024

EMail: rsparks@dynamicsoft.com