

SIPPING

Internet Draft

Document: [draft-parameswar-sipping-norefersub-00.txt](#)

Expires: April 2003

S. Parameswar

October 2002

Suppressing Refer Implicit Subscription

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#) [1].

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>.

Abstract

The recipient of the REFER method [1] creates an implicit subscription to the 'refer' event. This subscription causes the REFER recipient to send NOTIFY messages to the sender informing him of the progress of the REFER. This memo provides for the requirements and one potential mechanism to suppress the creation of this implicit subscription, via an option tag - 'norefersub'. This gives the agent sending the REFER control over the creation of the subscription, while being backwards compatible with [1].

Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

Suppressing Refer Implicit Subscription October 2002

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].

This document uses the terms "referor" for the UA that sends the REFER method, and "referee" for the receiving agent.

Table of Contents

1.	Introduction.....	2
1.1	Implicit subscription in REFER.....	2
2.	Suppressing Implicit subscriptions.....	3
2.1	Requirements for suppression of Implicit subscriptions.....	4
3.	Potential mechanisms.....	4
3.1	Immediate Un-subscription.....	4
3.2	New Option Tag.....	5
4.	Usage of the 'norefersub' tag.....	6
4.1	UAS (referee) behavior.....	6
4.2	UAC (referor) behavior.....	7
5.	IANA Registration of the 'norefersub' Option Tag.....	7
	Security Considerations.....	8
	References.....	8
	Acknowledgments.....	8
	Author's Addresses.....	8

[1.](#) Introduction

[RFC 3265](#) [2] allows for non-SUBSCRIBE mechanisms to create subscriptions, these are popularly called Implicit subscriptions. This mechanism is used in [1] to create an implicit subscription when a REFER method is received by a UA. The UA responds with an immediate NOTIFY, and may send further NOTIFYS describing the progress till the subscription expires.

The referor has no control over the creation of the subscription. There has also been some acknowledgement that the implicit subscription mechanism is not desirable. This memo defines a mechanism to allow the referor control over the creation of the implicit subscription caused by sending the REFER to the referee.

[1.1](#) Implicit subscription in REFER

Figure 1 below, shows a typical scenario. The REFER (F1) from the referor causes an implicit subscription to be created at the referee. The immediate NOTIFY (F3) is required as stated in [RFC 3265](#)[2] and provides the state of the subscription and the status associated with the event. This is followed by one or more NOTIFY messages describing

Suppressing Refer Implicit Subscription October 2002

the progresss, till the expiry of the subscription - as indicated by the Subscription-State header in the final NOTIFY.

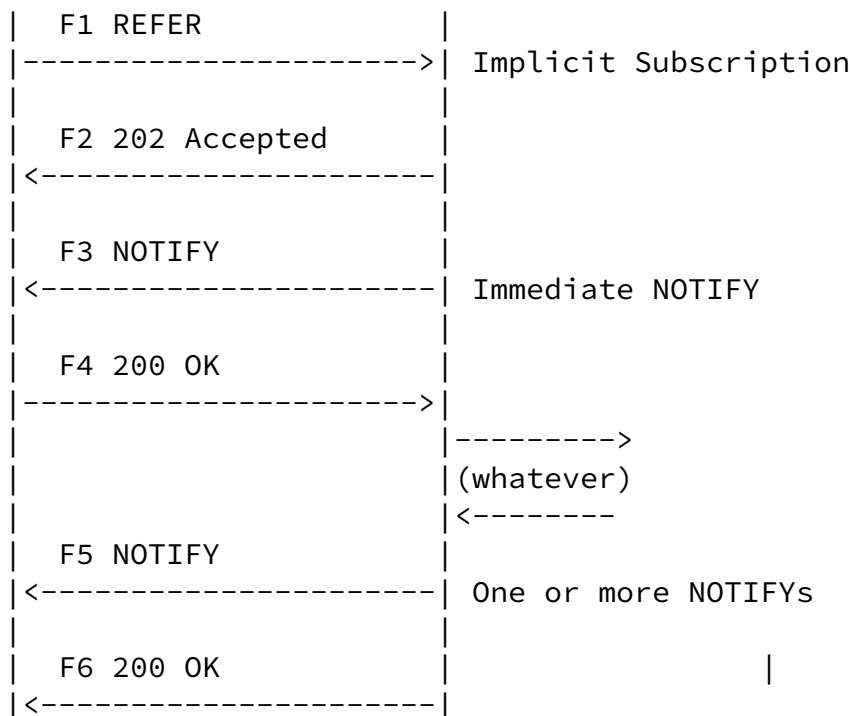


Figure 1. Implicit subscription caused by REFER

2. Suppressing Implicit subscriptions

As can be seen from Figure 1, the referor is neither in control of the creation of the implicit subscription on its behalf, nor of the number and rate of NOTIFYs. The matter of rate and number of NOTIFYs may be handled by subscription filters carried in the payload of the REFER method. It must be noted that this topic of subscription filters is a subject of ongoing work. In the case of implicit subscription as it stands today, the referee is in complete control.

In addition to the referor control of subscription, there are also some applications that may not need an implicit subscription to be created, either because the referor does not care or the Refer-To resource does not lend itself to providing a notion of success or failure. An example is discussed below to provide a use case:

Web-push: The REFER method may be used to push web pages for example the Refer-To header may carry the address `http://www.example.com`. In this case the referee may not wish to browse the pushed page immediately and a NOTIFY may be delayed or not forthcoming.

Attended Transfers: This is a weaker example - enterprise setting where all parties are served off the same

Suppressing Refer Implicit Subscription October 2002

proxy/registrar, the need to suppress implicit subscription exists. In an attended transfer scenario the transferor [4] first informs the transfer target of the impending transfer prior to completing the action. In such cases the transferor is reasonably sure that the REFER will succeed (transfer target contacted and ready, etc.) and the notification may not be required.

It must also be noted that control of implicit subscriptions provides the opportunity to provide differentiated services. For example, a service provider may offer Basic Transfer and Enhanced Transfer with Status Notification.

[2.1](#) Requirements for suppression of Implicit subscriptions

This section looks at the requirements for the suppression of implicit subscriptions caused by REFER. These formally address the requirements that were laid out in [Section 2](#) in an anecdotal fashion.

1. Referor-control-req: The referor (party initiating the REFER method) MUST be able to control creation of the implicit subscription. This provides the referor the ability to create or suppress the implicit subscription.
2. Session-integrity-req: In the case where the referee (party receiving the REFER method) chooses to, or does not support the suppression of implicit subscription, the session MUST not be affected. This allows the referor to retry the REFER based on the ability or choice of the referee.

3. Backwards-compatibility-req: Any mechanism that provides for suppression of implicit subscriptions MUST be backwardly compatible.
4. Simplicity-req: The mechanism MUST be easy to implement. As such mechanisms that use existing SIP functionality are preferred.
5. Congestion-limitation-req: Any mechanism that provides for suppression of implicit subscriptions SHOULD NOT create additional signaling burden on the network.

3. Potential mechanisms

This section details a couple of potential mechanisms to suppress implicit subscriptions. Each of these mechanisms are discussed in light of the requirements in [Section 2.1](#).

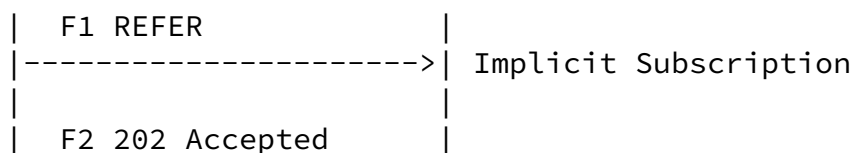
3.1 Immediate Un-subscription

Suppressing Refer Implicit Subscription October 2002

This section takes a look at immediately un-subscribing, following the first NOTIFY as an alternate to suppressing the implicit subscription mechanism.

This is depicted in Figure 2. The REFER (F1) from the referor creates an implicit subscription. This causes the referee to generate an immediate NOTIFY (F3) after sending an acknowledgement to the REFER (F2). The referor not wanting further notification simply sends a 481 "Subscription terminated" (F4) as per [Section 3.2.2](#). 'Notifier NOTIFY Behavior' of [RFC 3265](#)[2]. The 481 terminates the subscription and the referee will send no further NOTIFYs.

However this does not create full control over the implicit subscription and is a waste of bandwidth - which is important in bandwidth-constrained environments such as wireless, low-speed dialup etc.



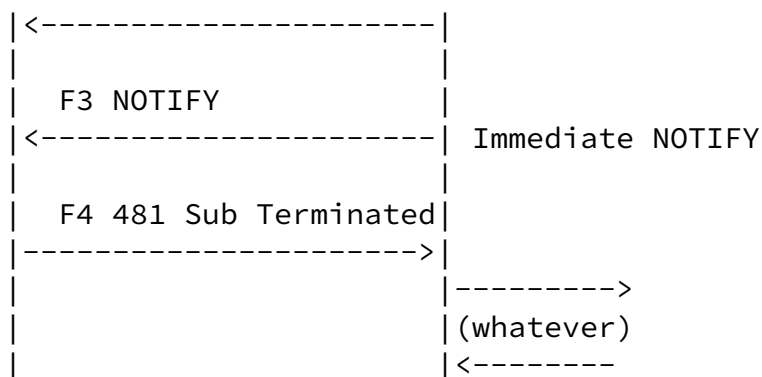


Figure 2. Un-subscribing after immediate NOTIFY

This option does not meet all the requirements laid out in [Section 2.1](#). The fact that the referor receives a NOTIFY and has to send a 481 violates the Referor-control-req and Congestion-limitation-req.

[3.2](#) New Option Tag

Given the discussion above - the author feels that there is definite value to being able to suppress implicit subscription in the case of the REFER message. This takes the form of a new option-tag "norefersub". In the simplest case the option-tag is sent in a Require header from the referor to the referee in a REFER message.

Example usage:

Required: norefersub

Supported: norefersub

Unsupported: norefersub

This option-tag may be applied as described in [3]. This memo discusses its use in the REFER method and 2xx messages that acknowledge the REFER method.

This mechanism meets all the requirements laid out in [Section 2.1](#). The use of this mechanism is backwards compatible; in the absence of this option-tag the original REFER behavior will result i.e. the implicit subscription is created. This mechanism does not result in additional signaling burden on the network.

[4.](#) Usage of the 'norefersub' tag

A simple example of the use of the 'norefersub' tag is shown in Figure 3. The REFER (F1) carries this tag in the Require header, this causes the suppression of the implicit subscription.

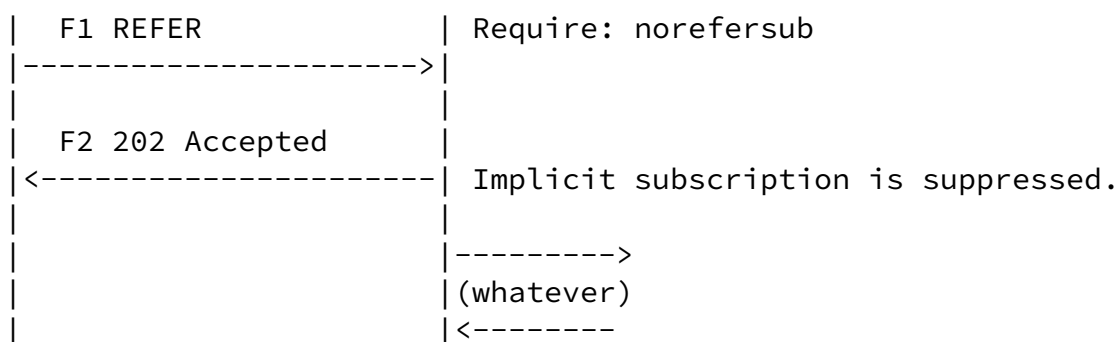


Figure 3. Using the option-tag 'norefersub'

```

Message One (F1)
REFER sip:b@agentland SIP/2.0
Via: SIP/2.0/UDP agenta.agentland;branch=z9hG4bK2293940223
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)
Require: norefersub
Contact: sip:a@agentland
Content-Length: 0

```

[4.1](#) UAS (referee) behavior

The UAS MUST suppress implicit subscriptions if the REFER request contained a Require header field with the option tag 'norefersub'.

If the UAS is unwilling to do so, it MUST reject the REFER request with a 420 (Bad Extension) and include an Unsupported header field containing the option tag 'norefersub'.

If the REFER contained a Supported header with the option tag 'norefersub', the UAS (referee) MAY decide to suppress the implicit subscription. In this case the UAS will send the option tag

'norefersub' in a Require header of the 2xx response to the REFER. This indicates to the referor that no NOTIFYs are to be expected for this REFER. However in this case if the 2xx does not contain the Require header with the 'norefersub' option-tag, the referor MUST be prepared to receive the NOTIFY messages from the UAS.

[4.2](#) UAC (referor) behavior

When the UAC (referor) creates a new REFER request, it can insist on suppressing implicit subscription for that request. To do that, it inserts a Require header field with the option tag 'norefersub' into the REFER request.

If the UAC wishes to leave the decision to create or suppress implicit subscription to event 'refer' in the hands of the UAS (referee) - it SHOULD include the Supported header with the option tag 'norefersub' in the REFER method. The presence of the option tag in the Require header of the 2xx response, indicates whether the UAS has decided to create or suppress the implicit subscription. If the option tag is present in the Require header then the UAC MUST not wait for the NOTIFY, and in the absence of this option-tag it MUST be prepared to receive the NOTIFY message for event 'refer'.

[5.](#) IANA Registration of the 'norefersub' Option Tag

This memo registers a single option tag, 'norefersub'. The required information for this registration, as specified in [RFC 3261](#), is:

Name: norefersub

Description: This option tag is for suppression of implicit subscriptions to event 'refer', caused by the reception of a REFER method. When present in a Supported header, it indicates that the UA can support/handle such suppression. When present in a Require header in a REFER request, it indicates that the UAS MUST suppress the implicit subscription.
When present in a Require header in a 2xx Response to the REFER, it indicates that the implicit

subscription has been suppressed.

Security Considerations

One of the advantages to implicit subscription is the case of forked REFERs - - of course, a REFER sent within the scope of an existing dialog will not fork. A REFER sent outside the context of a dialog MAY fork and the receipt of multiple NOTIFYs alerts the referor to this fact.

The use of implicit subscription suppression is advocated in those cases where the referor does not care about forked REFERs or the service being invoked via the REFER does not lend itself to sending back NOTIFYs.

References

- 1 Sparks, R., " The SIP Refer Method", Work In Progress, June 2002.
- 2 Roach, A., " Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002
- 3 Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- 4 Sparks, R., " SIP Call Control - Transfer", Work In Progress, July 2001.

Acknowledgments

Author gratefully acknowledges comments and support from Robert Sparks.

Author's Addresses

Sriram Parameswar
Phone: 214-929-1608
Email: sriramkp@attbi.com

