Abstract

This document lists several known and potential use cases for SIP INFO requests, as discussed on the SIP WG mailing list. The use cases were requested by the WG chairs at the SIP WG meeting in IETF 70, and are documented herein for the purpose of discussion only.

Table of Contents

1. Introduction................................................3
2. Terminology................................................3
3. Applicability...............................................3
4. Alternative Mechanisms....................................3
5. Documented Uses of INFO..................................4
   5.1. SIP-T................................................4
   5.2. ISUP/QSIG Interworking and Exchange..................4
   5.3. Media Server Control Markup Language (MSCML)........4
   5.4. User-Agent Computer Supported Telecommunications
        Applications (uaCSTA)..................................4
6. Potentially Valid Use-Cases................................4
   6.1. DTMF................................................4
   6.2. Sending a vcard asynchronously........................5
   6.3. Sending a user-icon image............................5
   6.4. Sending a vcalendar invitation.......................5
   6.5. Sending an HTTP URL................................6
   6.6. Performing a session traceroute........................6
   6.7. Sending soft-key labels and press events............6
   6.8. Sending geo-location information during call........7
7. Other Known Use-Cases.....................................7
   7.1. Sending RTP/RTCP statistics during call............7
   7.2. Sending access-location information after call
        establishment........................................7
   7.3. Sending media-control indications....................8
   7.4. Sending video fast update command....................8
   7.5. Sending peripheral control commands..................8
   7.6. Sending charging information for a call.............8
   7.7. Sending a screen-pop-up message.......................8
   7.8. Sending Bookmark Tags................................8
   7.9. Sending Hook-Flash Indication.........................9
   7.10. Session Refresh and Liveness Check...................9
   7.11. Message-Waiting Indication (MWI).....................9
   7.12. Media Server Markup Language (MSML)................9
8. Security Considerations...................................9
9. IANA Considerations......................................10
10. Informative References....................................10
Author’s Address............................................11
Intellectual Property Statement..............................12
1. Introduction

The SIP INFO method was defined in [RFC2976] to convey session related control information inside an Invite-created dialog. While it has been widely adopted for specific application use cases, and a couple RFCs rely on it, there are known limitations and issues with it today documented in [info-harmful]. Some of those issues, dealing with negotiation of use and context indication, are addressed in [info-events].

At IETF 70, the SIP WG chairs asked for a list of use-cases, for the purpose of deciding if a general framework such as [info-events] is warranted. This draft lists such use-cases, based on email discussion on the SIP WG mailing list sip@ietf.org. This draft is NOT suggesting the use-cases are legitimate or proper for SIP INFO use - that is to be discussed on the SIP WG mailing list.

2. Terminology

The terminology in this document conforms to RFC 2828, "Internet Security Glossary".

3. Applicability

This draft relates to the [RFC2976] SIP INFO request.

4. Alternative Mechanisms

In general, most use-cases for INFO have alternative mechanisms, which may or may not be more appropriate. Examples are:

1. Use Re-INVITE or UPDATE requests - for some cases this may make sense, but there is a question if INVITE or the [RFC3311] UPDATE method should be used as just carriers for things not affecting/updating the state of the session/dialog or SDP. (note that session-timers uses them for this though) For example, in some systems INVITE and UPDATE are treated differently for Call-Detail-Records and logging.

2. Use MESSAGE request - the MESSAGE request, defined in [RFC3428], is considered to be generally useful for sending MIME body content to be rendered for human users. It requires support for text/plain bodies, and is commonly expected to
handle message/cpim bodies, but it could transport any body type. There is an argument that MESSAGE is for human-rendered info, so vcards and vcalendars are inappropriate. However one could argue both are in fact rendered to the user, after processing based on their type, but that some UA’s can offer to skip rendering and do something else with them, again based on their type (like email clients do).

3. Use SUBSCRIBE/NOTIFY - [RFC3265] was written to handle scenarios similar to some of the use-cases listed herein.

4. Use a media-plane protocol, such as RTCP or MSRP - several use-cases in this draft should clearly be handled in the media-plane. Potentially any use-case could in theory be done by media, but clearly for some doing so represents a huge amount of overhead.

5. Documented Uses of INFO

5.1. SIP-T

This is documented in [RFC3372], which is a BCP.

5.2. ISUP/QSIG Interworking and Exchange

This is documented in [ECMA-355].

5.3. Media Server Control Markup Language (MSCML)

This is documented in [RFC5022], which is Informational.

5.4. User-Agent Computer Supported Telecommunications Applications (uaCSTA)

This is documented in [ECMA-TR/87].

6. Potentially Valid Use-Cases

The following represent known or potential use-cases which this author believes have reasonable applicability to INFO use.

6.1. DTMF

Several proprietary uses of INFO for transferring DTMF are known to exist, some of which are in wide use by multiple vendors. Alternative mechanisms: [KPML] for signaling-plane, or [RFC4733] for media-plane if possible.
6.2. Sending a vcard asynchronously

Example: Alice calls Bob, Alice says "can you send me John’s vcard?", Bob clicks something and voila it’s sent.

Reasons for INFO: it’s explicit what you’re doing when you send the vcard, and you can send it knowing the other end can accept it, and you can send it based on user input. Making it explicit means the receiving UA can automatically store the vcard for later use, for example into a contact list.

Alternative mechanisms: send a re-INVITE or UPDATE with a Call-Info, with either a URL reference, data URI, or MIME and CID URL; send a MESSAGE request.

Counter-Arguments to alternative mechanisms: sending it in an INVITE/UPDATE Call-Info would conflict with the purpose of that header being for caller/clee information only (as opposed to third party info, which this example shows). Furthermore, the DATA URL is generally discouraged, and a URL reference is hard to actually do in practice.

6.3. Sending a user-icon image

Example: Alice Alice calls Bob, Bob has an icon that represents himself, sends it when he picks up the phone or upon clicking what image he wants to represent himself.

Reasons for INFO: it’s explicit what you’re doing when you send the image, and you can send it knowing the other end can accept it and its type (jpeg/gif/bmp/etc.), and you can send it based on user input. Making it explicit means the receiving UA can automatically store the image for later use, for example as a contact/buddy image, which sending it in a MESSAGE would not do.

Alternative mechanisms: send a 200-ok, re-INVITE or UPDATE with a Call-Info, which has an explicit type for "icon"; send a MESSAGE request. P. Kyzivat notes MESSAGE is appropriate. J. Rosenberg notes that if the image is big, which it easily could be, media-path makes more sense.

6.4. Sending a vcalendar invitation

Example: Alice calls Bob, Bob says "hey let’s have a con call at time X", clicks and voila his phone sends a vcalendar.

Reasons for INFO: it’s explicit what you’re doing when you send the vcalendar, and you can send it knowing the other end can accept it,
and you can send it based on user input. Making it explicit means the receiving UA can automatically store the calendar invite time, for example if it has an integrated calendar app or alarm reminder.

Alternative mechanisms: send a MESSAGE request.

[Editor’s note: Whether the vcalendar is related to the session or not and thus whether it should be sent in an in-dialog request or not is certainly debatable, and makes MESSAGE more reasonable.]

6.5. Sending an HTTP URL

Example: Alice calls Bob, a sales guy; Alice asks for more info or a datasheet and Bob sends a URL for Alice to open with her web-browser.

Reasons for INFO: it’s explicit what you’re doing when you send the URL, and you can send it knowing the other end can accept it, and you can send it based on user input.

Alternative mechanisms: send a MESSAGE request. J. Rosenberg notes this can be done with a REFER with http URL based on [app-framework].

6.6. Performing a session traceroute

Example: Alice calls Bob, Bob answers, Alice does a sip-traceroute to figure out the path to Bob, by sending Info with an incrementing max-forwards type header starting at 0 (but not really the Max-Forwards header), with a sip-frag type response body or some such. The reason it’s not just using the Max-Forwards is because the 483 response generated by normal proxies would terminate the dialog. This wiould be a new header similar in concept, but only work for middle-boxes which support it and don’t generate a 483.

Reasons for INFO: If such a thing were to be defined, it could be done such that the response code did not terminate the dialog.

Alternative mechanisms: re-INVITE or UPDATE.

[Editor’s note: It’s debatable if certain types of B2BUA’s (ie, SBCs) would ever allow this type of thing to happen, due to security concerns, but I think they may do it at domain boundary hops.]

6.7. Sending soft-key labels and press events

Example: Alice calls her vmail server. Vmail server sends softkey-labels for the menu items available in the response or INFO, Alice presses softkeys and her UA sends them in INFO.
Reasons for INFO: similar to DTMF, the probability of users actually pressing the soft-key buttons is very low, so using INFO reduces SUBSCRIBE/NOTIFY overhead and ties it to the INVITE dialog implicitly.

Alternative mechanisms: SUBSCRIBE/NOTIFY similar to [KPML].
[Note: J. Rosenberg notes this in the [app-framework] draft scope]

6.8. Sending geo-location information during call

Example: Alice calls Bob, a hotel receptionist. Alice asks for directions to hotel, clicks button and sends him location info of her phone (or Bob clicks button and sends her his location). Or Alice calls emergency services from a mobile phone, and phone updates location based on GPS.

Reasons for INFO: it’s explicit what you’re doing when you send the geo-loc, and you can send it knowing the other end can accept it, and you can send it based on user input.

Alternative mechanisms: send a re-INVITE or UPDATE with geo-loc info, or SUBSCRIBE/NOTIFY.

[Editor’s note: There is general agreement there is no need for this to be done in INFO]

7. Other Known Use-Cases

These use-cases are known to exist or have been proposed.

7.1. Sending RTP/RTCP statistics during call

There is an implementation of this, and the rationale is the signaling plane box that wants this info is not actually the media plane box that gets RTCP.

[Editor’s note: There is general agreement this should be done with Sub/Not, so it can get stats after the call is over, and since it will probably want periodic reports the overhead of the Subscribe should be dwarfed by the number of Notifies]

7.2. Sending access-location information after call establishment

There is a P-Access-Network-Info header, and some have proposed to send an update for it as a phone roams access points or cells.

[Editor’s note: I think this is an odd thing to do inside an Invite session, vs. in a Sub/Not or Register (and besides most of the time the "network" inserts this header, not the UA).]
7.3. Sending media-control indications

This sends play/pause/resume commands in INFO. This is done today by at least one vendor. The argument is it’s like SDP re-Invite for hold, except at a media content layer above RTP and even RTCP, so not done in RTCP nor SDP.
[Editor’s note: There is general agreement this should be done in the media-plane.]
[J. Rosenberg agrees and notes: "There is a requirement for low latency here and there will be a lot of these that get sent."]

7.4. Sending video fast update command

This is an informational draft [fast-update], which documents what has been implemented, but states it should really be done in the media plane in the future.
[Editor’s note: There is general agreement the draft is correct in stating it should be done in the media-plane and not INFO]

7.5. Sending peripheral control commands

There is actually a patent on this. Someone thinks it makes sense to create a SIP session to your laptop, or vice-versa, and then send USB commands inside MIME in INFO messages.
[Editor’s note: There is general agreement this should be media-plane, if anything.]

7.6. Sending charging information for a call

There was a proposal to use this for Advice of Charge information in TISPAN.
[Editor’s note: IMO it should be a SUBSCRIBE/NOTIFY model, as they want this to survive the Invite session.]

7.7. Sending a screen-pop-up message

There is a patent for doing screen pop-ups using INFO. One could use MESSAGE, but I believe the patent is for generating screen pop-ups like warnings and such, not simple user instant messages.
[P. Kyzivat notes this should be MESSAGE]

7.8. Sending Bookmark Tags

There is a proposal in TISPAN and ATIS to indicate bookmark spots for video streams using INFO.
[Editor’s note: this seems like a media-plane thing, but the reason they’re using SIP is due to media layer processing issues, I think,
whereby the media is processed by a different box than media control?

7.9. Sending Hook-Flash Indication

There is a draft for doing this in [hook-flash], which is deployed. This is along the lines of needing signaling-plane flash indication a la DTMF. RFC 2833 included it for media-plane DTMF events, but [RFC4733] deprecated it.

[Editor’s note: although hook-flash is sometimes considered similar to DTMF, from a user perspective it’s really just a overloaded "button" for hold, transfer, or conference invocation and thus should be handled by the UA natively]

7.10. Session Refresh and Liveness Check

This is actually implemented by several vendors: use INFO to check the session is still alive (and keep it so). This is clearly what [RFC4028] was written for, which uses re-INVITEs and UPDATEs.

7.11. Message-Waiting Indication (MWI)

The idea is to set/clear the MWI light/icon on phones, but to do so it sends INFO outside of dialogs. This is implemented by several vendors. [Editor’s note: This really should be done with SUBSCRIBE/NOTIFY.]

7.12. Media Server Markup Language (MSML)

This is a deployed mechanism "used to control and invoke many different types of services on IP Media Servers", documented in [MSML].

8. Security Considerations

There are no security considerations, since this is merely an informative use-case document.
9. IANA Considerations

There are no IANA considerations associated with this specification.

10. Informative References


Author’s Address

Hadriel Kaplan
Acme Packet
71 Third Ave.
Burlington, MA 01803, USA

Email: hkaplan@acmepacket.com
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat 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 implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST 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 provided by the IETF Administrative Support Activity (IASA).