SIPCORE
Internet-Draft
Obsoletes: RFC 2976
(if approved)
Expires: April 2, 2010

E. Burger
NeuStar, Inc.
H. Kaplan
Acme Packet
C. Holmberg
Ericsson

September 29, 2009

Session Initiation Protocol (SIP) INFO Method and Package Framework
draft-ietf-sipcore-info-events-01

Status of this Memo

Copyright Notice

Abstract

This document provides new semantics for the SIP INFO method of RFC 2976. These new semantics defined here are fully backwards compatible with the old semantics. Core to the new semantics is a mechanism for defining, indicating support of, and exchanging Info Packages that use the INFO method. Applications that need to exchange application information within a SIP invite usage dialog (RFC 5057), can use these Info Packages. This document replaces RFC 2976 but still allows existing legacy INFO usages as defined in RFC 2976.

Conventions Used in this Document

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 [RFC2119]. The terminology in this document conforms to the Internet Security Glossary [RFC4949].

Table of Contents

1. Introduction

[RFC3261] defines a mechanism to setup and tear down SIP sessions. A SIP User Agent (UA) can use the re-INVITE and UPDATE methods during a session to change characteristics of the session, including media properties, target information or properties related to the SIP session timer mechanism [RFC4028].

The purpose of the INFO message [RFC2976] is to carry application level information between endpoints, using the SIP session signaling path. Note that the INFO method is not used to update characteristics of the SIP session, but to allow the applications which use the SIP session to exchange information.

While the INFO method has been widely adopted for specific application use cases, such as ISUP and DTMF exchange, [RFC2976] does not define a mechanisms for SIP UAs to indicate what usages of INFO they support. In addition, [RFC2976] does not provide a mechanism to explicitly indicate the type of application for which the INFO message is used. In some cases it can be determined by the INFO message body content, but not in a general way.

Example: If the Content-Type is "image/jpeg", the MIME-attached content is a JPEG image. Still, there are many useful ways a UA can render an image. The image could be a caller-id picture, a contact icon, a photo for sharing, and so on. The sender does not know which image to send to the receiver if the receiver supports an image content type. Likewise, the receiver does not know the context of an image the client is sending if the receiver supports receiving more than one image content type.

Due to the problems described above, the usage of INFO often requires static configuration about what INFO usages the UAs support, and the way the handle application information transported in INFO messages. That has caused a big risk interoperability problems in the industry, due to undefined content syntax, semantics and UA support of the INFO messages. Therefore, there is a need for a well defined and documented description of what the information sent in the INFO is used for. This situation is identical to the context issue in Internet Mail [RFC3458].

This document defines a mechanism, using Info Packages, which provides the possibility for UAs to indicate what INFO usages they support, and to define content syntax and semantics for the data transported in the INFO messages. The mechanism allows existing legacy INFO usages as defined in RFC 2976. New INFO usages MUST use the mechanism defined in this document.

Event Packages [RFC3265] perform the role of disambiguating the context of a message for subscription-based events. The Info Package mechanisms provides similar functionality for application information exchange using invite dialog usages [RFC5057].

Note that while Info Packages may be similar to subscription-based events, there is no formal relationship between this mechanism and the subscription mechanism.

The Info Package mechanism does not create a separate dialog usage. INFO messages are always part of, and share the fate of, an invite dialog usage. INFO message can not be sent as part of other dialog usages, and they can not be sent outside an existing session.

If a UA supports the Info Package mechanism it indicates, using the Recv-Info header field which Info Packages it is willing to receive, on a per-session basis. A UA can indicate a new set of Info Packages at any time during the lifetime of the invite dialog usage of the session. A UA can use a "nil" value to indicate that it is not willing to receive any Info Packages at a certain moment, but that the UA still supports the Info Package mechanism.

When a UA sends an INFO request, it uses the Info-Package header field to indicate which Info Package is associated with the request.

Section 3 describes the mechanism to indicate support of Info Packages.

Section 4 describes the usage of INFO messages.

Section 6 describes legacy usage of INFO, as defined in [RFC2976].

Section 7 describes guidelines on how to define Info Packages. This document does not define any specific Info Packages.

Annex A provides guidelines and issues to consider when deciding if usage of Info Packages is the most appropriate mechanism for a specific use-case.

2. Applicability

This document extends [RFC2976] to include a mechanism to in SIP messages explicitly indicate the supported Info Packages, and to explicitly indicate what Info Package is associated with an INFO request. The mechanism is backward compatible with legacy usage of INFO, as defined in [RFC2976], and allows such usage. New INFO usages MUST use the mechanism defined in this document.

3. Info Package Support

3.1. General

This section describes how SIP UAs indicate which Info Packages they are willing to receive.

3.2. User Agent Behavior

A UA which supports the Info Package mechanism MUST indicate the set of Info Packages it is willing to receive, using the Revc-Info header field. A UA can list multiple Info Packages in a single Recv-Info header field, and the UA can use multiple Recv-Info header fields.

The indication of Info Packages can take place during the session establishment, and during a target refresh. This includes INVITE, UPDATE, PRACK, ACK, and their non-failure responses (101-199 and 2xx only). Note that the UAC is not required to indicate its set of Info Packages in the initial INVITE request.

Once a UA has indicated that it is willing to to receive a specific Info Package, and a dialog has been established, the UA MUST be prepared to receive INFO request associated with that Info Package.

A UA MUST NOT send INFO request associated with Info Packages until it has received an indication of which Info Packages the remote UA is willing to receive.

If a UA indicates that it is willing to receive of multiple Info Packages, which provide similar functionality, it is not possible to indicate that the UA wishes to receive only one of them. It is up to the application logic associated with the Info Packages, and specific Info Package descriptions to describe application behavior in such cases.

If a UA is not willing to receive any Info Packages, during session establishment or later during the session, the UA MUST indicate this by including a Recv-Info header field with a header value of 'nil'. This enables other UAs to detect that the UA still supports the Info Package mechanism.

Example: If a UA has previously indicated support of Info Packages foo and bar, and the UA during the session wants to indicate that it does not want to receive any Info Packages anymore, the UA sends a target refresh request with a Recv-Info header field with a header value of 'nil'.

For backward compatibility purpose, even if a UA indicates support

the Info Package mechanism, it is still allowed to enable legacy usages of INFO.

This document does not define a SIP option tag [RFC3261] for the Info Package mechanism. However, Info Package specifications MAY define option-tags associated with the specific Info Package, as described in Section 7.5.

Note that, for backward compatibility purpose, if a UA indicates support of the INFO method, it does not implicitly indicate support of the Info Package mechanism. A UA MUST use the Recv-Info header field to indicate support of the Info Package mechanism. Likewise, even if a UA uses the Recv-Info header field to indicate that it support the Info Package mechanism, in addition the UA MUST still also explicitly indicate support of the INFO method.

3.3. Package Versioning

The Info Package mechanism does not support package versioning. Specific Info Package payloads MAY contain version information, which is handled by the applications associated wit the Info Package, but that is outside the scope of the Info Package framework.

Note: Even if an Info Package name contains version numbering (e.g. foo_v2), the Info Package mechanism does not distinguish a version number from the rest of the Info Package name.

3.4. REGISTER Processing

When a UA registers, it SHALL include Recv-Info header field in the REGISTER request, and list the Info Packages that it supports. The registrar MAY later use the information e.g. for forking decisions.

3.5. OPTIONS Processing

If a UA sends an OPTIONS request, or a response, the UA SHALL include Recv-Info header field in the message, and list the Info Packages that it supports.

A UA MUST NOT send INFO requests with Info Packages based on the information the UA received in an OPTIONS request. The Info Packages MUST be negotiated for each session.

3.6. Example

The UAC sends an INVITE request, where the UAC indicates that it is willing to receive Info Packages P and R.

```
   INVITE sip:bob@example.com SIP/2.0
   Via: SIP/2.0/TCP pc33.example.com;branch=z9hG4bK776
   Max-Forwards: 70
   To: Bob <sip:bob@example.com>
   From: Alice <sip:alice@example.com>;tag=1928301774
   Call-ID: a84b4c76e66710@pc33.example.com
   CSeq: 314159 INVITE
   Recv-Info: P, R
   Contact: <sip:alice@pc33.example.com>
   Content-Type: application/sdp
   Content-Length: ...

   ...

   The UAS sends a 200 OK response back to the UAC, where the UAS
   indicates that it is willing to receive Info Packages R and T.
  SIP/2.0 200 OK
  Via: SIP/2.0/TCP pc33.example.com;branch=z9hG4bK776;received=192.0.2.1
  To: Bob <sip:bob@example.com>;tag=a6c85cf
  From: Alice <sip:alice@example.com>;tag=1928301774
  Call-ID: a84b4c76e66710@pc33.example.com
  CSeq: 314159 INVITE
  Contact: <sip:alice@pc33.example.com>
  Recv-Info: R, T
  Content-Type: application/sdp
  Content-Length: ...

  ...

   Since the UAS does not support Info Package P, the UAC decides to
   indicate in the ACK request that it is only willing to receive Info
   Package R, which the UAS also indicated support of.

   ACK sip:ngw1@a.example.com SIP/2.0
   Via: SIP/2.0/TCP pc33.example.com;branch=z9hG4bK776
   To: Bob <sip:bob@example.com>;tag=a6c85cf
   From: Alice <sip:alice@example.com>;tag=1928301774
   Call-ID: a84b4c76e66710@pc33.example.com
   CSeq: 314163 ACK
   Recv-Info: R
   Content-Length: 0
```

4. The INFO Method

4.1. General

This section describes how the UA handling of INFO requests and responses, and message bodies carried in INFO messages. It also describes how an UA can indicate support of Info Packages in OPTIONS requests and during registration.

The INFO method provides additional, application level information that can further enhance a SIP application. Annex A gives more details on the types of application for which the usage of INFO is seen as appropriate.

The rules and procedures in this Section apply to implementations and applications which support this. Existing implementations of, and applications using, [RFC2976], may not follow the rules in this Section. Because of backward compatibility purpose such cases MUST NOT be regarded as error behavior, or wrong protocol usage, but simply part of legacy INFO usage.

4.2. INFO Request

A UA MUST include a Info-Package header field, which indicates the Info Package contained in the request, when it sends an INFO request carrying an Info Package. An INFO request can contain only a single Info Package. A UA MUST NOT send INFO requests associated with Info Packages for which the remote entity has not indicated willingness (using the Recv-Info header filed) to receive for the session.

A UA MAY send an INFO in a legacy usage context. In such case there is no Info Package associated with the usage, and the INFO request does not contain an Info-Package header field. In addition, the support of the legacy usage has not been negotiated using the Recv-Info header field. See Appendix B for examples of legacy usages.

The INFO method MUST NOT be used outside an INVITE dialog usage. The INFO method has no lifetime or usage of its own. Supported Info Packages are negotiated on a per session basis, and the negotiation result MUST NOT be used for other sessions. If a UA receives an INFO request outside an existing dialog, the UA MUST response with a 481 Call Does Not Exist error response.

Due to the possibility of forking, a UAC which during the early dialog phase indicates support of one or more Info Packages (using the Recv-Info header field) MUST be prepared to receive INFO requests from multiple remote entities. Note that different remote entities can indicate different sets of Info Packages which they are willing to receive.

The construction of the INFO request is the same as any other request within an existing INVITE dialog usage. A UA can send INFO requests both on early and confirmed dialogs.

The INFO request MUST NOT contain a Recv-Info header field. The UA can only indicate the Info Packages that it is willing to receive using the messages listed in Section 3.

4.3. INFO Request Message Body

The purpose of the INFO request is to carry application level information between SIP UAs. The application data associated with an Info Package SHOULD be carried as a payload in the message body of the INFO request, unless the information can be retrieved from a SIP header field.

Info Package specifications MUST describe the application level information associated with the Info Package. Message body payloads MUST have a MIME type value defined.

If a UA indicates that it is willing to receive a specific Info Package, it means that the UA also supports any associated message body MIME type associated with the Info Package. However, the UA MUST still indicate support of those MIME types also in the Accept header filed, according to the procedures in [RFC3261].

Some SIP extensions, which are orthogonal to INFO, MAY insert body parts unrelated to the Info Package. UAs MUST conform to [RFC3261] as updated by body-handling [I-D.ietf-sip-body-handling] to support multipart MIME handling.

Each message body (or body part in the case of multipart MIME) MUST contain a Content-Disposition header with an 'Info-Package' header value, in order to in an easy way distinguish payloads associated with the Info Package from other payloads.

If the whole message body is associated with the Info Package, the UA MUST insert a Content-Disposition header with an 'Info-Package' header value in the SIP part of the message. In that case, if multipart MIME is used, the UA does not need to insert an 'Info-Package' header value for the individual body parts.

NOTE: To avoid corner cases with legacy INFO usage, the Info-Package header field is used to indicate the Info Package name, rather than to use a Content-Disposition header field parameter in order to indicate the name.

4.4. INFO Response

If a UA receives an INFO request, associated with an Info-Package that the UA has indicated willingness to receive, and the INFO request contains data associated with that Info-Package, the UA MUST send a 200 OK response.

If a UA receives an INFO request, associated with an Info Package that the UA has not indicated willingness to receive, the UA MUST send a 469 Bad INFO Package response. In the terminology of Multiple Dialog Usages [RFC5057], this represents a Transaction Only failure.

If a UA receives an INFO request for legacy usage, for which no Info-Package is associated (the INFO request does not contain an Info-Package header filed), the UA must send a 200 OK response.

If a UA receives an INFO request, which does not match any existing INVITE dialog usage, the UA MUST send a 481 Call Leg/Transaction Does Not Exist response.

If a UA receives an INFO request, which carries a message body that the UA does not support, and support of the message body is required in the Content-Disposition header field, the UA MUST send a 415 Unsupported Media Type response. If support of the message body is optional, the UA MUST send a 200 OK response even if the UA does not support the message body.

The UAS MAY send other responses, such as Request Failure (4xx), Server Failure (5xx) and Global Failure (6xx) as appropriate for the request.

4.5. INFO Response Message Body

The response to the INFO request is normally generated by the SIP stack before the Info Package application data has been provided to the application associated with the Info Package. Therefore, an Info Package MUST NOT define the inclusion of a message body in an INFO response.

If the application that received the information needs to send some information in the other direction, it MUST trigger a new INFO request, rather than using the response of the received INFO request.

4.6. Order of Delivery

The Info Package framework relies on the CSeq header field to detect if an INFO request is received out of order.

If specific applications need additional mechanisms for order of delivery, those mechanisms, and related procedures, MUST be specified as part of the associated Info Package, and possible sequence numbers etc MUST be defined as application data.

5. Formal INFO Method Definition

5.1. INFO Method

This document describes one new SIP method: INFO. This document replaces the definition and registrations found in [RFC2976].

This table expands on Tables 2 and 3 in [RFC3261].

Header	Where	INFO
Accept	R	o
Accept-Encoding	R	o
Accept-Encoding	2xx	o
Accept-Encoding	415	c
Accept-Language	R	o
Accept-Language	2xx	o
Accept-Language	415	c
Alert-Info		-
Allow	R	o
Allow	200	-
Allow	405	o
Authentication-Info	2xx	o
Authorization	R	o
Call-ID	c	m
Call-Info		o
Contact		-
Content-Disposition		o
Content-Encoding		o
Content-Language		o
Content-Length		o
Content-Type		*
CSeq	c	m
Date		o
Error-Info	3xx-6xx	o
Expires		-
From	c	m
Geolocation	R	o
Max-Breadth	R	-
Max-Forwards	R	o
MIME-Version		o
Min-Expires		-

Organization		o	
Priority	R	-	
Privacy	R	o	
Proxy-Authenticate	407	o	
Proxy-Authorization	R	o	
Proxy-Require	R	o	
Reason	r	o	
Record-Route	R	o	
Record-Route	2xx,18x	o	
Require		o	
Retry-After	R	-	
Retry-After	404,480,486	o	
Retry-After	503	o	
Retry-After	600,603	o	
Route	R	o	
Security-Client	R	o	
Security-Server	421,494	o	
Security-Verify	R	o	
Server	r	o	
Subject	R	o	
Supported	R	o	
Supported	2xx	o	
Timestamp		o	
To	c	m	(w/ Tag)
Unsupported	420	o	
User-Agent		o	
Via		m	
Warning	r	o	
WWW-Authenticate	401	m	
WWW-Authenticate	407	o	

Figure 1: Table 1: Summary of Header Fields

5.2. INFO Header Fields

This table expands on tables 2 and 3 in [RFC3261].

Header field	where	ACK	BYE	CAN	INV	OPT	REG	PRA	INF	MSG	UPD	SUB	NOT	RFR
Info-Package	R	-	-	-	-	-	-	-	m*	-	-	-	-	-
Recv-Info	R	o	-	-	o	o	o	o	-	-	o	-	o	o
Recv-Info	2xx	o	-	-	o	o	-	o	-	-	o	-	o	-
Recv-Info	1xx	o	-	-	o	o	-	o	-	-	o	-	-	-
Recv-Info	r	o	-	-	-	o	-	o	-	-	o	-	-	-

* The Info-Package header field is MANDATORY for INFO requests associated with Info Packages. The Info-Package header field is not applicable for legacy usage INFO requests [RFC2976].

Table 2: INFO-related Header Fields

5.2.1. Info-Package header field

This document adds Info-Package to the definition of the element "message-header" in the SIP message grammar. Section 4 describes the Info-Package header field usage.

For the purposes of matching Info Package types indicated in Recv-Info with those in the Info-Package header field value, one compares the Info-package-name portion of the Info-package-type portion of the Info-Package header field octet-by-octet with that of the Recv-Info header field value. That is, the Info Package name is case sensitive. Info-package-param is not part of the comparison-checking algorithm.

This document does not define values for Info-Package types. Individual Info Package specifications define these values. Such specifications MUST register the values with IANA. These values are Specification Required [RFC5226].

5.2.2. Recv-Info header field

This document adds Recv-Info to the definition of the element "general-header" in the SIP [RFC3261] message grammar. Section 3 describes the Recv-Info header field usage.

6. Legacy INFO Usage

A number of applications, standardized and proprietary, make use of INFO messages as defined in [RFC2976], without defined Info Packages the and a possibility to use SIP to indicate what INFO usages UAs are willing to use. For backward compatibility purpose, this document does not deprecate such usage, and does not mandate to define Info Packages for existing usages. However, any new usage of INFO SHALL use the Info Package mechanism defined in this specification.

Since legacy INFO usages to not have associated Info Packages, it is not possible to use the Recv-Info and Info-Package header fields for legacy INFO usages. That is, a UA can not use the Recv-Info header filed to indicate for which legacy usages it is willing to receive INFO requests, and a UA can not use the Info-Package header to indicate for which legacy INFO usage an INFO request is associated with.

NOTE: For legacy INFO usages, static configuration is often used to define what specific legacy INFO usages UAs support.

An INFO request associated with an Info Package MUST contain a Info-Package header field. An INFO request without an Info-Package header field MUST NOT contain an Info-Package header field, and the request SHALL be interpreted as being a legacy INFO usage request.

UAs are allowed to enable both legacy INFO usages and Info Package usages as part of the same session.

7. Info Package Requirements

7.1. General

This Section provides guidance on how to define an Info Package, and what information needs to be provided.

If an Info Package extends or modifies the behavior described in this document, it MUST be described in the definition for that Info Package. Info Package definitions SHOULD NOT repeat procedures defined in this specification, unless needed for clarification or emphasis purpose.

Info Packages MUST NOT weaken any behavior designated with "SHOULD" or "MUST" in this specification. However, Info Packages MAY strengthen "SHOULD", "MAY", or "RECOMMENDED" requirements to "MUST" strength if applications associated with the Info Package requires it.

Info Package definitions SHALL address the issues defined in the following subsections, unless an issue is not applicable for the specific Info Package.

7.2. Applicability

The Info Package specification MUST describe why the Info Package mechanism, rather than some other mechanism, has been chosen for the specific use-case to transfer application information between SIP endpoints. Common reasons can be a requirement for SIP Proxies or back-to-back User Agents (B2BUAs) to see the transport application information, or that it is seen unfeasible to establish separate dialogs (subscription) for transporting the information.

Annex A provides more information, and describes alternative mechanisms which one should consider for solving the specific use-case.

7.3. Info Package Name

The Info Package specification MUST define an Info Package name.

The specification MUST also define the header field value to be used to indicate support of this package in the Recv-Info and Info-Package header fields. The header field value MUST conform to the ABNF defined in Section 8.2.

The specification MUST also include the information that appears in the IANA registration of the token. For information on registering such types, see Section **9.

7.4. Info Package Parameters

The Info Package specification MAY define Info Package parameters which can be used in the Recv-Info or Info-Package header fields, together with the header field value representing the Info Package.

The specification MUST describe the syntax and semantics of the parameters. It MUST be specified whether a specific parameter is only applicable to the Recv-Info header, the Info-Package header, or both.

Note that Info Package parameters are only applicable for the Info Package(s) for which they have been explicitly defined. If used for other Info Packages they MUST be discarded.

7.5. SIP Option Tags

The Info Package specification MAY define SIP option tags, which can be used as described in [RFC3261].

SIP option tags MUST conform to the SIP Change Process [RFC3427].

7.6. INFO Message Bodies

The Info Package specification MUST define what type of message bodies, if any, are associated with the Info Package, and MUST refer to specifications where the syntax, semantics and MIME type of the message body is described.

7.7. Info Package Usage Restrictions

The Info Package specification MUST define whether a UA is allowed to send overlapping (outstanding) INFO requests associated with the Info Package, or whether the UA has to wait for the response for a previous INFO request associated with the same Info Package.

The specification MUST define whether there SIP level restrictions in the usage of the Info Package. For example, an Info Package may require support of other SIP extensions (e.g. reliable provisional responses).

The specification MUST define whether there are restrictions on indicating support of, or using, the Info Package together with other Info Packages.

If Info Package restrictions are violated (i.e. if overlapping INFO requests are not allowed for an Info Package, but a UA still receives overlapping requests), the UA MUST NOT reject such requests. Instead the application logic associated with the Info Package MUST handle such situations.

7.8. Rate of INFO Requests

The Info Package specification MUST a maximum rate at which INFO requests associated with the specific Info Package can be generated by a UA in a dialog.

The specification MAY define Info Package parameters to be used for indicating or negotiating the INFO request rate. Alternatively the rate information can be included in the application information associated with the Info Package.

7.9. IANA Registrations

The Info Package specification MUST contain an IANA Considerations section that includes definitions for the Info Package Name and, if needed, supported MIME types.

7.10. Security Considerations

If the application information associated with the Info Package requires certain level of security, the Info Package specification MUST describe the mechanisms to be used in order to provide the required security.

Otherwise, even if no additional security than what is provided for the underlying SIP protocol is needed, it SHALL be stated in the Info Package specification.

NOTE: In some cases, it may not be sufficient to mandate TLS in order to secure the Info Package payload, since intermediaries will have access to the payload and past the first hop, there is no way to assure subsequent hops will not forwards the payload in clear text. The best way to ensure secure transport at the application level is

to have the security at the application level. The most common method of achieving this is to use end-to-end security techniques such as S/MIME [RFC3851].

7.11. Application Procedures

The Info Package specification SHOULD contain a description of the application procedures associated with the Info Package, or alternatively refer to application procedures defined elsewhere.

7.12. Examples

It is RECOMMENDED that Info Package specifications include demonstrative message flow diagrams, paired with complete messages and message descriptions.

Note that example flows are by definition informative, and MUST NOT replace normative text

8. Syntax

8.1. General

This Section describes the syntax extensions required for the INFO method. The previous sections describe the semantics. Note the formal syntax definitions described in this document use the ABNF format used in [RFC3261] and contain references to elements defined therein.

8.2. ABNF

```
INFOm               = %x49.4E.46.4F ; INFO in caps
extension-method    = INFOm / token

Info-Package        =  "Info-Package" HCOLON Info-package-type
Recv-Info           =  "Recv-Info" HCOLON Info-package-list
Info-package-list   =  "nil"
                    / Info-package-type *( COMMA Info-package-type )
Info-package-type   =  Info-package-name *( ";" Info-package-param)
Info-package-name   =  token
Info-package-param  =  generic-param
```

NOTE on the Recv-Info production: if the header field value is "nil", the header field MUST NOT contain any other Info Packages, and the SIP message MUST NOT contain more than one Recv-Info header field.

9. IANA Considerations

9.1. Update to Registration of SIP INFO Method

Please update the existing registration in the SIP Methods and Response Codes registry under the SIP Parameters registry that states:

Method: INFO
Reference: [RFC2976]

to:

Method: INFO
Reference: [RFCXXXX]

9.2. Registration of the Info-Package Header Field

Please add the following new SIP header field in the Header Fields subregistry under the SIP Parameters registry.

Header Name: Info-Package
Compact Form: (none)
Reference: [RFCXXXX]

9.3. Registration of the Recv-Info Header Field

Please add the following new SIP header field in the Header Fields subregistry under the SIP Parameters registry.

Header Name: Recv-Info
Compact Form: (none)
Reference: [RFCXXXX]

9.4. Creation of the Info Packages Registry

Please create a subregistry in the SIP Parameters registry for Info Packages. This subregistry has a modified First Come First Served [RFC5226] policy.

The following data elements populate the Info Package Registry.

- Info Package Name: The Info Package Name is a case-sensitive token. In addition, IANA shall not register multiple Info Package names that have identical case-insensitive values.
- Info Package Parameters: The Info Package Parameters are case-sensitive tokens. Info Package Parameters are only applicable to the Info Package for which they are defined, so the same Info Package Parameter Names may exist for different Info Packages.

- Info Package Payload MIME Types: A list of zero or more registered MIME types from the MIME Type Registry.
- Standards Status: Values are "Standards Track" or empty. See below for a discussion and rules on this field.
- Reference: If there is a published specification describing the Info Package, place a reference to that specification in this column. See below for a discussion on this field.

If there is a published specification, the registration MUST include a reference to such specification. The Standards Status field is an indicator of the level of community review for the Info Package specification. If the specification meets the requirements for Specification Required [RFC5226], the value for the Standards Status field is "Standards Track". Otherwise, the field is empty.

This document uses the Info Package Name "nil" to represent "no Info Package present" and as such, IANA shall not honor a request to register the "nil" Info Package.

The initial population of this table shall be:

Name	MIME Type	Standards Status	Reference
nil		Standards Track	[RFCXXXX]

9.5. Registration of the Info-Package Content-Disposition

Please add the following registration to the Content-Disposition registry. The description suitable for the IANA registry is as follows.

The payload of the message carrying this Content-Disposition header field value is the payload of an Info Package.

9.6. SIP Response Code 469 Registration

Please register the 469 response code in the Session Initiation Protocol Parameters - Response Codes registry as follows.
Response Code: 469
Default Reason Phrase: Bad INFO Package
Reference: RFCXXXX

10. Examples

10.1. Simple Info Package

Here Alice sends Bob a simple Info Package payload.

```
INFO sip:alice@192.0.2.1 SIP/2.0
Via: SIP/2.0/UDP 192.0.2.2:5060;branch=z9hG4bKnabcdef
To: Alice <sip:alice@example.net>;tag=1234567
From: Bob <sip:bob@example.com>;tag=abcdefg
Call-Id: 123456mcmxcix
CSeq: 2 INFO
Info-Package: foo
Content-type: application/foo
Content-Disposition: Info-Package
Content-length: 24

I am a foo message type
```

10.2. Multipart INFO Example

Other SIP extensions can put payloads into an INFO method, independent of the Info Package. In this case, the Info Package payload gets put into a Multipart MIME body, with the content disposition indicating which body belongs to the Info Package. Since there is one and only one Info Package payload in the message, we only need to tag which body part goes with the Info Package.

```
INFO sip:alice@192.0.2.1 SIP/2.0
Via: SIP/2.0/UDP 192.0.2.2:5060;branch=z9hG4bKnabcdef
To: Alice <sip:alice@example.net>;tag=1234567
From: Bob <sip:bob@example.com>;tag=abcdefg
Call-Id: 123456mcmxcix
CSeq: 7 INFO
Info-Package: foo
mumble-extension: <cid:abcd9999qq>
Content-Type: multipart/mixed;boundary="theboundary"
Content-Length: ...

--theboundary
Content-Type: application/mumble
Content-Id: abcd9999qq
...

<mumble stuff>

--theboundary
Content-Type: application/foo
Content-Disposition: Info-Package
Content-length: 24

I am a foo message type
--theboundary--
```

11. Modifications to SIP Change Process

By eliminating multiple uses of INFO messages without adequate community review and by eliminating the possibility for rogue SIP UAs from confusing another User Agent by purposely sending unrelated INFO requests, we expect this document's clarification of the use of INFO to improve the security of the Internet. Whilst rogue UAs can still send unrelated INFO requests, this framework provides mechanisms for which the UAS and other security devices can filter for approved Info Packages.

If the content of the Info Package payload is private, User Agents will need to use end-to-end encryption, such as S/MIME, to prevent access to the content. This is particularly important as transport of INFO is likely not to be end-to-end, but through SIP proxies and back-to-back user agents (B2BUA's), which the user may not trust.

The INFO mechanism transports application level information. One implication of this is INFO messages may require a higher level of protection than the underlying SIP-based session signaling. In particular, if one does not protect the SIP signaling from

eavesdropping or authentication and repudiation attacks, for example by using TLS transport, then the INFO request and its contents will be vulnerable, as well. Even with SIP/TLS, any SIP hop along the path from UAC to UAS can view, modify, or intercept INFO requests, as they can with any SIP request. This means some applications may require end-to-end encryption of the INFO payload, beyond, for example, hop-by-hop protection of the SIP signaling itself. Since the application dictates the level of security required, individual Info Packages have to enumerate these requirements. In any event, the Info Package mechanism described by this document provides the tools for such secure, end-to-end transport of application data.

One interesting property of Info Package use is one can reuse the same digest-challenge mechanism used for INVITE based authentication for the INFO request. For example, one could use a quality-of-protection (qop) value of authentication with integrity (auth-int), to challenge the request and its body, and prevent intermediate devices from modifying the body. However this assumes the device which knows the credentials in order to perform the INVITE challenge is still in the path for the INFO, or that the far-end UAS knows such credentials.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC3261] 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.

[I-D.ietf-sip-body-handling] Camarillo, G., "Message Body Handling in the Session Initiation Protocol (SIP)", draft-ietf-sip-body-handling-06 (work in progress), March 2009.

12.2. Informative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2976] Donovan, S., "The SIP INFO Method", RFC 2976, October 2000.

[RFC4497] Elwell, J., Derks, F., Mourot, P., and O. Rousseau, "Interworking between the Session Initiation Protocol (SIP) and QSIG", BCP 117, RFC 4497, May 2006.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

[RFC3080] Rose, M., "The Blocks Extensible Exchange Protocol Core", RFC 3080, March 2001.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

[RFC3427] Mankin, A., Bradner, S., Mahy, R., Willis, D., Ott, J., and B. Rosen, "Change Process for the Session Initiation Protocol (SIP)", BCP 67, RFC 3427, December 2002.

[RFC3841] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Caller Preferences for the Session Initiation Protocol (SIP)", RFC 3841, August 2004.

[RFC3372] Vemuri, A. and J. Peterson, "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures", BCP 63, RFC 3372, September 2002.

[RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[RFC3458] Burger, E., Candell, E., Eliot, C., and G. Klyne, "Message Context for Internet Mail", RFC 3458, January 2003.

[RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[RFC4028] Donovan, S. and J. Rosenberg, "Session Timers in the Session Initiation Protocol (SIP)", RFC 4028, April 2005.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[RFC4240] Burger, E., Van Dyke, J., and A. Spitzer, "Basic Network Media Services with SIP", RFC 4240, December 2005.

[RFC4730] Burger, E. and M. Dolly, "A Session Initiation Protocol (SIP) Event Package for Key Press Stimulus (KPML)", RFC 4730, November 2006.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[RFC5022] Van Dyke, J., Burger, E., and A. Spitzer, "Media Server Control Markup Language (MSCML) and Protocol", RFC 5022, September 2007.

[RFC5057] Sparks, R., "Multiple Dialog Usages in the Session Initiation Protocol", RFC 5057, November 2007.

[RFC5168] Levin, O., Even, R., and P. Hagendorf, "XML Schema for Media Control", RFC 5168, March 2008.

[W3C.REC-voicexml21-20070619]
Porter, B., McGlashan, S., Lee, A., Burnett, D., Carter, J., Oshry, M., Bodell, M., Baggia, P., Rehor, K., Burke, D., Candell, E., and R. Auburn, "Voice Extensible Markup Language (VoiceXML) 2.1", World Wide Web Consortium Recommendation REC-voicexml21-20070619, June 2007, <http://www.w3.org/TR/2007/REC-voicexml21-20070619>.

[I-D.ietf-speechsc-mrcpv2]
Shanmugham, S. and D. Burnett, "Media Resource Control Protocol Version 2 (MRCPv2)",

draft-ietf-speechsc-mrcpv2-19 (work in progress), June 2009.

[I-D.saleem-msml]
Sharratt, G. and A. Saleem, "Media Server Markup Language (MSML)", draft-saleem-msml-08 (work in progress), February 2009.

Appendix A. Info Package Considerations

A.1. General

This section covers considerations to take into account when deciding whether the usage of an Info Package is appropriate for transporting of application information for a specific use-case.

A.2. Appropriateness of Info Package Usage

When designing an Info Package, for application level information exchange, it is important to consider: is signaling, using INFO requests, within a SIP session, an appropriate mechanism for the use-case? Is it because it is the most reasonable and appropriate choice, or merely because "it's easy"? Choosing an inappropriate mechanism for a specific use-case can cause negative effects in SIP networks where the mechanism is used.

A.3. Dialog Fate Sharing

As described in [RFC5057], an INFO request is always part of an INVITE dialog usage.

One needs to consider the fate of the dialog usage of an INFO request is rejected. In some cases it may be acceptable that the whole dialog useage is terminated, while in other cases is is desirable to maintain the dialog usage.

A.4. INFO Request Rate and Volume

There is no default throttling mechanism for INFO requests. Apart from the session establishment, the number of SIP messages exchanged during a normal SIP session is rather small.

Some applications, like sending of DTMF tones, can generate a burst of up to 20 messages per second. Other applications, like constant GPS location updates, could generate a high rate of INFO requests during the whole session.

Furthermore, SIP messages tend to be relatively small, on the order of 500 Bytes to 32K Bytes. SIP is a poor mechanism for direct exchange of bulk data beyond these limits, especially if the headers plus body exceed the UDP MTU [RFC0768]. Appropriate mechanisms for such traffic include HTTP [RFC2616], MSRP [RFC4975], or other user plane data transport mechanisms.

A.5. Alternative Mechanisms

A.5.1. Alternative SIP signaling plane mechanisms

A.5.1.1. General

This subsection describes some alternative mechanisms for transporting application information on the SIP signaling plane, using SIP messages.

A.5.1.2. SUBSCRIBE/NOTIFY

An alternative for application level interaction is SIP Events, also known as SUBSCRIBE/NOTIFY [RFC3265]. In this model, a user agent requests state information, such as key pad presses from a device to an application server or key map images from an application server to a device.

A SUBSCRIBE requests creates a new session, and a subscription dialog usage [RFC5057], which is separate, and does not share the fate any other sessions.

The subscription mechanism can be used by SIP entities to receive state information about SIP sessions, without requiring the entities to be part of the route set of those sessions.

As SUBSCRIBE/NOTIFY messages traverse through stateful SIP proxies and B2BUAs, the resource impact caused by the subscription sessions needs to be considred. The number of subscription sessions per user also needs to be considered.

As for any other SIP signaling plane based mechanism for transporting application information, the SUBSCRIBE/NOTIFY messages can put a significant burden on intermediate SIP entities which are part of the session route set, but do not have any interest in the application information transported between the end users.

A.5.1.3. MESSAGE

The MESSAGE method [RFC3428] defines one-time instant message exchange, typically for sending MIME contents for rendering to the

user.

A.5.2. Media Plane Mechanisms

A.5.2.1. General

In SIP, media plane channels associated with SIP sessions are established using SIP signaling, but the data exchanged on the media plane channel does not traverse SIP signaling intermediates, so if there will be a lot of information exchanged, and there is no need for the SIP signaling intermediates routing to examine the information, it is recommended to use a media plane mechanism, rather than a SIP signaling based.

A low latency requirement for the exchange of information is one strong indicator for using a media channel. Exchanging information through the SIP routing network can introduce hundreds of milliseconds of latency.

A.5.2.2. MRCPv2

One mechanism for media plane exchange of application data is MRCPv2 [I-D.ietf-speechsc-mrcpv2], where a media plane connection-oriented channel, such as a TCP [RFC0793] or SCTP [RFC4960] stream is established.

A.5.2.3. MRSP

MSRP [RFC4975] defines session-based instant messaging as well as bulk file transfer and other such large-volume uses.

A.5.3. Non-SIP related mechanisms

Another alternative is to use a totally externally signaled channel, such as HTTP [RFC2616]. In this model, the user agent knows about a rendezvous point to direct HTTP requests to for the transfer of information. Examples include encoding of a prompt to retrieve in the SIP Request URI in [RFC4240] or the encoding of a SUBMIT target in a VoiceXML [W3C.REC-voicexml21-20070619] script.

Appendix B. Legacy INFO Usages

We do not intend this section to be a comprehensive catalog of INFO usages. However, it should give the reader a flavor for current INFO usages.

B.1. ISUP

SIP-T uses Content-Type to identify ISUP protocol elements in an INFO message. See RFC3372 [RFC3372].

B.2. QSIG

QSIG uses Content-Type to identify QSIG protocol elements in an INFO message. See RFC4497 [RFC4497].

B.3. MSCML

MSCML uses a Require to ensure the UAS understands that INFO messages of the MSCML type are in fact MSCML messages. See RFC5022 [RFC5022].

B.4. MSML

MSML endpoints just know the INFO messages carry MSML and from the Content-Type of the given INFO method request. See the MSML [I-D.saleem-msml] draft.

B.5. Video Fast Update

Microsoft, Polycom, and Radvision used INFO messages as an interim solution for requesting fast video update before the ability to request I-Frames in RTCP was available. See the XML Schema for Media Control [RFC5168] for more information.

Appendix C. Acknowledgements

We are standing on the shoulders of giants. Jonathan Rosenberg did the original "INFO Considered Harmful" Internet Draft on 26 December 2002, which influenced the work group and this document. Likewise, Dean Willis influenced the text from his Internet Draft, "Packaging and Negotiation of INFO Methods for the Session Initiation Protocol" of 15 January 2003. Four paragraphs come from Jonathan Rosenberg's INFO Litmus draft. My, we have been working on this for a long time!

This and other related drafts have elicited well over 450 messages on the SIP list. People who have argued with its thesis, supported its thesis, added to the examples, or argued with the examples, include the following individuals:

Adam Roach, Bram Verburg, Brian Stucker, Chris Boulton, Cullen Jennings, Dale Worley, Dean Willis, Frank Miller, Gonzalo Camarillo, Gordon Beith, Henry Sinnreich, James Jackson, James Rafferty, Jeroen van Bemmel, Joel Halpern, John Elwell, Johnathan Rosenberg, Juha Heinanen, Keith Drage, Kevin Attard Compagno,

Manpreet Singh, Martin Dolly, Mary Barnes, Michael Procter, Paul Kyzivat, Peili Xu, Peter Blatherwick, Raj Jain, Rayees Khan, Robert Sparks, Roland Jesske, Salvatore Loreto, Sam Ganesan, Sanjay Sinha, Spencer Dawkins, Steve Langstaff, Sumit Garg, and Xavier Marjou.

John Elwell and Francois Audet helped with QSIG references. In addition, Francois Audet provided actual text for the revised abstract. Keith Drage gave lots of excellent comments and helped immensely with Figure 1.

The work group version of this document benefited from the close readings and comments from
John Elwell, Paul Kyzivat, Dean Willis, Francois Audet, Dale Worley, Andrew Allen, Adam Roach, Anders Kristensen, Gordon Beith, Ben Campbell, Bob Penfield, Keith Drage, Jeroen van Bemmel, Mary Barnes, and Salvatore Loreto.

Since publication of the first work group version of this document, we have had over 329 messages. New voices in addition to those included above include
Arun Arunachalam, Christian Stredicke, Eric Rescorla, Inaki Baz Castillo, and Roni Evan.

However, any errors and issues we missed are still our own.

Appendix D. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-sip-info-events-03
- Clarified Abstract language
- All SIP dialogs are now refered to as sessions
- Clarified the image example in the Introduction
- Clarified the relationship (none) between SIP Event Packages and SIP Info Packages
- Really, really clarified the protocol is NOT a negotiation but an advertisement
- Split Section 3 into UAS and UAC behavior
- Moved the example in section 3 into its own sub-section, and used full SIP header fields
- Clarified forking behavior
- Clarified language around when to send a body
- Added 469 error response, instead of reusing 489
- Clarified overlapping INFO method handling

- Fixed table 1 to follow 3261, not 2543
- Added REFER to the INFO Headers table
- replaced token-nodot with token for Info-Package header field values
- Clarified end-to-end security considerations
- Info Package parameters are semi-colon delimited, not dot delimited

Changes from -02

- Applicability statement explicitly says we're backwards compatible
- Explicitly state we work like UPDATE (both early and confirmed dialogs)
- Agreed text for IANA Considerations package registry

Changes from -01

- One and only one Info Package per INFO
- Removed Send-Info header field, greatly simplifying negotiation
- Multiple body part identification through Content-Disposition: Info-Package
- Note that forking INVITEs may result in multiple INFOs coming back to INVITE originator
- Describe how a UAS can enforce strict adherence to this document
- Remove CANCEL INFO faux pas
- Better explained overlapping INFO issues and resolutions
- Token names are now really case sensitive
- Moved Info Package Considerations to an Appendix
- Introduced stronger, yet more open, IANA registration process
- Took a few more paragraphs from INFO Litmus to cover all bases.
- Added RFC 5168 to legacy usages

Changes from -00

- Corrected ABNF.
- Enabled sending of legacy INFO messages. Receiving legacy INFO messages was already here.
- Negotiation is not Offer/Answer, it is Offer/Offer.
- Created the explicit "nil" Info Package to indicate no info package.
- Fixed CANCEL impacting future transactions.
- Added Registrar behavior.
- Added OPTIONS processing.
- Clarified overlapping INFO method processing.
- Described multiple INFO bodies in a single INFO method.
- Took out Info-Package as a header field for responses to the INFO method.
- Expanded on risks of using INFO and filled-in more on the alternatives

- Moved definitions of INFO into the body of the text and cleaned up IANA Considerations section
- Added legacy usages descriptions

Authors' Addresses

Eric W. Burger
NeuStar, Inc.
46000 Center Oak Plaza
Sterling, VA 20166-6579
USA

Email: eburger@standardstrack.com
URI: http://www.standardstrack.com

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

Phone:
Fax:
Email: hkaplan@acmepacket.com
URI:

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas, 02420
Finland

Phone:
Fax:
Email: christer.holmberg@ericsson.com
URI: