SIPCORE
Internet-Draft
Obsoletes: RFC 2976
(if approved)
Intended status: Standards Track
Expires: July 22, 2010

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

January 18, 2010

Session Initiation Protocol (SIP) INFO Method and Package Framework
draft-ietf-sipcore-info-events-05

Abstract

This document defines a method, INFO, for the Session Initiation Protocol (SIP) [RFC3261], and an Info Package mechanism. The document obsoletes [RFC2976]. For backward compatibility the document also specifies a "legacy" mode of usage of the INFO method that is compatible with the usage previously defined in [RFC2976], referred to as "legacy INFO Usage" in this document.

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].

Status of this Memo

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

This Internet-Draft will expire on July 22, 2010.

Copyright Notice

Table of Contents

1. Introduction

This document defines a method, INFO, for the Session Initiation Protocol (SIP) [RFC3261].

The purpose of the INFO message is to carry application level information between endpoints, using the SIP dialog signaling path. Note that the INFO method is not used to update characteristics of a SIP dialog or session, but to allow the applications which use the SIP session to exchange information (which might update the state of those applications).

Use of the INFO method does not constitute a separate dialog usage. INFO messages are always part of, and share the fate of, an invite dialog usage [RFC5057]. INFO messages cannot be sent as part of other dialog usages, or outside an existing dialog.

This document also defines an Info Package mechanism. An Info Package specification defines the content and semantics of the information carried in an INFO message associated with the Info Package. The Info Package mechanism also provides a way for UAs to indicate for which Info Packages they are willing to receive INFO requests, and which Info Package a specific INFO request is associated with.

A UA uses the Recv-Info header field, on a per-dialog basis, to indicate for which Info Packages it is willing to receive INFO requests. A UA can indicate an initial set of Info Packages during dialog establishment and can indicate a new set during the lifetime of the invite dialog usage.

NOTE: A UA can use an empty Recv-Info header field (a header field without a value) to indicate that it is not willing to receive INFO requests for any Info-Package, but to inform other UAs that it 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. One particular INFO request can only be associated with a single Info Package.

2. Applicability

This document defines a method, INFO, for the Session Initiation Protocol (SIP) [RFC3261], and an Info Package mechanism. The document obsoletes [RFC2976]. For backward compatibility the document also specifies a "legacy" mode of usage of the INFO method

that is compatible with the usage previously defined in [RFC2976], referred to as "legacy INFO Usage" in this document.

3. The INFO Method

3.1. General

The INFO method provides a mechanism for transporting application level information that can further enhance a SIP application. Annex A gives more details on the types of applications for which the use of INFO is appropriate.

This section describes how a UA handles INFO requests and responses, as well as the message bodies included in INFO messages.

3.2. INFO Request

3.2.1. INFO Request Sender

An INFO request can be associated with an Info Package (see Section 4), or associated with a legacy INFO usage (see Section 9).

The construction of the INFO request is the same as any other non-target refresh request within an existing invite dialog usage as described in Section 12.2 of [RFC3261].

When a UA sends an INFO request associated with an Info Package, it MUST include an Info-Package header field that indicates which Info Package is associated with the request. A specific INFO request can be used only for a single Info Package.

When a UA sends an INFO request associated with an legacy INFO usage there is no Info Package associated with the request, and the UA MUST NOT include an Info-Package header field in the request.

The INFO request MUST NOT contain a Recv-Info header field. A UA can only indicate a set of Info Packages for which it is willing to receive INFO requests by using the SIP methods (and their responses) listed in Section 4.

A UA MUST NOT send an INFO request outside an invite dialog usage and MUST NOT send an INFO request for an Info Package inside an invite dialog usage if the remote UA has not indicated willingness to receive that Info-Package within that dialog.

If a UA receives a 469 (Bad INFO Package) response to an INFO request, based on [RFC5057] the response represents a Transaction

Only failure, and the UA MUST NOT terminate the invite dialog usage.

Due to the possibility of forking, the UA whichs sends the initial INVITE reqest MUST be prepared to receive INFO requests from multiple remote UAs during the early dialog phase. In addition, the UA MUST be prepared to receive different Recv-Info header field values from different remote UAs.

NOTE: If the UAS (receiver of the initial INVITE request) sends an INFO request just after it has sent the response which creates the dialog, the UAS needs to be prepared that the INFO request can reach the UAC before the dialog creating response, and might therefore be rejected by the UAC. In addition, an INFO request might be rejected due to a race condition, if a UA sends the INFO request at the same time as the remote UA sends a new set of Info Packages for which it is willing to receive INFO requests.

3.2.2. INFO Request Receiver

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 (see Section 11.6), which contains a Recv-Info Header field with Info Packages for which the UA is willing to receive INFO requests. The UA MUST NOT use the response to update the set of Info Packages, but simply to indicate the current set. In the terminology of Multiple Dialog Usages [RFC5057], this represents a Transaction Only failure, and does not terminate the invite dialog usage.

If a UA receives an INFO request associated with an Info Package and the message body part with Content-Disposition 'Info-Package' (see Section 3.3.1) has a MIME type that the UA supports but not in the context of that Info Package, it is RECOMMENDED that the UA send a 415 (Unsupported Media Type) response.

The UA MAY send other error responses, such as Request Failure (4xx), Server Failure (5xx) and Global Failure (6xx), in accordance with the error handling procedures in [RFC3261].

Otherwise, if the INFO request is syntactically correct and well structured, the UA MUST send a 200 (OK) response.

NOTE: If the application needs to reject the information which it received in an INFO request, that needs to be done on the application level. Ie the application needs to trigger a new INFO request, which contains information that the previously received application data was not accepted. Individual Info Package specifications need to describe the details for such procedures.

3.2.3. SIP Proxies

Proxies need no additional behavior beyond that described in [RFC3261] to support INFO.

3.3. INFO Message Body

3.3.1. INFO Request Message Body

The purpose of the INFO request is to carry application level information between SIP UAs. The application information data is carried in the payload of the message body of the INFO request.

NOTE: An INFO request assocated with an Info Package can also include information associated with the Info Package using Info-Package header field parameters.

If an INFO request associated with an Info Package contains a message body part, the body part is identified by a Content-Disposition header field 'Info-Package' value. The body part can contain a single MIME type, or it can be a multipart [RFC5621] which contains other body parts associated with the Info Package.

UAs MUST support multipart body parts in accordance with [RFC5621].

NOTE: An INFO request can also contain other body parts that are meaningful within the context of an invite dialog usage but are not specifically associated with the INFO method and the application concerned.

When a UA supports a specific Info-Package, the UA MUST also support message body MIME types in accordance with that Info-Package. However, in accordance with [RFC3261] the UA still indicates the supported MIME types using the Accept header.

3.3.2. INFO Response Message Body

A UA MUST NOT include a message body associated with an Info Package in an INFO response. Message bodies associated with Info Packages MUST only be sent in INFO requests.

A UA MAY include a message body which is not associated with an Info Package in an INFO response.

3.4. Order of Delivery

The Info Package mechanism does not define a delivery order mechanism. Info Packages can rely 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, are specified as part of the associated Info Package (e.g. the use of sequence numbers within the application data).

4. Info Packages

4.1. General

An Info Package specification defines the content and semantics of the information carried in an INFO message associated with an Info Package. The Info Package mechanism provides a way for UAs to indicate for which Info Packages they are willing to receive INFO requests, and which Info Package a specific INFO request is associated with.

4.2. User Agent Behavior

4.2.1. General

This section describes how a UA handles Info Packages, how a UA uses the Recv-Info header field, and how the UA acts in re-INVITE rollback situations.

4.2.2. UA Procedures

A UA which supports the Info Package mechanism MUST indicate, using the Recv-Info header field, the set of Info Packages for which it is willing to receive INFO requests. A UA can list multiple Info Packages in a single Recv-Info header field, and the UA can use multiple Recv-Info header fields. A UA can use an empty Recv-Info header field, ie a header field without any header field values.

A UA provides its set of Info Packages for which it is willing to receive INFO requests during the dialog establishment. A UA can update the set of Info Packages during the invite dialog usage.

If a UA is not willing to receive INFO requests for any Info Packages, during dialog establishment or later during the invite dialog usage, the UA MUST indicate this by including an empty Recv-Info header field. This informs other UAs that the UA still supports the Info Package mechanism.

Example: If a UA has previously indicated Info Packages 'foo' and 'bar' in a Recv-Info header field, and the UA during the lifetime of

the invite dialog usage wants to indicate that it does not want to receive INFO requests for any Info Packages anymore, the UA sends a message with an empty Recv-Info header field.

Once a UA has sent a message with a Recv-Info header field containing a set of Info Packages, the set is valid until the UA sends a new Recv-Info header field containing a new, or empty, set of Info Packages.

Once a UA has indicated that it is willing to receive INFO requests for a specific Info Package, and a dialog has been established, the UA MUST be prepared to receive INFO request associated with that Info Package until the UA indicates that it is no longer willing to receive INFO requests associated with that Info Package.

For a specific dialog usage, a UA MUST NOT send an INFO request associated with an Info Package until it has received an indication that the remote UA is willing to receive INFO requests for that Info Package, or after the UA has received an indication that the remote UA is no longer willing to receive INFO requests associated with that Info Package.

NOTE: When a UA sends a message which contains a Recv-Info header field with a new set of Info Packages for which the UA is willing to receive INFO requests the remote UA might, before it receives the message, send an INFO request based on the old set of Info Packages. In this case the receiver of the INFO requests rejects, and sends a 469 (Bad INFO Package) response to, the INFO request.

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

For backward compatibility purpose, even if a UA indicates support of the Info Package mechanism, it is still allowed to enable legacy INFO usages Appendix A. In addition, if a UA indicates support of the INFO method using the Allow header field [RFC3261], it does not implicitly indicate support of the Info Package mechanism. A UA MUST use the Recv-Info header field in order to indicate that it supports the Info Package mechanism. Likewise, even if a UA uses the Recv-Info header field to indicate that it supports the Info Package mechanism, in addition the UA still indicates support of the INFO method using the Allow header.

This document does not define a SIP option tag [RFC3261] for the Info

Package mechanism. However, an Info Package specification can define an option-tag associated with the specific Info Package, as described in Section 10.6.

4.2.3. Recv-Info header field rules

The text below defines rules on when a UA is required to include a Recv-Info header field in SIP messages. Section 6.1 lists the SIP methods, for which a UA can insert a Recv-Info header field in requests and responses.

- The sender of an initial INVITE request MUST include a Recv-Info header field in the initial INVITE request, even if the sender is not willing to receive INFO requests asscoiated with any Info Package.

- The receiver of a request which contains a Recv-Info header field MUST include a Recv-Info header field in a reliable 18x/2xx response to the request, even if the request contains an empty Recv-Info header field, and even if the header field value of the receiver has not changed since the previous time it sent a Recv-Info header field.

- A UA MUST NOT include a Recv-Info header field in a response if the associated request did not contain a Recv-Info header field.

NOTE: Different from the rules for generating SDP answers [RFC3264], the receiver of a request which contains a set of Info Packages is not restricted to generate its own set of Info Packages as a subset of the Info Package set received in the Info Package header field of the request.

Similar to SDP answers, the receiver can include the same Recv-Info header field value in multiple responses (18x/2xx) for the same INVITE/re-INVITE transaction, but the receiver MUST NOT include a Recv-Info header field value which is different from a value that the receiver has already included in a response for the same transaction.

4.2.4. Info Package fallback rules

If the receiver of a request which contains a Recv-Info header field rejects the request, both the sender and receiver of the request MUST roll back to the set of Info Packages which was used before the request was sent. This also applies to the case where the receiver of an INVITE/re-INVITE request has included a Recv-Info header field in a provisional response, but later rejects the request.

NOTE: The dialog state rollback rules for Info Packages might differ from the rules for other types of dialog state information (SDP, target, etc).

4.3. REGISTER Processing

This document allows a UA to insert a Recv-Info header field in a REGISTER request. However, a UA SHALL NOT include a header value for a specific Info Package unless the specific Info Package specification describes how the header field value shall be interpreted and used by the registrar, e.g. in order to determine request targets.

Rather than using the Recv-Info header field in order to determine request targets, it is recommended to use more appropriate mechanisms, e.g. based on [RFC3840]. However, this document does not define a feature tag for the Info Package mechanism, or a mechanism to define feature tags for specific Info Packages.

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	415	o
Accept-Encoding	R	o
Accept-Encoding	2xx	o
Accept-Encoding	415	c
Accept-Language	R	o
Accept-Language	2xx	o
Accept-Language	415	o
Accept-Resource-Priority	2xx,417	o
Alert-Info		-
Allow	R	o
Allow	405	m
Allow	r	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	
Geolocation-Error	r	o	
Max-Breadth	R	-	
Max-Forwards	R	o	
MIME-Version		o	
Min-Expires		-	
Organization		-	
Priority	R	-	
Privacy		o	
Proxy-Authenticate	401	o	
Proxy-Authenticate	407	m	
Proxy-Authorization	R	o	
Proxy-Require	R	o	
Reason	R	o	
Record-Route	R	o	
Record-Route	2xx,18x	o	
Referred-By	R	o	
Request-Disposition	R	o	
Require		o	
Resource-Priority		o	
Retry-After	R	-	
Retry-After	404,413,480,486	o	
Retry-After	500,503	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

6. INFO Header Fields

6.1. General

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

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG	PRA	INF	MSG	UPD
Info-Package	R		-	-	-	-	-	-	-	m*	-	-
Recv-Info	R		-	-	-	m	-	o	o	-	-	o
Recv-Info	2xx		-	-	-	o**	-	-	o***	-	-	o***
Recv-Info	1xx		-	-	-	o**	-	-	-	-	-	-
Recv-Info	469		-	-	-	-	-	-	-	m*	-	-
Recv-Info	r		-	-	-	o	-	-	o	-	-	o

Header field	where	SUB	NOT	RFR
Info-Package	R	-	-	-
Recv-Info	R	-	-	-
Recv-Info	2xx	-	-	-
Recv-Info	1xx	-	-	-
Recv-Info	469	-	-	-
Recv-Info	r	-	-	-

Table 2: INFO-related Header Fields

The support and usage of the Info-Package and Recv-Info header fields is not applicalbe to UAs that only support legacy INFO usages.

* Not applicalbe to INFO requests and responses associated with legacy INFO usages.

** Mandatory in at least one reliable 18x/2xx response, if sent, to the INVITE request, if the associated INVITE request contained a Recv-Info header field.

*** Mandatory if the associated request contained a Recv-Info header field.

6.2. Info-Package header field

This document adds Info-Package to the definition of the element "message-header" in the SIP message grammar [RFC3261]. Section 3 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.

6.3. Recv-Info header field

This document adds Recv-Info to the definition of the element "message-header" in the SIP message grammar [RFC3261]. Section 4 describes the Recv-Info header field usage.

7. Info Package Considerations

7.1. 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.

7.2. 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 dialog, 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.

7.3. INFO Request Rate and Volume

There is no default throttling mechanism for INFO requests. Apart from the SIP session establishment, the number of SIP messages exchanged during the lifetime 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 lifetime of the invite dialog usage.

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 media plane data transport mechanisms.

7.4. Alternative Mechanisms

7.4.1. Alternative SIP signaling plane mechanisms

7.4.1.1. General

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

7.4.1.2. SUBSCRIBE/NOTIFY

An alternative for application level interaction is to use subscription-based events [RFC3265], which uses the SIP SUBSCRIBE and NOTIFY methods. Using that mechanism, a UA 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.

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

While an INFO request is always part of, and shares the fate of, an existing invite dialog usage, a SUBSCRIBE request creates a separate dialog usage [RFC5057], and is normally sent outside an existing dialog usage.

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

As SUBSCRIBE/NOTIFY messages traverse through stateful SIP proxies and B2BUAs, the resource impact caused by the subscription dialogs needs to be considered. The number of subscription dialogs 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 dialog route set, but do not have any interest in the application

information transported between the end users.

7.4.1.3. MESSAGE

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

7.4.2. Media Plane Mechanisms

7.4.2.1. General

In SIP, media plane channels associated with SIP dialogs 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 intermediaries 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.

7.4.2.2. 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.

7.4.2.3. MRSP

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

7.4.3. Non-SIP related mechanisms

Another alternative is to use a a SIP-independent mechanism, such as HTTP [RFC2616]. In this model, the UA 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.

8. Syntax

8.1. General

This section describes the syntax extensions to the ABNF syntax defined in [RFC3261] required for the INFO method, and adds definitions for the Info-Package and Recv-Info header fields. The previous sections describe the semantics. The ABNF defined in this specification is conformant to [RFC5234].

8.2. ABNF

```
INFOm               = %x49.4E.46.4F ; INFO in caps
Method              =/ INFOm

message-header      =/ (Info-Package / Recv-Info) CRLF
Info-Package        =  "Info-Package" HCOLON Info-package-type
Recv-Info           =  "Recv-Info" HCOLON [Info-package-list]
Info-package-list   =  Info-package-type *( COMMA Info-package-type )
Info-package-type   =  Info-package-name *( SEMI Info-package-param)
Info-package-name   =  token
Info-package-param  =  generic-param
```

9. Legacy INFO Usage

9.1. General

A number of applications, standardized and proprietary, make use of the INFO method as it was previously defined in [RFC2976], referred to as "legacy INFO usage".

For backward compatibility purpose, this document does not deprecate such usages, and does not mandate users to define Info Packages for such usages. However, any new usage of INFO SHALL use the Info Package mechanism defined in this specification.

9.2. Problems

While legacy INFO usage has been widely adopted for specific application use cases, [RFC2976] did not define a mechanism for SIP UAs to indicate for which types of applications and contexts they support the INFO method. In addition, [RFC2976] did not provide a mechanism to explicitly indicate the type of application and context for which a specific INFO message is associated.

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.

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

Due to the problems described above, legacy INFO usages often require static configuration about for what type of applications and contexts UAs support the INFO method, and the way they handle application information transported in INFO messages. That has caused interoperability problems in the industry. Therefore, a need for a well defined and documented description of what the information sent in the INFO is used for has been identified. This situation is analogous to the context issue in Internet Mail [RFC3458].

9.3. Co-existence with Info Package based INFO usage

As described in Section 3, an INFO request associated with an Info Package always contains an Info-Package header field. A UA MUST NOT insert an Info-Package header field in a legacy INFO request.

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

See Appendix A for examples of existing legacy INFO usages.

10. Info Package Requirements

10.1. General

This section provides guidance on how to define an Info Package, and what information needs to exist in an Info Package specification.

If, for an Info Package, there is a need to extend or modify the behavior described in this document, that behaviour MUST be described in the Info Package specification. It is bad practice for Info Package specifications to repeat procedures defined in this document, unless needed for clarification or emphasis purpose.

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

Info Package specifications MUST address the issues defined in the following subsections, or document why an issue is not applicable for the specific Info Package.

Section 7.4 describes alternative mechanisms, which should be considered as part of the process for solving a specific use-case, when there is a need for transporting application information.

10.2. Overal Description

The Info Package specification MUST contain an overall description of the Info Package: what type of information are carried in INFO requests associated with the Info Package, and for what type of applications and functionalities UAs can use the Info Package.

If the Info Package is defined for a specific application, the Info Package specification MUST state which application UAs can use the Info Package with.

10.3. 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 transported application information (which would not be the case if the information was transported on a media path), or that it is not seen feasible to establish separate dialogs (subscription) in order to transport the information.

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

10.4. Info Package Name

The Info Package specification MUST define an Info Package name, which UAs use as a header field value (e.g. "infoX") to identify the Info 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 Info Package mechanism does not support package versioning.

Specific Info Package message body payloads can contain version information, which is handled by the applications associated with the Info Package. However, such feature is outside the scope of the generic Info Package mechanism.

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.

10.5. 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 which indicates the Info Package name (see Section 10.4.

The Info Package specification MUST define the syntax and semantics of the defined parameters. In addition, the specification MUST define whether a specific parameter is only applicable to the Recv-Info header field, the Info-Package header field, or both.

By default, an Info Package parameter is only applicable for the Info Package for which the parameter has been explicitly defined.

NOTE: Info Package parameters defined for specific Info Packages can share the name with parameters defined for other Info Packages, but the parameter semantics are specific to the Info Package for which they are defined.

10.6. SIP Option Tags

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

The registration requirements for option tags are defined in [I-D.peterson-rai-rfc3427bis].

10.7. INFO Message Body Parts

The Info Package specification MUST define which message body part MIME types are associated with the Info Package. The specification MUST either define those body parts, which include the syntax, semantics and MIME type of the each body part, or refer to other documents which define the body parts.

If multiple message body part MIME types are associated with an Info Package, the Info Package specification MUST define whether UAs need to use multipart body parts in order to include multiple body parts

in a single INFO request.

10.8. Info Package Usage Restrictions

If there are restrictions on how UAs can use an Info Package, the Info Package specification MUST document such restrictions.

There can be restrictions related to whether UAs are 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.

There can also be restrictions related to whether UAs need to support and use other SIP extensions and capabilities when they use the Info Package, and if there are restrictions related to how UAs can use the Info-Package together with other Info Packages.

As the SIP stack might not be aware of Info Package specific restrictions, it cannot be assumed that overlapping requests would be rejected. As defined in Section 3.2.2, UAs will normally send a 200 (OK) response to an INFO request. The application logic associated with the Info Package needs to handle situations where UAs do not follow restrictions associated with the Info Package.

10.9. Rate of INFO Requests

If there is a maximum or minumum rate at which UAs can send INFO requests associated with the Info Package within a dialog, the Info Package specification MUST document the rate values.

If the rates can vary, the Info Package specification MAY define Info Package parameters that UAs can use to indicate or negotiate the rates. Alternatively the rate information can be part of the application data information associated with the Info Package.

10.10. Info Package Security Considerations

If the application information carried in INFO requests associated with the Info Package requires certain level of security, the Info Package specification MUST describe the mechanisms that UAs need to use in order to provide the required security.

If the Info Package specification does not require any additional security, other than what the underlying SIP protocol provides, it MUST 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 beyond 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. One way of achieving this is to use end-to-end security techniques such as S/MIME [RFC3851].

10.11. Implementation Details

It is strongly RECOMMENDED that the Info Package specification defines the procedure how implementors shall implement and use the Info Package, or refer to other locations where implementors can find that information.

NOTE: Sometimes Info Package designer might choose to not reveal the details of an Info Package. However, in order to allow multiple implementations to support the Info Package, Info Package designers are stronly encouraged to provide the implementation details.

10.12. Examples

It is RECOMMENDED that the Info Package specification provides demonstrative message flow diagrams, paired with complete messages and message descriptions.

Note that example flows are by definition informative, and do not replace normative text.

11. IANA Considerations

11.1. 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]

11.2. 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]

11.3. 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]

11.4. Creation of the Info Packages Registry

Please create a subregistry in the SIP Parameters registry for Info Packages.

The registration policy for the registry is Specification Required [RFC5226].

The reviewer does not consider the applicability of the Info Package for the usage for which it is defined.

The following data elements populate the Info Package Registry.

- Info Package Name: The Info Package Name is a case insensitive token. In addition, IANA shall not register multiple Info Package names that have identical case-insensitive values.
- Reference: A reference to a specification which describes the Info Package.

The initial population of this table shall be:

Name	Reference

11.5. Registration of the Info-Package Content-Disposition

Please add the following new header field value to the Content-Disposition registry.
Name: info-package
Description: the body contains information associated with an Info Package
Reference: RFCXXXX

11.6. SIP Response Code 469 Registration

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

12. Examples

12.1. Indication for which Info Packages UAs are willing to receive INFO requests

12.1.1. Initial INVITE request

The UAC sends an initial INVITE request, where the UAC indicates that it is willing to receive INFO requests for 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 requests for 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:bob@pc33.example.com>
Recv-Info: R, T
Content-Type: application/sdp
Content-Length: ...

...
```

The UAC sends an ACK request.

```
ACK sip:bob@pc33.example.com SIP/2.0
Via: SIP/2.0/TCP pc33.example.com;branch=z9hG4bK754
Max-Forwards: 70
To: Bob <sip:bob@example.com>;tag=a6c85cf
From: Alice <sip:alice@example.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.example.com
CSeq: 314159 ACK
Content-Length: 0
```

12.1.2. Target refresh

The UAC sends an UPDATE request within the invite dialog usage, where the UAC indicates (using an empty Recv-Info header field) that it is not willing to receive INFO requests for any Info Packages.

```
UPDATE sip:bob@pc33.example.com SIP/2.0
Via: SIP/2.0/TCP pc33.example.com;branch=z9hG4bK776
Max-Forwards: 70
To: Bob <sip:bob@example.com>;tag=a6c85cf
From: Alice <sip:alice@example.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.example.com
CSeq: 314163 UPDATE
Recv-Info:
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 requests for Info Packages R, T.

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP pc33.example.com;branch=z9hG4bK893;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: 314163 INVITE
Contact: <sip:alice@pc33.example.com>
Recv-Info: R, T
Content-Type: application/sdp
Content-Length: ...

...
```

12.2. INFO request associated with Info Package

12.2.1. Single payload

The UA sends an INFO request associated with Info Package foo.

```
INFO sip:alice@pc33.example.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.2:5060;branch=z9hG4bKnabcdef
To: Bob <sip:bob@example.com>;tag=a6c85cf
From: Alice <sip:alice@example.com>;tag=1928301774
Call-Id: a84b4c76e66710@pc33.example.com
CSeq: 314333 INFO
Info-Package: foo
Content-type: application/foo
Content-Disposition: Info-Package
Content-length: 24

I am a foo message type
```

12.2.2. Multipart INFO

12.2.2.1. Non-Info Package body part

SIP extensions can sometimes add body part payloads into an INFO request, independent of the Info Package. In this case, the Info Package payload gets put into a Multipart MIME body, with a Content-Disposition header field that indicates which body part is associated with the Info Package.

```
INFO sip:alice@pc33.example.com 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: a84b4c76e66710@pc33.example.com
CSeq: 314400 INFO
Info-Package: foo
Content-Type: multipart/mixed;boundary="theboundary"
Content-Length: ...

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

<mumble stuff>

--theboundary
Content-Type: application/foo-x
Content-Disposition: Info-Package
Content-length: 59

I am a foo-x message type, and I belong to Info Package foo
--theboundary--
```

12.2.2.2. Info Package with multiple body parts inside multipart body part

Multiple body part payloads can be associated with a single Info Package. In this case, the body parts are put into a Multipart MIME body, with a Content-Disposition header field that indicates which body part is associated with the Info Package.

```
INFO sip:alice@pc33.example.com 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: a84b4c76e66710@pc33.example.com
CSeq: 314423 INFO
Info-Package: foo
Content-Type: multipart/mixed;boundary="theboundary"
Content-Disposition: Info-Package
Content-Length: ...

--theboundary
Content-Type: application/foo-x
Content-length: 59

I am a foo-x message type, and I belong to Info Package foo

<mumble stuff>

--theboundary
Content-Type: application/foo-y
Content-length: 59

I am a foo-y message type, and I belong to Info Package foo
--theboundary--
```

12.2.2.3. Info Package with single body part inside multipart body part

The body part payload associated with the Info Package can have a Content-Disposition header field value other than "Info-Package". In this case, the body part is put into a Multipart MIME body, with a Content-Disposition header field that indicates which body part is associated with the Info Package.

```
INFO sip:alice@pc33.example.com 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: a84b4c76e66710@pc33.example.com
CSeq: 314423 INFO
Info-Package: foo
Content-Type: multipart/mixed;boundary="theboundary"
Content-Disposition: Info-Package
Content-Length: ...

--theboundary
Content-Type: application/foo-x
Content-Disposition: icon
Content-length: 59

I am a foo-x message type, and I belong to Info Package foo
--theboundary--
```

13. Security Considerations

By eliminating multiple usages of INFO messages without adequate community review and by eliminating the possibility for rogue SIP UAs from confusing another UA 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 mechanism 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, UAs 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 request transports application level information. One implication of this is INFO messages may require a higher level of protection than the underlying SIP dialog 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.

14. References

14.1. Normative References

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

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

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 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.

[RFC5621] Camarillo, G., "Message Body Handling in the Session Initiation Protocol (SIP)", RFC 5621, September 2009.

14.2. 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.

[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.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[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.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[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.

[I-D.peterson-rai-rfc3427bis] Peterson, J., Jennings, C., and R. Sparks, "Change Process for the Session Initiation Protocol (SIP) and the Real-time Applications and Infrastructure Area", draft-peterson-rai-rfc3427bis-04 (work in progress), October 2009.

[W3C.REC-voicexml21-20070619] Oshry, M., Rehor, K., Bodell, M., Burke, D., Auburn, R., Baggia, P., McGlashan, S., Candell, E., Burnett, D., Carter, J., Lee, A., and B. Porter, "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-mrcpv2-20 (work in progress), August 2009.

[I-D.saleem-msml]
Saleem, A. and G. Sharratt, "Media Server Markup Language (MSML)", draft-saleem-msml-09 (work in progress), July 2009.

[Ecma-355]
"Standard ECMA-355 Corporate Telecommunication Networks - Tunnelling of QSIG over SIP", ECMA http://www.ecma-international.org/publications/standards/Ecma-355.htm, June 2008.

Appendix A. Legacy INFO Usage

A.1. General

This section provides examples of existing legacy INFO usages. The section is not meant to be a comprehensive catalog of legacy INFO usages, but it should give the reader a flavor for current legacy INFO usages.

A.2. ISUP

[RFC3372] specifies the encapsulation of ISUP in SIP message bodies. ITU-T and 3GPP have specified similar procedures.

A.3. QSIG

[Ecma-355] specifies the encapsulation of QSIG in SIP message bodies.

A.4. MSCML

[RFC5022] specifies how INFO is used as a transport mechanism by the MSCML protocol. MSCML uses an option-tag in the Require header field to ensure that the receiver understands the INFO content.

A.5. MSML

[I-D.saleem-msml] specifies how INFO us used as a transport mechanism by the MSML protocol.

A.6. Video Fast Update

Companies have been using INFO messages in order to request fast video update. Currently a standardized mechanism, based on RTCP, has been specified in [RFC5168]

Appendix B. Acknowledgements

The work on this document was influenced by the "INFO Considered Harmful" draft (26 December 2002) written by Jonathan Rosenberg, and by the "Packaging and Negotiation of INFO Methods for the Session Initiation Protocol" draft (15 January 2003) written by Dean Willis.

The following individuals have been involved in the work, and have provided input and feedback on this document:

Adam Roach, Anders Kristensen, Andrew Allen, Arun Arunachalam, Ben Campbell, Bob Penfield, Bram Verburg, Brian Stucker, Chris Boulton, Christian Stredicke, Cullen Jennings, Dale Worley, Dean Willis, Eric Rescorla, Frank Miller, Gonzalo Camarillo, Gordon Beith, Henry Sinnreich, Inaki Baz Castillo, James Jackson, James Rafferty, Jeroen van Bemmel, Joel Halpern, John Elwell, Johnathan Rosenberg, Juha Heinanen, Gordon Beith, 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, Roni Evan Salvatore Loreto, Sam Ganesan, Sanjay Sinha, Spencer Dawkins, Steve Langstaff, Sumit Garg and Xavier Marjoum.

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

Arun Arunachalam, Brett Tate, John Elwell, Keith Drage and Robert Sparks provided valuable feedback during the WGLC process, in order to prepare this document for publication.

Adam Roach, Dean Willis, John Elwell and Paul Kyzivat provided valuable input in order to sort out the message body part usage for Info Packages.

Appendix C. Change Log

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

Changes from draft-ietf-sipcore-info-events-04

- Further changes based on WGLC comments
- OPTIONS processing removed
- Clarification of Recv-Info header field in INFO 469 response added
- IANA registry procedures clarified

Changes from draft-ietf-sipcore-info-events-03

- Further changes based on WGLC comments
- New section 3.2.3 added

Changes from draft-ietf-sipcore-info-events-02
- Further changes based on WGLC comments
- Allignment with "specification" and "definition" terminology
- Location switch of sections 3 and 4
- Corrections in header table
- IANA Info Package registration input changed
- Clarifiaction regarding which SIP messages can contain the Recv-Info header field
- Recv-Info 'nil' value removed
- Rules on usage of Recv-Info header clarified
- Recv-Info fallback rules added
- Additional examples added

Changes from draft-ietf-sipcore-info-events-01
- Further changes based on WGLC comments
- Appending A moved into the main part of the document
- Section name changed from "Modifications to SIP Change Process" to "Security Considerations"
- "Syntax" section moved further up in the document
- Clarification on usage of Info Package related message body parts, and the usage of the Content-Disposition header field with those body parts
- Removed REFER and NOTIFY from the INFO Headers table
- Clarified usage of the Recv-Info header field in the REGISTER and OPTIONS requests
- Major re-write of the Introduction section
- Text about legacy INFO and subscription-based events moved from the Introduction to the main part of the document
- Wording about receiving Info-Packages has been replaced with wording about receiving INFO requests for Info-Packages
- The text about the usage of message body, and body parts, associated with Info Packages, has been clarified

Changes from draft-ietf-sip-info-events-04
- Major re-write of the document, due to problems to implement WGLC comments into the existing text structure
- Wording allignment
- Clarification or roles

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

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas, 02420
Finland

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

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: