

SIPCORE
Internet-Draft
Obsoletes: RFC [2976](#)
(if approved)
Expires: April 26, 2010

E. Burger
NeuStar, Inc.
H. Kaplan
Acme Packet
C. Holmberg
Ericsson
October 23, 2009

Session Initiation Protocol (SIP) INFO Method and Package Framework
draft-ietf-sipcore-info-events-02

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 26, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document defines a new 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](#)].

Table of Contents

1.	Introduction	5
2.	Applicability	6
3.	Info Package Support	6
3.1.	General	6
3.2.	User Agent Behavior	6
3.3.	Package Versioning	7
3.4.	REGISTER Processing	7
3.5.	OPTIONS Processing	8
4.	The INFO Method	8
4.1.	General	8
4.2.	INFO Request	8
4.3.	INFO Request Message Body	9
4.4.	INFO Response	10
4.5.	INFO Response Message Body	11
4.6.	Order of Delivery	11
5.	Formal INFO Method Definition	11
5.1.	INFO Method	11
6.	INFO Header Fields	13
6.1.	General	13
6.2.	Info-Package header field	13
6.3.	Recv-Info header field	14
7.	Info Package Considerations	14
7.1.	General	14
7.2.	Appropriateness of Info Package Usage	14
7.3.	Dialog Fate Sharing	14
7.4.	INFO Request Rate and Volume	14
7.5.	Alternative Mechanisms	15
7.5.1.	Alternative SIP signaling plane mechanisms	15
7.5.2.	Media Plane Mechanisms	16
7.5.3.	Non-SIP related mechanisms	17
8.	Syntax	17
8.1.	General	17
8.2.	ABNF	17
9.	Legacy INFO Usage	17
9.1.	General	17
9.2.	Problems	18
9.3.	Co-existence with Info Package based INFO usage	18
10.	Info Package Requirements	19
10.1.	General	19
10.2.	Applicability	19
10.3.	Info Package Name	19
10.4.	Info Package Parameters	20
10.5.	SIP Option Tags	20
10.6.	INFO Message Bodies	20
10.7.	Info Package Usage Restrictions	20
10.8.	Rate of INFO Requests	21

10.9.	IANA Registrations	21
10.10.	Info Package Security Considerations	21
10.11.	Application Procedures	22
10.12.	Examples	22
11.	IANA Considerations	22
11.1.	Update to Registration of SIP INFO Method	22
11.2.	Registration of the Info-Package Header Field	22
11.3.	Registration of the Recv-Info Header Field	23
11.4.	Creation of the Info Packages Registry	23
11.5.	Registration of the Info-Package Content-Disposition	24
11.6.	SIP Response Code 469 Registration	24
12.	Examples	24
12.1.	Indication of which Info Packages UAs are willing to receive INFO requests within an invite dialog usage	24
12.2.	INFO request with information associated with a simple Info Package	25
12.3.	Multipart INFO Example	25
13.	Security Considerations	26
14.	References	27
14.1.	Normative References	27
14.2.	Informative References	28
Appendix A.	Legacy INFO Usages	30
A.1.	General	30
A.2.	ISUP	30
A.3.	QSIG	30
A.4.	MSCML	30
A.5.	MSML	31
A.6.	Video Fast Update	31
Appendix B.	Acknowledgements	31
Appendix C.	Change Log	31
	Authors' Addresses	33

1. Introduction

This document defines a new 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 may update the state of those applications).

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 for which Info Packages they are willing to receive INFO requests. The document defines how the INFO method is used, new SIP header fields for the INFO method, and how to transport payload information associated with an Info Package using INFO requests.

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.

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 the Recv-Info header field with a 'nil' 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.

This document obsoletes [[RFC2976](#)]. However, for backward compatibility it 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.

2. Applicability

This document defines a new 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. Info Package Support

3.1. General

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

3.2. User Agent Behavior

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 request. 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 dialog 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.

If a UA is not willing to INFO requests for any Info Packages, during dialog establishment or later during the invite dialog usage, the UA MUST indicate this by including a Recv-Info header field with a 'nil' value. 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', 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 target refresh request with a Recv-Info header field with a header value of 'nil'.

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.

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, and a dialog has been established with the remote UA.

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 that the UA wishes to only receive INFO request for one of the Info Packages. It is up to the application logic associated with the Info Packages, and specific Info Package descriptions 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 [Section 9](#).

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 10.5](#).

For backward compatibility, 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 MUST still also explicitly indicate support of the INFO method using the Allow header field.

[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 with the Info Package, but that is outside the scope of the 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.

[3.4.](#) 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.

NOTE: 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](#)].

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 to receive.

NOTE: As for any other capability and extension, for a specific dialog UAs need to indicate which Info Packages they are willing to receive within that dialog.

4. The INFO Method

4.1. General

This section describes the UA handling of INFO requests and responses, and message bodies carried in INFO messages.

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.

4.2. INFO Request

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. For a specific dialog, a UA MUST NOT send INFO requests associated with Info Packages that the remote UA has not indicated that it is willing to receive.

A UA can send an INFO requests associated with a legacy INFO usage [Section 9](#). 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 UA cannot use the Recv-Info header field to indicate whether it is willing to receive INFO requests associated with that legacy INFO usage.

The INFO method MUST NOT be used outside an invite dialog usage. The INFO method has no lifetime beyond its transaction or usage of its own. UAs indicate, per-dialog basis, for which Info Packages they

are willing to receive INFO requests. The set of Info Packages cannot automatically be used within other dialogs.

Due to the possibility of forking, a UAC which, during the early dialog phase indicates that it is willing to receive INFO requests for one or more Info Packages MUST be prepared to receive INFO requests associated with those Info Packages from multiple remote UAs. Note that each remote UA can indicate a different set of Info Packages for which they are willing to receive INFO request.

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 within early and confirmed dialogs.

The INFO request MUST NOT contain a Recv-Info header field. The 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 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 is carried as payload in the message body of the INFO request, using one or more body parts.

Info Package specifications MUST describe the application level information associated with the Info Package. Each body part MUST have a MIME type value, and the syntax and content of the body part, defined.

Each body part, when associated with an Info Package, MUST have a Content-Disposition header field with an 'Info-Package' value assigned, in order to be able distinguish body parts associated with the Info Package from other body parts.

NOTE: Some SIP functions that are orthogonal to INFO may insert body parts unrelated to the Info Package.

Body parts associated with specific MIME types may sometimes have specific Content-Disposition header field values defined for them. For example, for body parts with a 'text/plain' MIME a Content-Disposition header field with a 'render' value is often assigned. However, when a body part in the INFO message is associated with an Info Package, it MUST always have a Content-Disposition header field with an 'Info-Package' value assigned. The Info Package specification defines how applications process the body part contents.

If a SIP message body contains multiple body parts, multipart body parts [[RFC5621](#)] are used to separate them. If all body parts within a multipart body part are associated with the Info Package, the multipart body part SHALL have a Content-Disposition header field with an 'Info-Package' value assigned to it. However, each body part within the multipart body part MUST still have a Content-Disposition header field with an 'Info-Package' value assigned to them, in order to avoid that the parser assigns a default Content-Disposition header value to the body part.

NOTE: According to [[RFC5621](#)], body parts within a multipart are not implicitly assigned the Content-Disposition header field value of the multipart body part which they belong to.

This document does not define Info Package specific rules on how body parts associated with Info Packages are to be inserted into multipart body parts, and what type of multipart are used. If an Info Package requires special rules regarding the usage of multipart body parts, the specification for that Info Package MUST specify such rules.

UAs MUST conform to [[RFC5621](#)] to support multipart body parts.

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

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

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

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 [Section 11.6](#). In the terminology of Multiple Dialog Usages [[RFC5057](#)], this represents a Transaction Only failure.

If a UA receives an INFO request that 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 that 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.

[4.5.](#) INFO Response Message Body

The Info Package mechanism allows a SIP stack to generate a response to an INFO request without application interaction. As a result, Info Packages cannot require a message body in INFO responses, require different response codes, or otherwise require the response to the INFO request to contain application information. If the application needs to send information in the other direction, it can send a new INFO request which contains the information.

[4.6.](#) Order of Delivery

The Info Package mechanism 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

6. INFO Header Fields

6.1. General

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	-	-	-
Recv-Info	2xx	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

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 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](#)].

[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 3](#) 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.](#) 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 usage is terminated, while in other cases it is desirable to maintain the dialog usage.

[7.4.](#) 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 user plane data transport mechanisms.

[7.5.](#) Alternative Mechanisms

[7.5.1.](#) Alternative SIP signaling plane mechanisms

[7.5.1.1.](#) General

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

[7.5.1.2.](#) 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 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.

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 new session and a subscription dialog usage [[RFC5057](#)] which is separate, and does not share the fate any other sessions.

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 sessions needs to be considered. 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 dialog route set, but do not have any interest in the application information transported between the end users.

7.5.1.3. MESSAGE

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

7.5.2. Media Plane Mechanisms

7.5.2.1. 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 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.

7.5.2.2. 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.

7.5.2.3. MRSP

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

7.5.3. 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-voicexml121-20070619](#)] script.

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. 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 4](#), an INFO request associated with an Info Package always contains an Info-Package header field. A legacy INFO request MUST NOT contain an Info-Package header field.

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 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, or document why an issue is not applicable for the specific Info Package.

10.2. 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 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.

10.3. Info Package Name

The Info Package specification **MUST** define a for Info Package name (e.g. "Info Package for X").

The specification **MUST** also define the header field value (e.g. "infoX") 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](#).

[10.4.](#) 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. They MUST NOT be used for other Info Packages.

NOTE: Info Package parameters defined for specific Info Packages may 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.

[10.5.](#) 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 [[I-D.peterson-rai-rfc3427bis](#)].

[10.6.](#) INFO Message Bodies

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

If multiple body parts are used with an Info Package, the Info Package specification MUST define whether there are special rules on how the body parts are to be inserted in multipart body parts, and what types of multipart to use.

[10.7.](#) 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 are 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.

As the SIP stack may not be aware of Info Package specific restrictions, it cannot be assumed that overlapping requests would be rejected. As defined in [Section 4.4](#), in most cases a 200 OK response will be sent for the INFO request. The application logic associated with the Info Package needs to handle situations which can occur due to overlapping requests.

[10.8.](#) Rate of INFO Requests

The Info Package specification MUST specify 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.

[10.9.](#) 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.

[10.10.](#) Info Package 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, this fact 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 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. 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.

10.12. 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 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. This subregistry has a modified First Come First Served [[RFC5226](#)] policy.

The following data elements populate the Info Package Registry.

- o 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.
- o 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.
- o Info Package Payload MIME Types: A list of zero or more registered MIME types from the MIME Type Registry.
- o Standards Status: Values are "Standards Track" or empty. See below for a discussion and rules on this field.
- o 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]

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 of which Info Packages UAs are willing to receive INFO requests within an invite dialog usage

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: ...
```

...

The UAC sends ACK.

```
ACK sip:ngw1@a.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
```

12.2. INFO request with information associated with a 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

12.3. Multipart INFO Example

Other SIP extensions can sometimes add payload body parts 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@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--
```

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.
- [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.
- [RFC4497] Elwell, J., Derks, F., Mourrot, 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.
- [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)", [draft-peterson-rai-rfc3427bis-03](#) (work in progress), September 2009.
- [W3C.REC-voicexml21-20070619]
McGlashan, S., Lee, A., Carter, J., Porter, B., Auburn, R., Oshry, M., Rehor, K., Bodell, M., Burke, D., Baggia, P., Candell, E., and D. Burnett, "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](http://www.ecma-international.org/publications/standards/Ecma-355.htm), June 2008.

[Appendix A.](#) Legacy INFO Usages

[A.1.](#) General

This section provides examples of existing legacy INFO usages. This 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 is 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 Bommel, 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.

John Elwell and Robert Sparks provided valuable feedback during the WGLC process, in order to prepare this document for publication.

Appendix C. Change Log

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

Changes from [draft-ietf-sipcore-info-events-01](#)

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

- o Major re-write of the document, due to problems to implement WGLC comments into the existing text structure
- o Wording alignment
- o Clarification of roles

Changes from [draft-ietf-sip-info-events-03](#)

- o Clarified Abstract language
- o All SIP dialogs are now referred to as sessions
- o Clarified the image example in the Introduction
- o Clarified the relationship (none) between SIP Event Packages and SIP Info Packages
- o Really, really clarified the protocol is NOT a negotiation but an advertisement
- o Split [Section 3](#) into UAS and UAC behavior
- o Moved the example in [section 3](#) into its own sub-section, and used full SIP header fields
- o Clarified forking behavior
- o Clarified language around when to send a body
- o Added 469 error response, instead of reusing 489
- o Clarified overlapping INFO method handling
- o Fixed table 1 to follow 3261, not 2543
- o Added REFER to the INFO Headers table
- o replaced token-nodot with token for Info-Package header field values
- o Clarified end-to-end security considerations
- o Info Package parameters are semi-colon delimited, not dot delimited

Changes from -02

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

Changes from -01

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

Changes from -00

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

