Internet Engineering Task Force (IETF)
Request for Comments: 7462
Updates: 3261
Category: Standards Track
ISSN: 2070-1721

L. Liess, Ed.
R. Jesske
Deutsche Telekom AG
A. Johnston
Avaya
D. Worley
Ariadne
P. Kyzivat
Huawei

March 2015

URNs for the Alert-Info Header Field of the Session Initiation Protocol (SIP)

Abstract

The Session Initiation Protocol (SIP) supports the capability to provide a reference to a specific rendering to be used by the User Agent (UA) as an alerting signal (e.g., a ring tone or ringback tone) when the user is alerted. This is done using the Alert-Info header field. However, the reference (typically a URL) addresses only a specific network resource with specific rendering properties. There is currently no support for standard identifiers for describing the semantics of the alerting situation or the characteristics of the alerting signal, without being tied to a particular rendering. To overcome these limitations and support new applications, a new family of URNs for use in Alert-Info header fields (and situations with similar requirements) is defined in this specification.

This document normatively updates RFC 3261, which defines the Session Initiation Protocol (SIP). It changes the usage of the Alert-Info header field defined in RFC 3261 by additionally allowing its use in any non-100 provisional response to INVITE. This document also permits proxies to add or remove an Alert-Info header field and to add or remove Alert-Info header field values.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7462.

Copyright Notice

Table of Contents

1. Introduction

The Session Initiation Protocol (SIP) [RFC3261] includes a means to suggest to a User Agent (UA) a particular ringback tone or ring tone to be used during session establishment. In [RFC3261], this is done by including a URI, in the Alert-Info header field, that specifies a reference to the tone. The URI is most commonly the HTTP URL to an audio file. On the receipt of the Alert-Info header field, the UA may fetch the referenced ringback tone or ring tone and play it to the user.

This mechanism hinders interoperability when there is no common understanding of the meaning of the referenced tone, which might be country- or vendor-specific. It can lead to problems for the user trying to interpret the tone and for the UA wanting to substitute its own tone (e.g., in accordance with user preferences) or provide an alternative alerting mode (e.g., for deaf and hard-of-hearing users). If the caller and the callee are from different countries, their understanding of the tones may differ significantly. Deaf or hard-of-hearing users may not sense the specific tone if it is provided as an audio file. The tone, per se, is also not useful for automata.

Another limitation of using URLs of audio files is that the referenced tones are tied to particular renderings. There is no method to signal the semantic intention of the alert while enabling the recipient UA to choose the specific alert indication (such as a particular tone, vibration, or visual display) to use to signal the intention. Similarly, there is no method to signal particular rendering features (such as short duration, delay, or country-specific conventions).

The issues with URLs that reference audio files can be avoided by using fixed URLs with specific meanings. However, this approach has its own interoperability issues. For example, consider the Private Branch Exchange (PBX) special ring tone for an external (to the PBX) caller. Different vendors use different approaches such as:

```
Alert-Info: <file://ring.pcm>;alert=external
```

where ring.pcm is a dummy file name, or:

```
Alert-Info: <file://external.ring.pcm>
```

```
Alert-Info: <sip:external-ringtone@example.com>
```

As a result, the Alert-Info header field currently only works when the same vendor provides a PBX and UA, and only then if the same artificial proprietary URI convention is used.

To solve the described issues, this specification defines the new URN namespace "alert" for the SIP Alert-Info header field that allows for programmatic user interface adaptation and for conversion of equivalent alerting tones in the Public Switched Telephone Network (PSTN) when the client is a gateway. The work to standardize an "alert" URN will increase SIP interoperability for this header field by replacing proprietary conventions used today.

The "alert" namespace provides a syntax for several different application spaces, for example:

- Names for service indications, such as call waiting or automatic callback, not tied to any particular rendering.
- Names for common ring tones generated by PBX phones for cases such as an internal enterprise caller, external caller, ringback tone after a transfer failure or expiration of a hold timer, etc.
- Names for country-specific ringback tones.
- Names for things with specific renderings that aren’t purely audio. They might be static icons, video sequences, text, etc.

Some advantages of a URN rather than a URL of a downloadable resource:

- There is no need to download it or deal with security issues associated with dereferencing.
- There are no formatting or compatibility issues.
- There is no security risk of rendering something unexpected and undesirable.
- The tone can be stored locally in whatever format and at whatever quality level is appropriate, because it is specified "by name" rather than "by value".
- It is easier to make policy decisions about whether or not to use it.
- It facilitates translation for the deaf and hard of hearing.

The downside is that if the recipient does not understand the URN, then it will only be able to render a default ringback tone or ring tone.

This document creates a new URN namespace and registry for alert indications and registers some initial values.

In practice, this specification extends the usage of the Alert-Info header field in that it will cause the use of a new class of URIs and the use of multiple URIs. Backward compatibility issues are not expected, as devices that do not understand an "alert" URN should ignore it, and devices should not malfunction upon receiving multiple Alert-Info header field values (<alert-param>s in [RFC3261]) (which was syntactically permitted before, but rarely used).

2. Requirements Language

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 [RFC2119].

3. Terminology

This specification uses a number of terms to refer to the roles involved in the use of alerting indications in SIP. A "specifier" sends an "alerting indication" (one or more URNs in an Alert-Info header field) to a "renderer", which then "renders" a "signal" or "rendering" based on the indication to a human user. A "category" is a characteristic whose "values" can be used to classify indications.

This specification uses the terms "ring tone" and "ringback tone". A "ring tone" or "calling signal" (terminology used in [E182]) is a signal generated by the callee’s end device, advising the callee about an incoming call. A "ringback tone" or "ringing tone" (terminology used in [E182]) is a signal advising the caller that a connection has been made and that a ring tone is being rendered to the callee.

4. Updates to RFC 3261

4.1. Allow Alert-Info in Provisional Responses

This specification changes the usage of the Alert-Info header field defined in [RFC3261] by additionally allowing its use in any non-100 provisional response to INVITE.

Previously, the Alert-Info header field was only permitted in 180 (Ringing) responses. But in telephony, other situations indicated by SIP provisional responses, such as 181 (Call Is Being Forwarded) and 182 (Call Is Being Queued), are often indicated by tones. Extending the applicability of the Alert-Info header field allows the telephony practice to be implemented in SIP.

To support this change, the following paragraph replaces the the first paragraph of Section 20.4 of [RFC3261]:

> When present in an INVITE request, the Alert-Info header field specifies an alternative ring tone to the User Agent Server (UAS). When present in a non-100 provisional response, the Alert-Info header field specifies an alternative ringback tone to the UAC. A typical usage is for a proxy to insert this header field to provide a distinctive ring feature.

4.2. Proxies May Alter Alert-Info Header Fields

A SIP proxy MAY add or remove an Alert-Info header field, and it MAY add or remove Alert-Info header field values, in a SIP request or a non-100 provisional response.

5. Requirements

This section discusses the requirements for an alerting indication to transport the semantics of the alerting situation or the characteristics of the rendering.

REQ-1: The mechanism will allow UAs and proxies to provide in the Alert-Info header field an alerting indication that describes the semantics of the signaling situation or the characteristics of the rendering and allows the recipient to decide how to render the received information to the user.

REQ-2: The mechanism will allow the alerting indication to be specified "by name" rather than "by value", to enable local policy decisions whether or not to use it.

REQ-3: The mechanism will enable alerting indications to represent a wide variety of signals, which have many largely orthogonal characteristics.

REQ-4: The mechanism will enable the set of alerting indications to support extensibility by a wide variety of organizations that are not coordinated with each other. Extensions will be able to:

- add further values to any existing category
- add further categories that are orthogonal to existing categories
- semantically subdivide the meaning provided by any existing indication

REQ-5: The mechanism will be flexible, so new alerting indications can be defined in the future, when SIP-applications evolve. For example, "alert" URNs could identify specific media by name, such as "Beethoven's Fifth", and the end device could render some small part of it as a ring tone.

REQ-6: The mechanism will provide only an indication capability, not a negotiation capability.

REQ-7: The mechanism will not require an alerting indication to depend on context provided by a previous alerting indication in either direction.

REQ-8: The mechanism will allow transmission in the Alert-Info header field of SIP INVITE requests and provisional 1xx responses excepting the 100 responses.

REQ-9: The mechanism will be able to accommodate both renderers that are customized with a limited or uncommon set of signals that they can render and renderers that are provided with a set of signals that have uncommon semantics. (The canonical example is a UA for the deaf and hard of hearing, customized with an alternative set of signals, video or text instead of audio. By REQ-6, the renderer has no way of transmitting this fact to the specifier.)

REQ-10: The mechanism will allow an alerting indication to reliably carry all extensions if the specifier and the renderer have designs that are properly coordinated.

REQ-11: The mechanism will allow a renderer to select a tone that approximates to that intended by the specifier if the renderer is unable to provide the precise tone indicated.

REQ-12: The mechanism will support alerting indications relating to services such as call waiting, call forwarding, transfer recall, auto callback, and hold recall.

REQ-13: The mechanism will allow rendering common PBX ring tone types.

REQ-14: The mechanism will allow rendering specific country ringback tones.

REQ-15: The mechanism will allow rendering tones for emergency alerts. (Use cases and definitions of URN values for emergency calls are not a subject of this specification.)

REQ-16: The mechanism will allow rendering using other means than tones, e.g., text or images.

REQ-17: The mechanism will allow PSTN gateways to map ring/ringback tones from legacy protocols to SIP at the edge of a network, e.g., national ring tones as defined in TIA/EIA-41-D and 3GPP2 A.S0014. (Use cases and values definition for this situation are not a subject of this specification.)

REQ-18: The mechanism will ensure that if an UA receives "alert" URNs or portions of an "alert" URN it does not understand, it can ignore them.

REQ-19: The mechanism will allow storage of the actual encoding of the rendering locally rather than fetching it.

REQ-20: The mechanism must provide a simple way to combine two or more alerting indications to produce an alerting indication that requests a combination of the intentions of the two alerting indications, where any contradictions or conflicts between the two alerting indications are resolved in favor of the intention of the first alerting indication.

6. Use Cases

This section describes some use cases for which the "alert" URN mechanism is needed today.

6.1. PBX Ring Tones

This section defines some commonly encountered ring tones on PBX or business phones. They are as listed in the following subsections.

6.1.1. Normal

This tone indicates that the default or normal ring tone should be rendered. This is essentially a no-operation "alert" URN and should be treated by the UA as if no "alert" URN is present. This is most useful when Alert-Info header field parameters are being used. For example, in [RFC7463], an Alert-Info header field needs to be present containing the "appearance" parameter, but no special ring tone needs to be specified.

6.1.2. External

This tone is used to indicate that the caller is external to the enterprise or PBX system. This could be a call from the PSTN or from a SIP trunk.

6.1.3. Internal

This tone is used to indicate that the caller is internal to the enterprise or PBX system. The call could have been originated from another user on this PBX or on another PBX within the enterprise.

6.1.4. Priority

A PBX tone needs to indicate that a priority level alert should be applied for the type of alerting specified (e.g., internal alerting).

6.1.5. Short

In this case, the alerting type specified (e.g., internal alerting) should be rendered shorter than normal. In contact centers, this is sometimes referred to as "abbreviated ringing" or a "zip tone".

6.1.6. Delayed

In this case, the alerting type specified should be rendered after a short delay. In some bridged-line/shared-line-appearance implementations, this is used so that the bridged line does not ring at exactly the same time as the main line but is delayed a few seconds.

6.2. Service Tones

These tones are used to indicate specific PBX and public network telephony services.

6.2.1. Call Waiting

The call-waiting service [TS24.615] permits a callee to be notified of an incoming call while the callee is engaged in an active or held call. Subsequently, the callee can either accept, reject, or ignore the incoming call. There is an interest on the caller side to be informed about the call-waiting situation on the callee side. Having this information the caller can decide whether to continue waiting for callee to pickup or better to call some time later when it is estimated that the callee could have finished the ongoing conversation. To provide this information, a callee’s UA (or proxy) that is aware of the call-waiting condition can add the call-waiting indication to the Alert-Info header field in the 180 (Ringing) response.

6.2.2. Forward

This feature is used in a 180 (Ringing) response when a call forwarding feature has been initiated on an INVITE. Many PBX system implement a forwarding "beep" followed by normal ringing to indicate this. Note that a 181 response can be used in place of this URN.

6.2.3. Transfer Recall

This feature is used when a blind transfer [RFC5589] has been performed by a server on behalf of the transferor and fails. Instead of failing the call, the server calls back the transferor, giving them another chance to transfer or otherwise deal with the call. This service tone is used to distinguish this INVITE from a normal incoming call.

6.2.4. Auto Callback

This feature is used when a user has utilized a server to implement an automatic callback service [RFC6910]. When the user is available, the server calls back the user and utilizes this service tone to distinguish this INVITE from a normal incoming call.

6.2.5. Hold Recall

This feature is used when a server implements a call hold timer on behalf of an endpoint. After a certain period of time of being on hold, the user who placed the call on hold is alerted to either retrieve the call or otherwise dispose of the call. This service tone is used to distinguish this case from a normal incoming call.

6.3. Country-Specific Ringback Tone Indications for the Public Switched Telephone Network

In the PSTN, different tones are used in different countries. End users are accustomed to hear the callee’s country ringback tone and would like to have this feature for SIP.

7. URN Specification for the "alert" Namespace Identifier

This section provides the registration template for the "alert" URN namespace identifier (NID) according to [RFC2141] and [RFC3406].

Namespace ID: alert

Registration Information:
Registration version: 1
Registration date: 2014-12-10

Declared registrant of the namespace:
Registering organization: Real-time Applications and Infrastructure Area, IETF
Designated contact: RAI Area Director
Designated contact email: rai-ads@ietf.org

Declaration of syntactic structure:

The Namespace Specific String (NSS) for the "alert" URNs is called an <alert-identifier> and has a hierarchical structure. The first colon-separated part after "alert" is called the <alert-category>; the parts to the right of that are <alert-ind-part>s, and together form the <alert-indication>. The general form is urn:alert:<alert-category>:<alert-indication>.

The following <alert-category> identifiers are defined in this document: "service" , "priority" , "source" , "duration", "delay", and "locale". The <alert-category> set can be extended in the future, either by standardization or by private action. The <alert-category>s describe distinct features of alerting signals.

Any "alert" URN defined in this specification is syntactically valid for ring and ringback tones and can be used in SIP INVITE requests or in provisional 1xx responses excepting the 100 response.

The ABNF [RFC5234] for the "alert" URNs is shown below:

```
alert-URN         = "urn:alert:" alert-identifier
alert-identifier  = alert-category ":" alert-indication
alert-category    = alert-name
alert-indication  = alert-ind-part *(":" alert-ind-part)
alert-ind-part    = alert-name
alert-name        = alert-label / private-name
private-name      = alert-label "@" provider
provider          = alert-label
alert-label       = let-dig [ *let-dig-hyp let-dig ]
let-dig-hyp       = let-dig / "-"
let-dig           = ALPHA / DIGIT
ALPHA             = %x41-5A / %x61-7A   ; A-Z / a-z
DIGIT             = %x30-39 ; 0-9
```

<alert-label>s MUST comply with the syntax for Non-Reserved LDH labels [RFC5890]. Registered URNs and components thereof MUST be transmitted as registered (including case).

Relevant ancillary documentation: RFC 7462

Namespace considerations: This specification defines a URN namespace "alert" for URNs representing signals or renderings that are presented to users to inform them of events and actions. The initial usage is to specify ring tones and ringback tones when dialogs are established in SIP, but they can also be used for other communication-initiation protocols (e.g., H.323), and more generally, in any situation (e.g., web pages or endpoint device software configurations) to describe how a user should be signaled.

An "alert" URN does not describe a complete signal, but rather it describes a particular characteristic of the event it is signaling or a feature of the signal to be presented. The complete specification of the signal is a sequence of "alert" URNs specifying the desired characteristics/significance of the signal in priority order, with the most important aspects specified by the earlier URNs. This allows the sender of a sequence of URNs to compose very detailed specifications from a restricted set of URNs, and to clearly specify which aspects of the specification it considers most important.

The initial scope of usage is in the Alert-Info header field, in initial INVITE requests (to indicate how the called user should be alerted regarding the call) and non-100 provisional (1xx) responses to those INVITE requests (to indicate the ringback, how the calling user should be alerted regarding the progress of the call).

In order to ensure widespread adoption of these URNs for indicating ring tones and ringback tones, the scheme must allow replication of the current diversity of these tones. Currently, these tones vary between the PSTNs of different nations and between equipment supplied by different vendors. Thus, the scheme must accommodate national variations and proprietary extensions in a way that minimizes the information that is lost during interoperation between systems that follow different national variations or that are supplied by different vendors.

The scheme allows definition of private extension URNs that refine and extend the information provided by standard URNs. Private extension URNs can also refine and extend the information provided by other private extension URNs. Private extensions can also define entirely new categories of information about calls. We expect these extensions to be used extensively when existing PBX products are converted to support SIP operation.

The device that receives an Alert-Info header field containing a sequence of "alert" URNs provides to the user a rendering that represents the semantic content of the URNs. The device is given great leeway in choosing the rendering, but it is constrained by rules that maximize interoperability between systems that support different sets of private extensions. In particular, earlier URNs in the sequence have priority of expression over later URNs in the sequence, and URNs that are not usable in their entirety (because they contain unknown extensions or are incompatible with previous URNs) are successively truncated in attempt to construct a URN that retains some information and is renderable in the context.

Due to the practical importance of private extensions for the adoption of URNs for alerting calls and the very specific rules for private extensions and the corresponding processing rules that allow quality interoperation in the face of private extensions, the requirements of the "alert" URN scheme cannot be met by a fixed enumeration of URNs and corresponding meanings. In particular, the existing namespace "urn:ietf:params" does not suffice (unless the private extension apparatus is applied to that namespace).

There do not appear to be other URN namespaces that uniquely identify the semantic of a signal or rendering feature. Unlike most other currently registered URN namespaces, the "alert" URN does not identify documents and protocol objects (e.g., [RFC3044], [RFC3120], [RFC3187], [RFC3188], [RFC4179], [RFC4195], [RFC4198]), types of telecommunications equipment [RFC4152], people, or organizations [RFC3043].

The <alert-URN>s are hierarchical identifiers. An <alert-URN> asserts some fact or feature of the offered SIP dialog, or some fact or feature of how it should be presented to a user, or of how it is being presented to a user. Removing an <alert-ind-part> from the end of an <alert-URN> (which has more than one <alert-ind-part>) creates a shorter <alert-URN> with a less specific meaning; the set of dialogs to which the longer <alert-URN> applies is necessarily a subset of the set of dialogs to which the shorter <alert-URN> applies. (If the starting <alert-URN> contains only one <alert-ind-part>, and thus the <alert-ind-part> cannot be removed to make a shorter <alert-URN>, we can consider the set of dialogs to which the <alert-URN> applies to be a subset of the set of all dialogs.)

The specific criteria defining the subset to which the longer <alert-URN> applies, within the larger set of dialogs, is considered to be the meaning of the final <alert-ind-part>. This meaning is relative to and depends upon the preceding <alert-

category> and <alert-ind-part>s (if any). The meanings of two <alert-ind-part>s that are textually the same but are preceded by different <alert-category>s or <alert-ind-part>s have no necessary connection. (An <alert-category> considered alone has no meaning in this sense.)

The organization owning the <provider> within a <private-name> specifies the meaning of that <private-name> when it is used as an <alert-ind-part>. (The organization owning a <provider> is specified by the registry described in Section 9.3.)

The organization owning the <provider> within a <private-name> (in either an <alert-category> or an <alert-ind-part>) specifies the meaning of each <alert-ind-part>, which is an <alert-label> that follows that <private-name> and that precedes the next <alert-ind-part> which is a <private-name> (if any).

The meaning of all other <alert-ind-part>s (i.e., those that are not <private-name>s and do not follow a <private-name>) is defined by standardization.

Community considerations: The "alert" URNs are relevant to a large cross-section of Internet users, namely those that initiate and receive communication connections via the Session Initiation Protocol. These users include both technical and non-technical users, on a variety of devices and with a variety of perception capabilities. The "alert" URNs will allow Internet users to receive more information about offered calls and enable them to better make decisions about accepting an offered call, and to get better feedback on the progress of a call they have made.

User interfaces that utilize alternative sensory modes can better render the ring and ringback tones based on the "alert" URNs because the URNs provide more detailed information regarding the intention of communications than is provided by current SIP mechanisms.

Process of identifier assignment:

Assignment of standardized "alert" URNs is by insertion into the IANA registry described in Section 9.2. This process defines the meanings of <alert-ind-part>s that have standardized meanings, as described in "Namespace Considerations".

A new URN MUST NOT be registered if it is equal by the comparison rules to an already registered URN.

Private extensions are "alert" URNs that include <alert-ind-part>s that are <private-name>s and <alert-label>s that appear after a <private-name> (either as an <alert-category> or an <alert-indication>). If such an <alert-ind-part> is a <private-name>, its meaning is defined by the organization that owns the <provider> that appears in the <private-name>. If the <alert-ind-part> is an <alert-label>, its meaning is defined by the organization that owns the <provider> that appears in the closest <private-name> preceding the <alert-label>. The organization owning a <provider> is specified by the registry described in Section 9.3.

Identifier uniqueness and persistence considerations: An "alert" URN identifies a semantic feature of a call or a sensory feature of how the call alerting should be a rendered at the caller's or callee's end device.

For standardized <alert-ind-part>s in URNs, uniqueness and persistence of their meanings is guaranteed by the fact that they are registered with IANA in accordance with the procedures of Section 9.2; the feature identified by a particular "alert" URN is distinct from the feature identified by any other standardized "alert" URN.

Assuring uniqueness and persistence of the meanings of private extensions is delegated to the organizations that define private extension <alert-ind-part>s. The organization responsible for a particular <alert-ind-part> in a particular "alert" URN is the owner of a syntactically determined <provider> part within the URN.

An organization SHOULD use only one <provider> value for all of the <private-name>s it defines.

Process for identifier resolution: The process of identifier resolution is the process by which a rendering device chooses a rendering to represent a sequence of "alert" URNs. The device is allowed great leeway in making this choice, but the process MUST obey the rules of Section 11.1. The device is expected to provide renderings that users associate with the meanings assigned to the URNs within their cultural context. A non-normative example resolution algorithm is given in Section 12.1.

Rules for lexical equivalence: "alert" URNs are compared according to case-insensitive string equality.

Conformance with URN syntax: All "alert" URNs must conform to the ABNF in the "Declaration of Syntactic Structure" in Section 7. That ABNF is a subset of the generic URN syntax [RFC2141]. <alert-label>s are constrained to be Non-Reserved LDH labels [RFC5890], that is, "ordinary ASCII labels". Future standardization may allow <alert-label>s that are A-labels [RFC5890], and so interpreters of "alert" URNs MUST operate correctly (per Section 11.1) when given such URNs as input.

Validation mechanism: An "alert" URN containing no private extensions can be validated based on the IANA registry of standardized "alert" URNs. Validating an "alert" URN containing private extensions requires obtaining information regarding the private extensions defined by the organization that owns the <provider> in the relevant <private-name>. The identity of the organization can be determined from the IANA registry described in Section 9.2. However, if an "alert" URN contains at least one <alert-identifier> that precedes the first <private-name>, the portion of the "alert" URN that precedes the first <private-name> must itself be a valid standardized "alert" URN, which may be validated as above.

Scope: The scope for this URN is public and global.

8. "alert" URN Values

8.1. <alert-category> Values

The following <alert-category> values are defined in this document:

- service
- source
- priority
- duration
- delay
- locale

8.2. <alert-indication> Values

This section describes the "alert" URN indication values for the <alert-category>s defined in this document.

For each <alert-category>, a default <alert-indication> is defined, which is essentially a no-operation "alert" URN and should be treated by the UA as if no "alert" URN for the respective category is present. "alert" URN default indications are most useful when Alert-Info header field parameters are being used. For example, in

[RFC7463], an Alert-Info header field needs to be present containing the "appearance" parameter, but no special ringtone need be specified.

The <private-name> syntax is used for extensions defined by independent organizations, as described in Section 10.2.

8.2.1. <alert-indication> Values for the <alert-category> "service"

- normal (default)
- call-waiting
- forward
- recall:callback
- recall:hold
- recall:transfer
- <private-name>

Examples: <urn:alert:service:call-waiting> or <urn:alert:service:recall:transfer>.

8.2.2. <alert-indication> Values for the <alert-category> "source"

- unclassified (default)
- internal
- external
- friend
- family
- <private-name>

(These <alert-indication>s will rarely be provided by the sending UA; rather they will usually be inserted by a proxy acting on behalf of the recipient UA to inform the recipient UA about the origins of a call.)

Examples: <urn:alert:source:external>.

8.2.3. <alert-indication> Values for the <alert-category> "priority"

- normal (default)
- low
- high
- <private-name>

Examples: <urn:alert:priority:high>.

8.2.4. <alert-Indication> Values for the <alert-category> "duration"

- normal (default)
- short
- long
- <private-name>

Examples: <urn:alert:duration:short>.

8.2.5. <alert-indication> Values for the <alert-category> "delay"

- none (default)
- yes
- <private-name>

Examples: <urn:alert:delay:yes>.

8.2.6. <alert-indication> Values for the <alert-category> "locale"

- default (default)
- country:<ISO 3166-1 country code>
- <private-name>

The ISO 3166-1 country code [ISO3166-1] is used to inform the renderer on the other side of the call that a country-specific rendering should be used. For example, to indicate ringback tones from South Africa, the following URN would be used: <urn:alert:locale:country:za>.

9. IANA Considerations

9.1. URN Namespace Identifier "alert"

This section registers a new URN namespace identifier (NID), "alert", in accordance with [RFC3406] with the registration template provided in Section 7.

9.2. 'Alert URN Identifiers' Registry

Standard "alert" URNs are recorded as <alert-identifier>s in a new registry called "Alert URN Identifiers". Thus, creating a new standard "alert" URN requires IANA action. IANA manages the "Alert URN Identifiers" registry under the policy 'Specification Required' [RFC5226] following the guidelines in Section 10.1.

The registry contains entries in the following formats:

<alert-category>/ <alert-identifier>	Reference	Description
foo	[RFCxyz]	Description of the 'foo' <alert-category>;
foo:bar	[RFCabc]	Description of the 'foo:bar' <alert-identifier>
foo:<range> 'foo:<category>'	[RFCdef]	Description of the <alert-identifer>s (which will reference the <range> value)

The first value in each row is the value that is registered, which is either: (1) an <alert-category> value, (2) an <alert-identifier> value, composed of an <alert-category> followed by an <alert-indication>, in turn composed of one or more <alert-label>s, or (3) a pattern for <alert-identifier> values (e.g., for the "locale" <alert-category> in Section 9.2.1.6).

The second value in each row is the reference to the required specification for the value.

The third value in each row is a short description of the semantics of the value.

A new URN MUST NOT be registered if it is equal by the comparison rules (that is, case-insensitive string comparison) to an already registered URN.

<alert-category> and <alert-identifier> values that contain <private-name>s are not managed by IANA. The process of assigning these values is described in Section 10.2.

9.2.1. Initial IANA Registration

This document defines the <alert-category>s 'service', 'source', 'priority', 'duration', 'delay' and 'locale'. The entries to be added to the 'Alert URN Identifiers' registry table for each <alert-category> are given in the respective sections below.

9.2.1.1. The "service" <alert-category> and <alert-identifier>s

The following table contains the initial IANA registration for the "service" <alert-category> and <alert-identifier>s. The value of this indicator is set to a value different from "normal" if the caller or callee is informed that a specific telephony service has been initiated.

<alert-category>/ <alert-identifier>	Reference	Description
service	RFC 7462	Specific telephony service used in this call
service:normal	RFC 7462	Normal ring/ringback rendering (default value)
service:call-waiting	RFC 7462	Call waiting was initiated at the other side of the call
service:forward	RFC 7462	Call has been forwarded
service:recall:callback	RFC 7462	Recall due to callback
service:recall:hold	RFC 7462	Recall due to call hold
service:recall:transfer	RFC 7462	Recall due to transfer

9.2.1.2. The "source" <alert-category> and <alert-identifier>s

The following table contains the initial IANA registration for the "source" <alert-category> and <alert-identifier>. The value of this indicator provides information about the user at the other side of the call.

<alert-category>/ <alert-identifier>	Reference	Description
source	RFC 7462	Classification of the other party to the call
source:unclassified	RFC 7462	Unclassified ring/ringback rendering (default value)
source:internal	RFC 7462	User at the other side of the call is internal to the enterprise or PBX system
source:external	RFC 7462	User at the other side of the call is external to the enterprise or PBX system
source:friend	RFC 7462	User at the other side of the call is a friend
source:family	RFC 7462	User at the other side of the call is a family member

9.2.1.3. The "priority" <alert-category> and <alert-identifier>s

The following table contains the initial IANA registration for the "priority" <alert-category> and <alert-identifier>s. The value of this indicator provides information about the priority the alerted user should give to the call.

<alert-category>/ <alert-identifier>	Reference	Description
priority	RFC 7462	Priority of the call
priority:normal	RFC 7462	Normal ring/ringback rendering (default value)
priority:low	RFC 7462	Low priority call
priority:high	RFC 7462	High priority call

9.2.1.4. The "duration" <alert-category> and <alert-identifier>s

The following table contains the initial IANA registration for the "duration" <alert-category> and <alert-identifier>s. The value of this indicator provides information about the duration of the alerting signals compared to the default alerting signals.

<alert-category>/ <alert-identifier>	Reference	Description
duration	RFC 7462	Duration of alerting signal
duration:normal	RFC 7462	Normal ring/ringback rendering (default value)
duration:short	RFC 7462	Shorter than normal
duration:long	RFC 7462	Longer than normal

9.2.1.5. The "delay" <alert-category> and <alert-identifier>s

The following table contains the initial IANA registration for the "delay" <alert-category> and <alert-identifier>s. The value of this indicator provides information about whether the presentation of the alerting signal should be delayed compared to the default presentation process. For more details see Section 6.1.6.

<alert-category>/ <alert-identifier>	Reference	Description
delay	RFC 7462	Delay of rendering of alerting signal
delay:none	RFC 7462	Immediate alerting (default value)
delay:yes	RFC 7462	Delayed alerting

9.2.1.6. The "locale" <alert-category> and <alert-identifier>s

The following table contains the initial IANA registration for the "locale" <alert-category> and <alert-identifier>s. The value of this indicator provides information about whether the alerting signals characteristic of the specified location should be used.

<alert-category>/ <alert-identifier>	Reference	Description
locale	RFC 7462	Location-specific alerting signals
locale:default	RFC 7462	Alerting not location specific (default value)
locale:country:<ISO 3166-1 country code>	RFC 7462	Alerting according to the conventions of the specified country

9.3. 'Alert URN Providers' Registry

Values of <provider>, which are used to create <private-name>s, are recorded in a new registry called "Alert URN Providers". (Private extension "alert" URNs that are defined are not recorded by IANA.) The registry is managed by IANA under the policy 'First Come First Served' [RFC5226].

The registry contains entries in the following format:

<provider>	Registrant	Contact URI
example	IETF	rai-ads@ietf.org

The first value in each row is the <provider> value that is registered. This value is case-insensitive and MUST comply with the syntax for Non-Reserved LDH labels [RFC5890].

The second value in each row is the name of the registrant of the value.

The third value is a contact URI for the registrant.

The registry initially contains the one entry shown above, which can be used for constructing examples of private extension URNs.

10. Extension Rules

10.1. General Extension Rules

The set of "alert" URNs is extensible. An extension "at the top level" creates a new <alert-category> (which represents a new alerting characteristic), an extension "at the second level" creates a new <alert-indication> value for an existing <alert-category>, an extension "at the third level" creates a subdivision of an existing <alert-indication> (that has one <alert-ind-part>), etc. URNs allow (in principle) indefinite subdivision of existing <alert-indication> values, although most of the standard "alert" URNs have only one level of subdivision and a few have two levels of subdivision.

Extensions, either standard or private, MUST conform to the following principles:

A new <alert-category> is independent of all previously existing <alert-category>s: For any combination of one <alert-identifier> in the new <alert-category> with any one <alert-identifier> in any of the previously existing <alert-category>s, there are potential calls to which the combination can be meaningfully applied.

A new <alert-identifier> that has more than one <alert-ind-part> is a semantic refinement of a parent <alert-identifier>, the parent being obtained by deleting the final <alert-ind-part>. The new <alert-identifier> has as parent the most specific previously existing <alert-identifier> whose meaning includes all potential calls to which the new <alert-identifier> could be meaningfully applied.

A new <alert-identifier> has no semantic overlap with any sibling <alert-identifier> (<alert-identifier>s that differ only in the final <alert-ind-part>). That is, there could be no call to which both <alert-identifier>s could be meaningfully applied.

The process for defining new standard "alert" URNs is described in Section 9.2; all such definitions require registering a publicly available specification. The process for defining new "alert" URNs via the private extension mechanism is described in Section 10.2.

10.2. Private Extension Rules

The <private-name> syntax is used to create private extensions, extensions that are not registered with IANA. The <private-name> has the form of an <alert-label> followed by "@" and then a <provider> that designates the organization defining the extension. Both <alert-label> and <provider> have the same syntax as an ordinary ASCII DNS label. A private extension URN is created by using a <private-name> as either an <alert-category> or an <alert-ind-part>.

If the <private-name> is used as an <alert-category>, the characteristic of the alerting signal that the <alert-category> describes is defined by the organization. If the <private-name> is used as the first <alert-ind-part>, the organization defines an alternative value for the standardized <alert-category> of the URN. If the <private-name> is used as the second or later <alert-ind-part>, the organization defines the meaning of the URN as a subset of the meaning of the shorter URN resulting when the <private-name> (and any subsequent <alert-ind-part>s) are removed.

Within a URN, all <alert-label> components that follow a <private-name> but are before any following <private-name>s are additional private extensions whose meaning is defined by the organization defining the nearest preceding <private-name>.

A URN that contains a private extension can be further subdivided by the private extension of a different organization: the second organization appends an <alert-ind-part> that is a <private-name> containing a the <provider> value for the second organization.

The meaning of a <private-name> or an <alert-label> that is defined privately (because of a preceding <private-name>) is only fixed within the context provided by the sequence of preceding <alert-name>s; these components have no meaning in isolation and there is no necessary relationship between the meaning of textually identical <alert-name>s that are preceded by different sequences of <alert-name>s.

Creating private extension "alert" URNs is not a Standards Action and they are not registered with IANA.

The organization defining a private extension is responsible for ensuring persistence of the meaning of the private extension.

Private extensions MUST conform to the principles of Section 10.1, both in regard to previously existing standard <alert-URN>s and in regard to any previously existing private extensions using the same <provider> value, and any other private extensions that the organization is aware of. In particular, a private extension MUST NOT duplicate any standard URN or any private extension that the organization is aware of. (In either of those cases, the organization MUST use the existing URN for its purposes.)

An organization obtains a <provider> value for constructing <private-name>s by registering the value with IANA as provided in Section 9.3.

10.3. Examples

10.3.1. Subsetting an Existing URN

The organization registering the <provider> "example" can define distinctive versions of <urn:alert:service:call-waiting>:

urn:alert:service:call-waiting:abc@example

urn:alert:service:call-waiting:def@example

It can create a more specialized URN that applies to a subset of the situations to which the first URN above applies:

urn:alert:service:call-waiting:abc@example:xyz

Because "xyz" follows "abc@example" (and there is no intervening <private-name>), its meaning is defined by the owner of the <provider> "example".

10.3.2. A New Value within an <alert-category>

The organization registering the <provider> "example" can define URNs in the "service" category to express a new service that is not covered by any of the standardized URNs:

urn:alert:service:ghi@example

However, before defining such a URN, the organization should verify that the set of calls to which the URN applies is not a subset of the set of calls for some existing URN. If it is a subset, the extension URN should be a subdivision of the existing URN.

10.3.3. A New <alert-category>

The organization registering the <provider> "example" can define an extension <alert-category> named "jkl@example" with two <alert-indication>s "a1" and "a2":

urn:alert:jkl@example:a1

urn:alert:jkl@example:a2

10.3.4. Subsetting a Private Extension URN

The organization registering the <provider> "foo" wants to define a set of URNs that specify the different ring patterns used by a "distinctive ring" service to alert for incoming calls that are directed to different directory numbers. These ring patterns are composed of groups of ring sounds that have particular patterns of lengths.

The company can create a private <alert-category> "distinctive@foo", and within it assign three ’alert’ URNs that indicate the three different ring patterns used by the company’s service:

urn:alert:distinctive@foo:long-long

urn:alert:distinctive@foo:short-long-short

urn:alert:distinctive@foo:short-short-long

Later, the company registering the <provider> "bar" wants to define an additional ’alert’ URN for the ring pattern "short short", which it uses to support a fourth directory number for a phone instrument.

The company can create a <private-name> to be used with the "distinctive@foo" <alert-category>:

urn:alert:distinctive@foo:short-short@bar

11. Combinations of "alert" URNs

11.1. Priority Rules

This section describes combination rules for the case when all the Alert-Info header fields only contain "alert" URNs. Other combinations of URIs in the Alert-Info header fields of the same SIP message are not defined in this specification.

In many cases, more than one URN will be needed to fully define a particular tone. This is done by including multiple "alert" URNs, in one or more Alert-Info header fields in a request or a response. For example, an internal, priority call could be indicated by Alert-Info: <urn:alert:source:internal>, <urn:alert:priority:high>. A priority call-waiting tone could be indicated by Alert-Info: <urn:alert:service:call-waiting>, <urn:alert:priority:high>.

The sender of the Alert-Info header field may include an arbitrary list of "alert" URNs, even if they are redundant or contradictory. An earlier URN has priority over any later contradictory URN. This allows any element to modify a list of URNs to require a feature value (by adding a URN at the beginning of the list) or to suggest a feature value (by adding a URN at the end of the list).

The receiving UA matches the received "alert" URN combination with the signal(s) it is able to render.

The implementation is free to ignore an "alert" URN if it does not recognize the URN, or if it is incapable of rendering its effect in the context. Similarly, it can remove a final series of one or more <alert-ind-part>s of an "alert" URN to create a "more generic" URN that it recognizes and whose meaning it can render in the context.

The exact way in which a UA renders a received combination of "alert" URNs is left as an implementation issue. However, the implementation MUST comply to following rules:

(a) Each "alert" URN has precedence over all URNs that follow it, and its interpretation is subordinate to all URNs that precede it.

(b) If the UA cannot implement the effect of a URN (because it does not recognize the URN or the URN's effect is precluded by preceding URNs), the UA repeatedly removes the final <alert-ind-part> of the URN until either:

(1) the resulting URN is recognized and can be given effect by some signal (without reducing the degree of expression of any preceding URN), or

(2) the resulting URN is reduced to having no <alert-ind-part> in which case, that URN in the series cannot be given effect, and so is ignored.

(c) In case that after processing all the received URNs, the UA can generate more than one signal that are equally effective at expressing the URNs (under the preceding rules), one of those signals is selected. When selecting from the set of equally effective signals, the least specific signal in the set should be chosen: a signal should not be chosen if a less-specific signal is also in the set. (Specificity is to be judged based on the defined meanings of the signals to the user.) (For example, if each signal is considered to express certain <alert-indication>s of certain <alert-category>s, one signal is less-specific than a second signal if the first signal's <alert-indication>s are a subset or are prefixes of the second signal's <alert-indication>s.) However, a more-specific signal may be chosen if the choice is based on information derived from the containing SIP message. For example, a signal implying <urn:alert:priority:high> may be chosen if the SIP message contains the header field "Priority: urgent".

In all situations, the set of signals that can be rendered and their significances may change based on user preferences and local policy. In addition, the chosen signal may change based on the status of the UA. For example, if a call is active on the UA, all audible signals may become unavailable, or audible signals may be available only if <urn:alert:priority:high> is specified.

11.2. Multi-mode Signals

There are cases when the device can render two signal modes (e.g., audio and visual, or video and text) at the same time.

Formally, the device must be considered to be making its choice from the set of all combined signals that it can render (pairs of one signal from the first mode and one signal from the second mode), and that choice must conform to the above rules. However, it can be proven that if the device makes its rendering choice for each of the

two modes independently, with each choice separately conforming to the above rules, its combined choice also conforms to the above rules, when it is regarded as a choice from among all possible combinations.

In such a situation, it may simplify implementation to make each choice separately. It is an implementation decision whether to chose from among combined signals or to combine choices made from each signal mode.

12. Non-normative Algorithm for Handling Combinations of URNs

The following text is a non-normative example of an algorithm for handling combinations of URNs that complies with the rules in Sections 10 and 11. Thus, it demonstrates that the rules are consistent and implementable. (Of course, a device may use any other algorithm that complies with Sections 10 and 11.)

12.1. Algorithm Description

For each <alert-category> (feature) known by the implementation, there is a "feature tree" of the known <alert-indication>s for that <alert-category>, with the sequence of <alert-ind-part>s in an <alert-indication> specifying the path in the tree from the root to the node representing the <alert-indication>. For this description, we will name each tree and its root node by the <alert-category> name, and name each non-root node by the <alert-identifier>. Each URN thus corresponds to one non-root node in one feature tree. For example, there is a tree named "source", whose root node is also named "source", and which has the children source:internal, source:external, source:friend, and source:family. The URN <urn:alert:source:external> is placed at the node "source:external" in the "source" tree. If the implementation understands <urn:alert:source:foo@example>, there is a node source:foo@example that is a child of node "source". If the implementation understands <urn:alert:source:external:bar@example>, there is a node source:external:bar@example that is a child of node source:external. (Of course, there are an infinite number of potential additional nodes in the tree for private values, but we don’t have to represent those nodes explicitly unless the device has a signal representing the private value.)

We assign similar locations to signals, but each signal has a position in *every* tree, describing the specific combination of meanings that it carries. If a signal has a simple meaning, such as "external source", its place in the "source" tree is source:external,

showing that it carries the "external source" meaning, but its place in every other feature tree is at the root node, meaning that it has no particular meaning for those features.

A signal that has a complex meaning may have non-root positions in more than one feature tree. For example, an "external, high priority" signal would be placed at source:external and priority:high in those trees, but be at the root in all other feature trees.

In order to assure that the algorithm always selects at least one signal, we require that there is a "default" signal, whose position in every feature tree is at the root. This default signal will never be excluded from the set of acceptable signals for any set of URNs, but will be the lowest priority signal for any set of URNs.

The algorithm proceeds by considering each URN in the received Alert-Info header fields from left to right, while revising a set of signals. The set of signals starts as the entire set of signals available to the device. Each URN excludes some signals from the set, and "sorts" the signals that remain in the set according to how well they represent the URN. (The details of these operations are described below.) The first URN is the "major sort", and has the most influence on the position of a signal in the set. The second URN is a "minor sort", in that it arranges the orders of the signals that are tied within the first sort, the third URN arranges the orders of the signals that are tied within the first two sorts, etc.

At the end of the algorithm, a final, "most minor" sort is done, which orders the signals that remain tied under all the sorts driven by the URNs. This final sort places the least specific signals (within their tied groups) "first". (If one signal’s position in each feature tree is ancestral or the same as a second signal’s position in that tree, the first signal is "less specific" than the second signal. Other cases are left to the implementation to decide.)

Once all the URNs are processed and the sorting of the signals that have not been excluded is done, the device selects the first signal in the set.

Here is how a single sort step proceeds, examining a single URN to modify the set of signals (by excluding some signals and further sorting the signals that remain):

- The URN specifies a specific node in a specific feature tree.

- All signals in the set that are, within that feature tree, positioned at the URN's node, or at an ancestor node of the URN's node, are kept. All other signals are removed from the set (because they have meanings that are incompatible with the URN's meaning).

- Each group of signals that are tied under the previous sorts are further sorted into groups based on how much of the URN's meaning they represent: those which are positioned at the node of the URN are tied for first position, those which are positioned at the parent node of the URN are tied for second position, etc., and those which are positioned at the root node of the feature tree are tied for last position.

12.2. Examples of How the Algorithm Works

The following examples show how the algorithm described in the previous section works:

12.2.1. Example 1

The device has a set of four alerting signals. We list their primary meanings, and the locations that they are placed in the feature trees:

Signal 1

```
Meaning: external
Locations:
- source:external
- priority (that is, the root node of the priority tree)
```

Signal 2

```
Meaning: internal
Locations:
- source:internal
- priority
```

Signal 3

```
Meaning: low
Locations:
- source
- priority:low
```

Signal 4

Meaning: high
Locations:
- source
- priority:high

To which we add:

Signal 5

Meaning: default
Locations:
- source
- priority

If the device receives <urn:alert:source:internal>, then the sort is:

Signals at source:internal: (this is, first place)

Signal 2: internal

Signals at source: (tied for second place)

Signal 3: low
Signal 4: high
Signal 5: default

And these signals are excluded from the set:

Signal 1: external

So, in this example, the sorting algorithm properly gives first place to Signal 2 "internal".

12.2.2. Example 2

Let us add to the set of signals in Example 1 ones that express combinations like "internal, high priority", but let us specifically exclude the combination "internal, low priority" so as to set up some tricky examples. This enlarges our set of signals:

Signal 1

Meaning: default
Locations:
- source
- priority

Signal 2

Meaning: external
Locations:
- source:external
- priority

Signal 3

Meaning: internal
Locations:
- source:internal
- priority

Signal 4

Meaning: low
Locations:
- source
- priority:low

Signal 5

Meaning: high
Locations:
- source
- priority:high

Signal 6

Meaning: external high
Locations:
- source:external
- priority:high

Signal 7

Meaning: external low
Locations:
- source:external
- priority:low

Signal 8

Meaning: internal high
Locations:
- source:internal
- priority:high

If the device receives <urn:alert:source:internal>, then the sort is:

Signals at source:internal: (that is, tied for first place)

- Signal 3: internal
- Signal 8: internal high

Signals at source: (tied for second place)

- Signal 4: low
- Signal 5: high
- Signal 1: default

Signals excluded from the set:

- Signal 2: external
- Signal 7: external low
- Signal 6: external high

Two signals are tied for the first place, but the final sort orders them:

- Signal 3: internal
- Signal 8: internal high

because it puts the least-specific signal first. So, the Signal 3 "internal" is chosen.

12.2.3. Example 3

The same device receives <urn:alert:source:external>, <urn:alert:priority:low>. The first sort (due to <urn:alert:source:external>) is:

Signals at source:external:

- Signal 2: external
- Signal 7: external low
- Signal 6: external high

Signals at source:

- Signal 4: low
- Signal 5: high
- Signal 1: default

Signals excluded:

- Signal 3: internal
- Signal 8: internal high

The second sort (due to <urn:alert:priority:low>) puts signals at priority:low before signals at priority, and excludes signal at priority:high:

- Signal 7: external low
- Signal 2: external
- Signal 4: low
- Signal 1: default

Excluded:

- Signal 6: external high
- Signal 5: high
- Signal 3: internal
- Signal 8: internal high

So, we choose Signal 7 "external low".

12.2.4. Example 4

Suppose the same device receives <urn:alert:source:internal>, <urn:alert:priority:low>. Note that there is no signal that corresponds to this combination.

The first sort is based on source:internal, and results in this order:

- Signal 3: internal
- Signal 8: internal high
- Signal 4: low
- Signal 5: high
- Signal 1: default

Excluded:

- Signal 2: external
- Signal 7: external low
- Signal 6: external high

The second sort is based on priority:low, and results in this order:

- Signal 3: internal
- Signal 4: low
- Signal 1: default

Excluded:

- Signal 8: internal high
- Signal 5: high
- Signal 7: external low
- Signal 2: external
- Signal 6: external high

So, we choose the Signal 3 "internal".

Note that <urn:alert:priority:low> could not be given effect because it followed <urn:alert:source:internal>. If the two URNs had appeared in the reverse order, the Signal 2 "external" would have been chosen, because <urn:alert:priority:low> would have been given precedence.

12.2.5. Example 5

Let us set up a simple set of signals, with three signals giving priority:

Signal 1

Meaning: default
Locations:
- priority

Signal 2

Meaning: low
Locations:
- priority:low

Signal 3

Meaning: high
Locations:
- priority:high

Notice that we've used the "default" signal to cover "normal priority". That is so the signal will cover situations where no priority URN is present, as well as the ones with <urn:alert:priority:normal>. So, we're deliberately failing to distinguish "priority:normal" from the default priority.

If the device receives <urn:alert:priority:low>, the sort is:

- Signal 2: low
- Signal 1: default

Excluded:

- Signal 3: high

and Signal 2 "low" is chosen.

Similarly, if the device receives <urn:alert:priority:high>, Signal 3 is chosen.

If the device receives <urn:alert:priority:normal>, the sort is:

-Signal 1 :default

Excluded:

- Signal 2: low
- Signal 3: high

and Signal 1 "default" is chosen.

If no "priority" URN is received, Signal 1 "default" will be put before Signal 2 "low" and Signal 3 "high" by the final sort, and so it will be chosen.

13. User Agent Behaviour

A SIP UA MAY add a URN or multiple URNs to the Alert-Info header field in a SIP request or a provisional 1xx response (excepting a 100 response) when it needs to provide additional information about the call or about the provided service.

Upon receiving a SIP INVITE request or a SIP provisional response with an Alert-Info header field that contains a combination of Alert-Info URNs, the UA attempts to match the received Alert- Info URNs combination with a signal it can render. The process the UA uses MUST conform to the rules described in Section 11. (A non-normative algorithm example for the process is described in Section 12.)

The UA must produce a reasonable rendering regardless of the combination of URIs (of any schemes) in the Alert-Info header field: it MUST produce a rendering based on the URIs that it can understand and act on (if any), interpreted as prescribed by local policy, and ignore the other URIs. In particular, unless the containing message is a request and is immediately rejected, the UA SHOULD provide some alert unless it is instructed not to (for example, by Alert-Info URIs that it understands, the presence of a Replaces or Joins header field, local policy, or direction of the user).

Subsequent provisional responses, even within the same dialog, may contain different Alert-Info header field values. The Alert-Info header field values received within different provisional responses are treated independently. If subsequent provisional responses containing different Alert-Info header field values were received within the same dialog, the UA SHOULD render, at any time, the last received Alert-Info header field value. The handling of provisional responses containing different Alert-Info header field values that were not received within the same dialog is left as an implementation issue.

14. Proxy Behaviour

A SIP proxy MAY add or remove an Alert-Info header field, and MAY add or remove Alert-Info header field values, in a SIP request or a non-100 provisional response when it needs to modify the information about the call or about the provided services.

There are many reasons a proxy may choose do this, for example, (1) to add indications based on information that the proxy can determine about the call, such as that it is coming from an external source, or that the INVITE contains a "Priority: urgent" header field; (2) to add indication that a particular service is being invoked at this end of the call; (3) to remove undesirable indications, such as possibly deceptive indications from untrusted sources; and (4) to remove indications that contain information that should be suppressed for privacy reasons.

The following example shows a typical example of a 180 (Ringing) provisional response that has been modified by a proxy. The response sent by the UAS to the proxy was very similar, but had no Alert-Info header field. The proxy has added Alert-Info header field values specifying both a network audio resource referenced by the HTTP URI and the URN indication for the call-waiting service. This allows the UAC to render the network audio resource, to choose a rendering based on the URN, or to perform some combination of these actions. Due to Section 10, the UAC must produce some reasonable rendering in this situation.

```
SIP/2.0 180 Ringing
Alert-Info: <http://www.example.com/sound/moo.wav>,
             <urn:alert:service:call-waiting>
To: Bob <sip:bob@biloxi.example.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
Contact: <sip:bob@192.0.2.4>
CSeq: 314159 INVITE
Via: SIP/2.0/UDP server10.biloxi.example.com;
            branch=z9hG4bK4b43c2ff8.1
Content-Length: 0
```

15. Internationalization Considerations

The <alert-identifier> labels are protocol elements [RFC6365] and are not normally seen by users. Thus, the character set for these elements is restricted, as described in Section 7.

Allowance has been made for the possibility of internationalizing <alert-identifier>s by allowing them to be A-labels: a processor that does not understand such <alert-identifier>s is required to ignore them as specified in Sections 7 and 11.1.

The URNs <urn:alert:locale:country:<ISO 3166-1 country code>> select renderings that are conventional in the specified country.

16. Security Considerations

As an identifier, the "alert" URN does not appear to raise any particular security issues. The indications described by the "alert" URN are meant to be well-known.

However, the provision of specific indications may raise privacy issues by revealing information about the source UA, e.g., its nature, its dialog state, or services initiated at its end of the call. For example, call-waiting (Section 6.2.1) and call-forwarding (Section 6.2.2) services can reveal the dialog state of the UA. Such a provision SHALL always require authorization on behalf of the user of the source UA (usually through accessing configured policy). Authorization SHALL NOT assume that there is any limitation of the potential recipients of the indications without obtaining specific information about the SIP transaction.

Based on local policy, a UA MAY choose to ignore undesirable indications (e.g., possibly deceptive indications from untrusted sources), and it MAY choose not to send indications that are

otherwise valid in the context (e.g., for privacy reasons). A proxy acting on behalf of a UA MAY add or delete indications going to or from the UA for the same reasons.

Since the alert indications can be sensitive, end-to-end SIP encryption mechanisms using S/MIME MAY be used to protect it. UAs that implement alert indications SHOULD also implement SIP over TLS [RFC5246] and the sips: scheme [RFC5630].

17. References

17.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2141] Moats, R., "URN Syntax", RFC 2141, May 1997, <http://www.rfc-editor.org/info/rfc2141>.

[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, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3406] Daigle, L., van Gulik, D., Iannella, R., and P. Faltstrom, "Uniform Resource Names (URN) Namespace Definition Mechanisms", BCP 66, RFC 3406, October 2002, <http://www.rfc-editor.org/info/rfc3406>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, October 2009, <http://www.rfc-editor.org/info/rfc5630>.

17.2. Informative References

[E182] ITU-T, "Application of tones and recorded announcements in telephone services", ITU-T Recommendation E.182, 1998, <http://www.itu.int/rec/T-REC-E.182-199803-I/en>.

[ISO3166-1] ISO, "English country names and code elements", ISO 3166-1, <http://www.iso.org/iso/english_country_names_and_code_elements>.

[RFC3043] Mealling, M., "The Network Solutions Personal Internet Name (PIN): A URN Namespace for People and Organizations", RFC 3043, January 2001, <http://www.rfc-editor.org/info/rfc3043>.

[RFC3044] Rozenfeld, S., "Using The ISSN (International Serial Standard Number) as URN (Uniform Resource Names) within an ISSN-URN Namespace", RFC 3044, January 2001, <http://www.rfc-editor.org/info/rfc3044>.

[RFC3120] Best, K. and N. Walsh, "A URN Namespace for XML.org", RFC 3120, June 2001, <http://www.rfc-editor.org/info/rfc3120>.

[RFC3187] Hakala, J. and H. Walravens, "Using International Standard Book Numbers as Uniform Resource Names", RFC 3187, October 2001, <http://www.rfc-editor.org/info/rfc3187>.

[RFC3188] Hakala, J., "Using National Bibliography Numbers as Uniform Resource Names", RFC 3188, October 2001, <http://www.rfc-editor.org/info/rfc3188>.

[RFC4152] Tesink, K. and R. Fox, "A Uniform Resource Name (URN) Namespace for the Common Language Equipment Identifier (CLEI) Code", RFC 4152, August 2005, <http://www.rfc-editor.org/info/rfc4152>.

[RFC4179] Kang, S., "Using Universal Content Identifier (UCI) as Uniform Resource Names (URN)", RFC 4179, October 2005, <http://www.rfc-editor.org/info/rfc4179>.

[RFC4195] Kameyama, W., "A Uniform Resource Name (URN) Namespace for the TV-Anytime Forum", RFC 4195, October 2005, <http://www.rfc-editor.org/info/rfc4195>.

[RFC4198] Tessman, D., "A Uniform Resource Name (URN) Namespace for Federated Content", RFC 4198, November 2005, <http://www.rfc-editor.org/info/rfc4198>.

[RFC5589] Sparks, R., Johnston, A., and D. Petrie, "Session Initiation Protocol (SIP) Call Control - Transfer", BCP 149, RFC 5589, June 2009, <http://www.rfc-editor.org/info/rfc5589>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010, <http://www.rfc-editor.org/info/rfc5890>.

[RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, September 2011, <http://www.rfc-editor.org/info/rfc6365>.

[RFC6910] Worley, D., Huelsemann, M., Jesske, R., and D. Alexeitsev, "Completion of Calls for the Session Initiation Protocol (SIP)", RFC 6910, April 2013, <http://www.rfc-editor.org/info/rfc6910>.

[RFC7463] Johnston, A., Ed., Soroushnejad, M., Ed., and V. Venkataramanan, "Shared Appearances of a Session Initiation Protocol (SIP) Address of Record (AOR)", RFC 7463, March 2015, <http://www.rfc-editor.org/info/rfc7463>.

[TS24.615] 3GPP, "Communication Waiting (CW) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol Specification", 3GPP TS 24.615, September 2015.

Acknowledgements

The authors wish to thank Denis Alexeitsev, the editor of the initial version in BLISS, Anwar Siddiqui for his contributions to the document, Christer Holmberg for his careful review of the document, Adam Roach, Dean Willis, Martin Huelsemann, Shida Schubert, John Elwell, and Tom Taylor for their comments and suggestions and Alfred Hoenes for his extensive comments and proposals related to new namespace identifiers for URNs.

Authors' Addresses

Laura Liess (editor)
Deutsche Telekom AG
Heinrich-Hertz Str 3-7
Darmstadt, Hessen 64295
Germany

Phone: +49 6151 5812761
EMail: laura.liess.dt@gmail.com

Roland Jesske
Deutsche Telekom AG
Heinrich-Hertz Str. 3-7
Darmstadt, Hessen 64295
Germany

Phone: +49 6151 5812766
EMail: r.jesske@telekom.de

Alan Johnston
Avaya, Inc.
St. Louis, MO
United States

EMail: alan.b.johnston@gmail.com

Dale R. Worley
Ariadne Internet Services, Inc.
738 Main St.
Waltham, MA 02451
United States

Phone: +1 781 647 9199
EMail: worley@ariadne.com

Paul Kyzivat
Huawei
United States

EMail: pkyzivat@alum.mit.edu