

SIPPING
Internet-Draft
Expires: January 16, 2005

J. Rosenberg
dynamicsoft
H. Schulzrinne
Columbia University
O. Levin, Ed.
Microsoft Corporation
July 18, 2004

**A Session Initiation Protocol (SIP) Event Package for Conference
State
draft-ietf-sipping-conference-package-05**

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of [section 3 of RFC 3667](#). By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she become aware will be disclosed, in accordance with [RFC 3668](#).

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 January 16, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This document defines a conference event package for the Session Initiation Protocol (SIP) Events framework, along with a data format used in notifications for this package. The conference package

allows users to subscribe to a conference URI. Notifications are sent about changes in the membership of this conference and optionally about changes in the state of additional conference components.

Table of Contents

1.	Introduction	4
2.	Terminology	5
3.	Conference Event Package	6
3.1	Event Package Name	6
3.2	SUBSCRIBE Bodies	6
3.3	Subscription Duration	6
3.4	NOTIFY Bodies	7
3.5	Notifier Processing of SUBSCRIBE Requests	7
3.6	Notifier Generation of NOTIFY Requests	7
3.7	Subscriber Processing of NOTIFY Requests	8
3.8	Handling of Forked Requests	8
3.9	Rate of Notifications	8
3.10	State Agents	8
4.	Conference Data Format	10
4.1	Conference Information	10
4.1.1	User Element	10
4.1.1.1	User Attributes	11
4.1.1.2	User Status Elements	12
4.1.1.3	Media Information	13
4.1.1.3.1	Media Attributes	13
4.1.1.3.2	Media Elements	14
4.1.1.4	User Role	14
4.1.2	Sidebar Element	15
4.1.3	Additional Conference Identifiers	15
4.1.4	Policy URIs	15
4.1.5	Recording	15
4.1.6	Streaming	16
4.2	Constructing Coherent State	16
4.2.1	The Algorithm	17
4.3	Schema	18
4.4	Example	21
5.	Security Considerations	23
6.	IANA Considerations	24
6.1	conference Event Package Registration	24
6.2	application/conference-info+xml MIME Registration	24
6.3	URN Sub-Namespace Registration for urn:ietf:params:xml:ns:conference-info	24
6.4	XML Schema Registration	25

7.	Acknowledgements	26
8.	Changes History	27
8.1	Changes since -04	27

8.2	Changes since -03	27
8.3	Changes since -02	27
8.4	Changes since -01	28
9.	References	29
9.1	Normative References	29
9.2	Informative References	29
	Authors' Addresses	30
	Intellectual Property and Copyright Statements	32

1. Introduction

The Session Initiation Protocol (SIP) [6] Events framework Events Framework [7] defines general mechanisms for subscribing to, and receiving notifications of, events within SIP networks. It introduces the notion of a package, which is a specific "instantiation" of the events framework for a well-defined set of events. Here, we define an event package for SIP conferences. This package provides the conference notification service as outlined in the SIP conferencing framework [14]. As described there, subscriptions to a conference URI are routed to the focus that is handling the conference. It acts as the notifier, and provides clients with updates on conference state.

The information provided by this package is comprised of conference identifier(s), conference participants (optionally with their statuses and media description), conference sidebars, conference policy URIs, etc.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [2] and indicate requirement levels for compliant implementations.

3. Conference Event Package

The conference event package allows a user to subscribe to a conference. In SIP, conferences are represented by URIs. These URIs route to a SIP user agent, called a focus, that is responsible for ensuring that all users in the conference can communicate with each other, as described in Conferencing Framework [14]. The focus has sufficient information about the state of the conference to inform subscribers about it.

It is possible a participant in the conference may in fact be another focus. In order to provide a more complete participant list, the focus MAY subscribe to the conference package of the other focus to discover the participant list in the cascaded conference. This information can then be included in notifications by using of the "cascaded-focus" attribute as specified by this package.

This section provides the details for defining a SIP Events package, as specified by RFC 3265 [7].

3.1 Event Package Name

The name of this event package is "conference". This package name is carried in the Event and Allow-Events header, as defined in RFC 3265 [7].

3.2 SUBSCRIBE Bodies

A SUBSCRIBE for a conference package MAY contain a body. This body defines a filter to apply to the subscription. Filter documents are not specified in this document, and at the time of writing, are expected to be the subject of future standardization activity.

A SUBSCRIBE for a conference package MAY be sent without a body. This implies the default subscription filtering policy. The default policy is:

- o Notifications are generated every time there is any change in the state of the conference.
- o Notifications do not normally contain full state; rather, they only indicate the state that has changed. The exception is a

NOTIFY sent in response to a SUBSCRIBE. These NOTIFYs contain the full state of the information requested by the subscriber.

3.3 Subscription Duration

The default expiration time for a subscription to a conference is one hour. Once the conference ends, all subscriptions to that particular conference are terminated, with a reason of "noresource" [RFC 3265](#)

[7].

3.4 NOTIFY Bodies

As described in [RFC 3265](#) [7], the NOTIFY message will contain bodies that describe the state of the subscribed resource. This body is in a format listed in the Accept header field of the SUBSCRIBE, or a package-specific default if the Accept header field was omitted from the SUBSCRIBE.

In this event package, the body of the notification contains a conference information document. This document describes the state of a conference. All subscribers and notifiers MUST support the "application/conference-info+xml" data format described in [Section 4](#). The subscribe request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/conference-info+xml". If the header field is present, it MUST include "application/conference-info+xml", and MAY include any other types capable of representing dialog state.

Of course, the notifications generated by the server MUST be in one of the formats specified in the Accept header field in the SUBSCRIBE request.

3.5 Notifier Processing of SUBSCRIBE Requests

The conference information contains very sensitive information. Therefore, all subscriptions SHOULD be authenticated and then authorized before approval. Authorization policy is at the discretion of the administrator, as always. However, a few recommendations can be made.

It is RECOMMENDED that all users in the conference be allowed to subscribe to the conference.

3.6 Notifier Generation of NOTIFY Requests

Notifications SHOULD be generated for the conference whenever there is a change in the state in any of the information delivered to the subscriber.

The changes generally occur when a new participant joins (i.e. gets "connected" to) or a participant leaves (i.e. gets "disconnected" from) the conference.

Subject to a local focus policy, additional changes in participant's status, changes in its media types, and other optional media attributes MAY be reported by the focus.

Changes in sidebar rosters SHOULD be reported by the focus to their participants and MAY be reported to others, subject to local policy.

Changes in conference identifiers and policy URIs SHOULD be reported by the focus to the conference participants.

3.7 Subscriber Processing of NOTIFY Requests

The SIP Events framework expects packages to specify how a subscriber processes NOTIFY requests in any package specific ways, and in particular, how it uses the NOTIFY requests to construct a coherent view of the state of the subscribed resource.

Typically, the NOTIFY for the conference package will only contain information about those users whose state in the conference has changed. To construct a coherent view of the total state of all users, a subscriber to the conference package will need to combine NOTIFYS received over time.

Notifications within this package can convey partial information; that is, they can indicate information about a subset of the state associated with the subscription. This means that an explicit algorithm needs to be defined in order to construct coherent and consistent state. The details of this mechanism are specific to the particular document type. See [Section 4.2](#) for information on constructing coherent information from an application/conference-info+xml document.

3.8 Handling of Forked Requests

By their nature, the conferences supported by this package are centralized. Therefore, SUBSCRIBE requests for a conference should not generally fork. Users of this package MUST NOT install more than a single subscription as a result of a single SUBSCRIBE request.

3.9 Rate of Notifications

For reasons of congestion control, it is important that the rate of notifications not become excessive. As a result, it is RECOMMENDED that the server not generate notifications for a single subscriber at

a rate faster than once every 5 seconds.

3.10 State Agents

Conference state is ideally maintained in the element in which the conference resides. Therefore, the elements that maintain the conference are the ones best suited to handle subscriptions to it. Therefore, the usage of state agents is NOT RECOMMENDED for this

package.

4. Conference Data Format

Conference information is an XML document that MUST be well-formed and SHOULD be valid. Dialog information documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying dialog information documents and document fragments. The namespace URI for elements defined by this specification is a URN [3], using the namespace identifier 'ietf' defined by [4] and extended by [1]. This URN is:

```
urn:ietf:params:xml:ns:conference-info
```

A conference information document begins with the root element tag "conference-info".

4.1 Conference Information

Conference information begins with the top level element "conference-info". This element has three mandatory attributes:

- version: This mandatory attribute allows the recipient of conference information documents to properly order them. Versions start at 0 and increment by one for each new document sent to a subscriber. Versions are scoped within a subscription. Versions MUST be represented using a 32 bit integer.
- state: This mandatory attribute indicates whether the document contains the full conference information, or whether it contains only the information that has changed since the previous document (partial).
- entity: This mandatory attribute contains the conference URI that identifies the conference being described in the document.

The "conference-info" element has zero or more "user" sub-elements which contain information on the users in the conference. This is followed by zero or more "sidebar" sub-elements which contain information on the sidebars in the conference. This is followed by zero or more "conf-uri" sub-elements which contain information on additional URIs that the conference can be accessed by. This is followed by zero or more "policy-uri" sub-elements which contain information on additional URIs that the conference policies can be accessed by. This is followed by "recording" and "streaming" elements describing recording and streaming statuses of the

conference.

[4.1.1.1](#) User Element

Rosenberg, et al.

Expires January 16, 2005

[Page 10]

[4.1.1.1](#) User Attributes

The user element has one mandatory attribute, "uri" that indicates the URI for the user in the conference. This is a logical identifier, which corresponds to the authenticated identity of the participant. The "uri" attribute **MUST** be unique in the user element list because it is used as the key in partial notifications about users' state.

If a conference participant has more than a single signaling dialog associated with the conference, the conference focus **MAY** present the user's aggregated information (e.g. the statuses) and display all its media streams under a single user element.

Note, that the optional element "instance" of "media" (see below) **MAY** be used in this case to specify the actual signaling dialog for each media stream.

An anonymous participant in a conference **SHOULD** be represented by an anonymous URI generated by the focus. For multiple anonymous participants, the focus must ensure that each anonymous URI is unique. The guidelines for generating anonymous URIs in [RFC 3323](#) [8] should be followed. For example,

```
"Anonymous1" <sip:anonymous1@anonymous.invalid>
```

could be used for a participant requesting privacy.

The optional attribute "display-name" contains a display name for the user. The standard "xml:lang" language attribute can also be present to indicate the language of the display-name.

The optional attribute "cascaded-focus" contains a conference URI (different from the main conference URI) for users that are connected to the main conference as a result of focus cascading. In accordance with the SIP conferencing framework [14], this package allows for representation of peer-to-peer (i.e. "flat") focus cascading only. The actual cascading graph can not be deduced from the information provided in the package alone. Advanced applications can construct the graph by subscribing to both this package and the Dialog Package

[[15](#)] of the cascaded foci and correlating the relevant information.

If the main conference "state" is "full", the state of its user(s) MUST "full". If the main conference "state" is "partial", the state of its user(s) MAY be either "partial" or "full".

[4.1.1.2](#) User Status Elements

Three optional status elements are defined: status, joining-mode, and disconnection-reason.

- o "status": provides information about user's current level of participation in the conference.
- o "joining-mode": if present, provides information about the way the user joined the conference.
- o "disconnection-reason": if present, provides information about the way the user left the conference.

The following statuses are defined for the "status" element:

connected: The user is a participant in the conference. Depending on the media policies, he/she can send and receive media to and from other participants.

disconnected: The user is not a participant in the conference and no active dialog exists between the user and the focus.

on-hold: Active SIP dialog exists between a user and a focus, but user is "on-hold" for this conference, i.e. neither he/she is "hearing" the conference mix, nor is his/her media being mixed in the conference. As an example, the user has asked to join the conference using SIP, but his/her participation is pending based on moderator approval. In the meantime he/she is hearing music-on-hold or some other kind of related content.

muted-via-focus: Active SIP dialog exists between a user and a focus and the user can "listen" to the conference, but user's media is not being mixed into the conference. Note that sometimes a subset of user media streams can be muted by focus (such as poor quality video) while others (such as voice or IM) can still be active. In this case, it is RECOMMENDED that the "aggregated" user connectivity "status" reflects the status of the mostly active media.

blocked: User is denied from ever participating in this conference.

pending: User is not yet in the session, but it is anticipated that he/she will join in the near future.

calling: User is being called by the focus.

ringing: An PSTN ALERTING or SIP 180 Ringing was returned for the outbound call, user is being alerted.

dialing-in: User is dialing into the conference, not yet in the roster (probably being authenticated).

disconnecting: Focus is in the process of disconnecting user (either DISCONNECT or BYE was sent to the user's device).

removed: This status is used to remove the user from the roster using partial notifications mechanism.

Note that the defined transient states (e.g., calling, ringing, etc.) could generate a lot of notifications. Implementations MAY choose not to generate notifications on these to all participants if it will

generate too much traffic.

The following statuses are defined for the "joining-mode" element:

dialed-in: The user dialed into the conference, i.e. sent INVITE to the focus, which resulted in successful dialog establishment.

dialed-out: The focus has brought the user into the conference by sending a successful INVITE to the user.

focus-owner: The user is the focus for this conference. This status is used only when a participant UA acts as a conference focus.

The following statuses are defined for the disconnection-reason element:

departed: The user sent a BYE, thus leaving the conference.

booted: The user was sent a BYE by the focus, booting him/her out of the conference. Alternatively, the user tried to dial into to conference without success because was rejected by the focus according to local policy decisions.

failed: The server tried to bring the user into the conference, but its attempt to contact the specific user resulted in a non-200 class final response. Alternatively, the user tried to dial into the conference without success due to technical reasons.

4.1.1.3 Media Information

Each user has zero or more "media" sub-elements.

Each "media" element indicates the media that the user is currently connected to. Here, "connected to" implies that a user has a media line in his/her SDP [[12](#)] document(s). With this definition, a user is connected to a media stream even if he/she is not sending any media.

4.1.1.3.1 Media Attributes

The "media" element has a mandatory "media-type" attribute which identifies the media type (e.g. audio, video, message and application) and MUST have one of the values registered for "media" of SDP [[12](#)].

The optional "id" attribute serves as a unique reference to a "media"

element within the "user" element. It MUST be included for each "media" element for all notifications if the focus uses "partial" user notifications for this conference. Otherwise, the "id" attribute MAY be omitted.

If the user "state" is "full", the state of its "media" element(s) MUST be "full". If the user "state" is "partial", the state of its "media" element(s) MAY be either "partial" or "full".

[4.1.1.3.2](#) Media Elements

The "media" element has also an optional "proto" sub-element, which MUST have the value registered for "proto" of SDP [[12](#)].

An optional "ssrc" sub-element, if present, carries the value of SSRC (defined in RTP/RTCP [[10](#)]) as generated by the user for the stream it sends.

When an RTP mixer generates a CSRC list according to RTP/RTCP [[10](#)], it inserts a list of the SSRC identifiers of the sources that contributed to the generation of a particular packet into the RTP header of that packet. "An example application is audio conferencing where a mixer indicates all the talkers whose speech was combined to produce the outgoing packet, allowing the receiver to indicate the current talker, even though all the audio packets contain the same SSRC identifier (that of the mixer)."

An optional "info" sub-element, if present, carries a human readable description for this stream populated by the focus. The value of this element corresponds to the information media attribute "i" in SDP [[12](#)].

An optional "label" sub-element, if present, carries a unique identifier for this stream among all streams in the conference and is assigned by the focus. The value of this element corresponds to the "label" media attribute in SDP [[12](#)] and defined in [[18](#)].

An optional "instance" sub-element, if present, carries a URI, which MUST uniquely identify the signaling dialog being used for establishing of this media stream. In SIP, for example, values of Contact URI or GRUU [[17](#)] can be used for this purpose. It is RECOMMENDED to include the "instance" information for every user that has more than a single dialog associated with the conference. This element SHOULD NOT be included for an anonymous participant.

An optional "status" sub-element, if present, is used to remove "media" elements during partial notifications.

Optional "snd-status" and "rcv-status" sub-elements, if present,

describe the status of media streams in each direction.

4.1.1.4 User Role

The optional "role" element conveys the role of the user in the conference, e.g. participant, presenter, panelist, host, etc. User's role MAY change dynamically in the course of the conference. Also, a user MAY have more than a single role in one time.

This document does not define fixed values for the "role" element, instead it is expected that conferencing applications will define custom-fit roles by templates.

[4.1.2](#) Sidebar Element

The sidebar element is of the general "conference-type" and MAY use all the attributes and elements defined by it. Typically, only the "entity", which uniquely identifies the sidebar, and the "user" elements will be useful to present to the majority of the participants in the conference.

The "conference-type" mandatory attributes MUST be included for each sidebar.

The value of the "version" attribute is meaningless for "sidebar" elements and MUST be ignored because it is always overruled by the "version" attribute of the main "conference-info".

If the main conference "state" is "full", the state of its sidebar(s) MUST be "full". If the main conference "state" is "partial", the state of its sidebar(s) MAY be either "full" or "partial".

The "entity" URI attribute MUST be unique among the sidebar identifiers of the same conference. Attribute "entity" is used as the key for "sidebar" elements in partial notifications for "conference-info".

[4.1.3](#) Additional Conference Identifiers

In addition to the Conference URI present in the "entity" attribute, a conference MAY have additional URIs of various types. Connecting to these URIs will result in joining to the same conference.

[4.1.4](#) Policy URIs

A policy URI specifies where and how a certain policy pertaining to the conference can be accessed. The actual policy name and usage is deduced from the URI schema name.

An example for the "policy-uri" usage is inclusion of the URI of the CPCP [\[16\]](#). A subscriber to the Conference package can use the Policy URI to access and modify the conference policy.

[4.1.5](#) Recording

In many cases, legal regulations require conference providers to announce to the participants that a specific conference is being

recorded.

In addition to the recording "status" information, the "recording" element MAY include the URIs specifying the location and the format of the recorded data. Typically, the recorded data becomes available after the conference ends. Multiple URIs can be provided, for example, specifying different content types. For Web-Page embedded media, a plain HTTP URI MAY be provided.

[4.1.6](#) Streaming

The "streaming" element, if present, specifies whether the conference output is being streamed (to general public, for example), in what streaming format, and at what (e.g. multicast) addresses it can be listened at. RTSP [[11](#)] is an example of such streaming protocol.

[4.2](#) Constructing Coherent State

The conference information is described by a hierarchal XML structure with the root element "conference-info". The root element is the only element in the schema that carries meaningful version number for all the elements in the document. The whole conference information is associated with this version number.

The version number MUST be initialized with the value of the "version" attribute from the "conference-info" element in the first document received. Each time a new document is received, the value of the local version number, and the "version" attribute in the new document, are compared. If the value in the new document is one higher than the local version number, the local version number is increased by one, and the document is processed. If the value in the document is more than one higher than the local version number, the local version number is set to the value in the new document, the document is processed, and the subscriber SHOULD generate a refresh request to trigger a full state notification. If the value in the document is less than the local version, the document is discarded without processing.

Further processing of the conference information document depends on whether it contains full or partial state. If it contains full state, indicated by the value of the "state" attribute in the

"conference-info" element, the whole local content is flushed and repopulated from the document.

If the document contains partial state, as indicated by the value of the "state" attribute in the "conference-info" element, the document is used to update the local content as described below.

All sub-elements in the "conference-info" hierarchical XML structure can be classified in two groups: those that carry relatively small amount of data and those that can potentially carry a lot of data. During partial notifications, the light elements are updated as atomic pieces of data. On the other hand, elements that can carry a substantial amount of data have the general "state" attribute attached to them. That is in order to support partial notifications for their content.

A "state" attribute of a child element in the document MUST adhere to its parent "state". It means that if the parent's "state" is "full", the state of its children MUST be "full". If the parent's "state" is "partial", the state of its children MAY be either "partial" or "full".

For elements with optional "state" attribute, if the attribute is not included for an element, it means that the element's state is "full".

For a parent element with "state", its sub-elements with possible multiple appearances under the parent have keys that uniquely identify each element among others in the same list.

[4.2.1](#) The Algorithm

The conference package subscriber locally maintains a local element for each element in the schema and a table for each "element with key(s)" in the schema. The tables are indexed by the key(s) defined in schema for the element.

Starting from outer elements in the received document,

1. If the parent element contains full state, the element is replaced with the new information as a whole.

2. Otherwise, if the parent element contains partial state,

- 2.1 For elements with keys, the subscriber compares the keys received in the update with the keys in the local tables.

2.1.1 If a key doesn't exist in the local table, a row is added, and its content is set to the element information from the update.

2.1.2 Otherwise, if a key of the same value does exist, for each sub-element in the row the algorithm is applied from step 2.2.

2.2 For each atomic element received in the schema, the element is replaced with the new information as a whole. Also, for each non-atomic element received in the schema with either no "state"

attribute included or the state attribute is set to "full", the element is replaced with the new information as a whole.

2.2.1 If the updated or created element carries the "removed" status, that element SHOULD be removed from the local content. If the element is updated or created, such that it is empty, that element MAY be removed from the local content at any time.

2.3 For each non-atomic element with the state attribute set to "partial", the algorithm is applied recursively starting from step 2.

[4.3](#) Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:conference-info"
xmlns:tns="urn:ietf:params:xml:ns:conference-info" xmlns:xs="http://www.w3.org/
2001/XMLSchema" xmlns="urn:ietf:params:xml:ns:conference-info"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!--
    This import brings in the XML language attribute xml:lang
  -->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/03/xml.xsd" />

  <xs:element name="conference-info" type="tns:conference-type"/>

  <xs:simpleType name="state-type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="full" />
      <xs:enumeration value="partial" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="conference-type">

    <xs:sequence>
      <xs:element name="user" type="user-type" minOccurs="0"
maxOccurs="unbounded" />
```

```
<xs:element name="sidebar" type="conference-type" minOccurs="0"
maxOccurs="unbounded" />
<xs:element name="conf-ids" type="conf-ids-type" minOccurs="0"
maxOccurs="1" />
<xs:element name="policy-ids" type="policy-ids-type" minOccurs="0"
maxOccurs="1" />
<xs:element name="recording" type="recording-type" minOccurs="0"
maxOccurs="1" />
<xs:element name="streaming" type="streaming-type" minOccurs="0"
maxOccurs="1" />
<xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
```

```
<xs:attribute name="version" type="xs:nonNegativeInteger" use="required"/>
<xs:attribute name="state" type="tns:state-type" use="required"/>
<xs:attribute name="entity" type="xs:anyURI" use="required"/>
<xs:anyAttribute />
```

```
</xs:complexType>
```

```
<xs:complexType name="conf-ids-type">
  <xs:sequence>
    <xs:element name="conf-uri" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded" />
    <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
```

```
  <xs:anyAttribute />
</xs:complexType>
```

```
<xs:complexType name="policy-ids-type">
  <xs:sequence>
    <xs:element name="policy-uri" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded" />
    <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
```

```
  <xs:anyAttribute />
</xs:complexType>
```

```
<xs:complexType name="recording-type">
  <xs:sequence>
    <xs:element name="uri" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded" />
    <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
```

```
  <xs:attribute name="status" type="stream-status-type" use="required"/>
  <xs:anyAttribute />
</xs:complexType>
```

```
<xs:complexType name="streaming-type">
  <xs:sequence>
    <xs:element name="uri" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded" />
    <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
```

```
<xs:attribute name="status" type="stream-status-type" use="required"/>
<xs:anyAttribute />
</xs:complexType>
```

```
<xs:complexType name="user-type">
  <xs:sequence>
    <xs:element name="status" type="tns:user-status-type" minOccurs="0"/>
    <xs:element name="joining-mode" type="tns:user-joining-mode-type"
minOccurs="0"/>
    <xs:element name="disconnection-reason" type="tns:user-disconnection-
reason-type" minOccurs="0"/>
    <xs:element name="media" type="tns:media-type" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="role" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
```

```
<xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
```

```
</xs:sequence>
```

```
<xs:attribute name="uri" type="xs:anyURI" use="required"/>
<xs:attribute name="display-name" type="xs:string" use="optional"/>
<xs:attribute ref="xml:lang" use="optional"/>
<xs:attribute name="cascaded-focus" type="xs:anyURI" use="optional"/>
<xs:attribute name="state" type="tns:state-type" use="optional"/>
<xs:anyAttribute />
```

```
</xs:complexType>
```

```
<xs:simpleType name="user-status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="connected"/>
    <xs:enumeration value="disconnected"/>
    <xs:enumeration value="on-hold"/>
    <xs:enumeration value="muted-via-focus"/>
    <xs:enumeration value="blocked"/>
    <xs:enumeration value="pending"/>
    <xs:enumeration value="calling"/>
    <xs:enumeration value="ringing"/>
    <xs:enumeration value="dialing-in"/>
    <xs:enumeration value="disconnecting"/>
    <xs:enumeration value="removed"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="user-joining-mode-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="dialed-in" />
    <xs:enumeration value="dialed-out" />
    <xs:enumeration value="focus-owner" />
  </xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="user-disconnection-reason-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="departed" />
    <xs:enumeration value="booted" />
    <xs:enumeration value="failed" />
  </xs:restriction>
</xs:simpleType>
```

```
</xs:restriction>  
</xs:simpleType>
```

```
<xs:complexType name="media-type">  
  <xs:sequence>  
    <xs:element name="proto"      type="xs:string"          minOccurs="0"/>
```



```
<xs:element name="ssrc"          type="xs:nonNegativeInteger" minOccurs="0"/>
<xs:element name="info"          type="xs:string"              minOccurs="0"/>
<xs:element name="label"         type="xs:string"              minOccurs="0"/>
<xs:element name="instance"      type="xs:anyURI"              minOccurs="0"/>
<xs:element name="status"        type="tns:media-status-type" minOccurs="0"/>
<xs:element name="snd-status"    type="tns:stream-status-type" minOccurs="0"/
>
<xs:element name="rcv-status"    type="tns:stream-status-type" minOccurs="0"/
>
  <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>

<xs:attribute name="media"      type="xs:string"              use="required"/>
<xs:attribute name="id"         type="nonNegativeInteger"      use="optional"/>
<xs:attribute name="state"      type="tns:state-type"          use="optional"/>
<xs:anyAttribute />

</xs:complexType>

<xs:simpleType name="media-status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="removed" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="stream-status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="on"/>
    <xs:enumeration value="off"/>
    <xs:enumeration value="muted" />
  </xs:restriction>
</xs:simpleType>

</xs:schema>
```

[4.4](#) Example

The following is an example conference information document:

Rosenberg, et al.

Expires January 16, 2005

[Page 21]

```
<?xml version="1.0" encoding="utf-8" ?>
<conference-info version="0" state="full" entity="sip:conf233@example.com">
  <user uri="sip:bob@example.com" display-name="Bob Jones">
    <status>connected</status>
    <joining-mode>dialed-in</joining-mode>
    <media media="audio">
      <proto>RTP/AVP</proto>
      <ssrc>583398</ssrc>
    </media>
  </user>
  <user uri="sip:barbara@example.com" display-name="Barbara Jones">
    <status>on-hold</status>
  </user>
  <user uri="sip:bill@example.com" display-name="Bill Minelli">
    <status>on-hold</status>
  </user>

  <sidebar version="0" state="full" entity="sip:conf233.1@example.com">
    <user uri="sip:barbara@example.com" />
    <user uri="sip:bill@example.com" />
  </sidebar>

  <conf-ids>
    <conf-uri>tel:+18005671234</conf-uri>
    <conf-uri>h323:conf545@example.com</conf-uri>
  </conf-ids>

  <recording status="on">
    <uri>http://quicktime.streaming.com/54634/recording.mov</uri>
    <uri>http://real.streaming.com/54634/recording.ram</uri>
    <uri>http://windowsmedia.streaming.com/54634/recording.wmv</uri>
    <uri>http://www.streaming.com/54634/recording.html</uri>
  </recording>

</conference-info>
```

This conference currently has three users, two of which are in a sidebar conversation. The conference is being recorded. There are additional means to join the conference either by phone using tel URI [\[14\]](#) or by H.323 protocol using H.323 URL [\[13\]](#).

5. Security Considerations

Subscriptions to conference state can reveal very sensitive information. For this reason, the document recommends authentication and authorization, and provides guidelines on sensible authorization policies.

Since the data in notifications is sensitive as well, end-to-end SIP encryption mechanisms using S/MIME SHOULD be used to protect it.

Since a focus provides participants identity information using this event package, participant privacy needs to be taken into account. A focus MUST support requests by participants for privacy. Privacy can be indicated by the conference policy - for every participant or select participants. It can also be indicated in the session signaling. In SIP this can be done using the Privacy header field described in [RFC 3323](#) [8]. For a participant requesting privacy, no identity information SHOULD be revealed by the focus such as a URI (e.g. the Address of Record, Contact, or GRUU). For these cases, the anonymous URI generation method outlined in section "User Element" of this document MUST be followed.

6. IANA Considerations

This document registers a SIP event package, a new MIME type, application/conference-info+xml, a new XML namespace, and a new XML schema.

6.1 conference Event Package Registration

This specification registers an event package, based on the registration procedures defined in [RFC 3265](#) [7]. The following is the information required for such a registration:

Package Name: conference

Package or Template-Package: This is a package.

Published Document: RFC XXXX (Note to RFC Editor: Please fill in XXXX with the RFC number of this specification).

Person to Contact: Jonathan Rosenberg, jdrosen@jdrosen.net.

6.2 application/conference-info+xml MIME Registration

MIME media type name: application

MIME subtype name: conference-info+xml

Mandatory parameters: none

Optional parameters: Same as charset parameter application/xml as specified in [RFC 3023](#) [5].

Encoding considerations: Same as encoding considerations of application/xml as specified in [RFC 3023](#) [5].

Security considerations: See [Section 10 of RFC 3023](#) [5] and [Section 5](#) of this specification.

Interoperability considerations: none.

Published specification: This document.

Applications which use this media type: This document type has been used to support SIP conferencing applications.

Additional Information:

 Magic Number: None

 File Extension: .cif or .xml

 Macintosh file type code: "TEXT"

Personal and email address for further information: Jonathan Rosenberg, <jdrosen@jdrosen.net>

Intended usage: COMMON

Author/Change controller: The IETF.

6.3 URN Sub-Namespace Registration for urn:ietf:params:xml:ns:conference-info

This section registers a new XML namespace, as per the guidelines in [\[1\]](#).

URI: The URI for this namespace is
urn:ietf:params:xml:ns:conference-info.
Registrant Contact: IETF, SIPING working group, <sipping@ietf.org>,
Jonathan Rosenberg <jdrosen@jdrosen.net>.
XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml1-basic/xhtml1-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>Conference Information Namespace</title>
</head>
<body>
  <h1>Namespace for Conference Information</h1>
  <h2>urn:ietf:params:xml:ns:conference-info</h2>
  <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
</body>
</html>
END
```

[6.4](#) XML Schema Registration

This specification registers a schema, as per the guidelines in in
[\[1\]](#).

URI: please assign.
Registrant Contact: IETF, SIPING Working Group
(sipping@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).
XML: The XML can be found as the sole content of [Section 4.3](#).

7. Acknowledgements

The authors would like to thank Dan Petrie, Sean Olson, Alan Johnston, and Rohan Mahy for their comments and inputs.

8. Changes History

8.1 Changes since -04

- o**
- o "Sidebar-type" has been removed. "Sidebar" conference element is defined using the general "conference-type".
- o "Recording" conference attribute has been replaced with "recording" and "streaming" elements within "conference-type". New "recording-type" and "streaming-type" have been introduced.
- o Attribute "state" has been added to "user-type".
- o Element "media-stream" within "user-type" has been renamed to "media".
- o Element "role" within "user-type" has been introduced.
- o The following statuses have been added to "user-status-type": blocked, pending, calling, ringing, dialing-in, disconnecting, removed.
- o User status "muted-by-focus" has been renamed to "muted-via-focus".
- o Attributes "id" and "state" have been added to "media-type".
- o Elements "status", "snd-status" and "rcv-status" have been added to "media-type".
- o Element "dialog-id" has been renamed to "instance".
- o "Constructing Coherent State" section has been updated to include user and media partial notifications.

8.2 Changes since -03

- o "Constructing Coherent State" section has been updated.**
- o In order to support partial notifications, two placeholders "conference-ids" and "policy-ids" (for "conf-uri" and "policy-uri" elements, correspondingly) are created.
- o Discussion and security considerations regarding anonymous participation have been added.
- o Optional elements "dialog-uri", "info" and "label" per media stream are added.

8.3 Changes since -02

- o State "muted-by-focus" is added to user's status.**
- o Optional conference attribute "recording" is added.
- o Policy URI placeholder (i.e. element "policy-uri") is created.
- o Example's syntax is corrected.
- o Optional attribute "cascaded-focus" URI per user is added.
- o Optional additional conference identifiers (i.e. element "conf-uri") are added.
- o In order to cover all possible cases, participant's status is expressed using three optional statuses: "status", "joining-mode"

and "disconnection-reason". That is instead of "activity-status",
"history-status" and "is-on-dial-out-list".

8.4 Changes since -01

- o Package parameters are removed. Decision about performing**
"recursive" membership algorithm is perceived as a focus local policy.
- o General information (i.e. pointers to additional available services) is removed. The defined XML schema can be extended in future to include those when XCON work matures.
- o Dialog information is removed. It can be obtained by direct subscription to a dialog package of a participant.
- o Media stream information is aligned with SDP definitions (media and proto) and SSRC attribute is added.
- o Participant's status is expressed using two optional statuses: "activity" and "history". Optional "is-on-a-dial-out-list" indication is added.
- o Normative references to XCON work are removed.
- o Optional sidebar rosters are added.

9. References

9.1 Normative References

- [1] Mealling, M., "The IETF XML Registry", [draft-mealling-iana-xmlns-registry-05](#) (work in progress), June 2003.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [3] Moats, R., "URN Syntax", [RFC 2141](#), May 1997.
- [4] Moats, R., "A URN Namespace for IETF Documents", [RFC 2648](#), August 1999.
- [5] Murata, M., St. Laurent, S. and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [6] 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.
- [7] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [8] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", [RFC 3323](#), November 2002.
- [9] Camarillo, G., Eriksson, G., Holler, J. and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", [RFC 3388](#), December 2002.
- [10] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.

9.2 Informative References

- [11] Schulzrinne, H., Rao, A. and R. Lanphier, "Real Time Streaming Protocol (RTSP)", [RFC 2326](#), April 1998.

- [12] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", [RFC 2327](#), April 1998.

- [13] Levin, O., "H.323 Uniform Resource Locator (URL) Scheme Registration", [RFC 3508](#), April 2003.

- [14] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol",
[draft-ietf-sipping-conferencing-framework-02](#) (work in progress), June 2004.
- [15] Rosenberg, J. and H. Schulzrinne, "An INVITE Initiated Dialog Event Package for the Session Initiation Protocol (SIP)",
[draft-ietf-sipping-dialog-package-04](#) (work in progress),
February 2004.
- [16] Koskelainen, P. and H. Khartabil, "Requirements for Conference Policy Control Protocol", [draft-ietf-xcon-cpcp-reqs-03](#) (work in progress), April 2004.
- [17] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", [draft-ietf-sip-gruu-02](#) (work in progress), July 2004.
- [18] Levin, O. and G. Camarillo, "The SDP (Session Description Protocol) Label Attribute",
[draft-levin-mmusic-sdp-media-label-00](#) (work in progress), July 2004.

Authors' Addresses

Jonathan Rosenberg
dynamicsoft
600 Lanidex Plaza
Parsippany, NJ 07054
US

Phone: +1 973 952-5000
EMail: jdrosen@dynamicsoft.com
URI: <http://www.jdrosen.net>

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.

New York, NY 10027
US

E-Mail: schulzrinne@cs.columbia.edu
URI: <http://www.cs.columbia.edu/~hgs>

Internet-Draft

Conference Package

July 2004

Orit Levin (editor)
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA

EMail: oritl@microsoft.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

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

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

Rosenberg, et al.

Expires January 16, 2005

[Page 32]