

SIPPING
Internet-Draft
Expires: December 29, 2003

J. Rosenberg
dynamicsoft
H. Schulzrinne
Columbia University
June 30, 2003

**A Session Initiation Protocol (SIP) Event Package for Conference
State
draft-ietf-sipping-conference-package-01**

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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 December 29, 2003.

Copyright Notice

Copyright (C) The Internet Society (2003). 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, the state of the dialogs that compose the conference, and general information about the conference.

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Conference Event Package	5
3.1	Event Package Name	5
3.2	Event Package Parameters	5
3.3	SUBSCRIBE Bodies	6
3.4	Subscription Duration	6
3.5	NOTIFY Bodies	6
3.6	Notifier Processing of SUBSCRIBE Requests	7
3.7	Notifier Generation of NOTIFY Requests	7
3.8	Subscriber Processing of NOTIFY Requests	7
3.9	Handling of Forked Requests	8
3.10	Rate of Notifications	8
3.11	State Agents	8
4.	Conference Data Format	9
4.1	Structure of the Format	9
4.1.1	Conferencing Service Element	9
4.1.2	User Element	10
4.1.3	Status	10
4.1.4	Dialog	11
4.1.5	Media Streams	11
4.2	Constructing Coherent State	12
4.3	Schema	13
4.4	Example	15
5.	Security Considerations	16
6.	IANA Considerations	17
6.1	conference Event Package Registration	17
6.2	application/conference-info+xml MIME Registration	17
6.3	URN Sub-Namespace Registration for urn:ietf:params:xml:ns:conference-info	18
6.4	XML Schema Registration	19
7.	Acknowledgements	20
	Normative References	21
	Informative References	22
	Authors' Addresses	23
	Intellectual Property and Copyright Statements	24

1. Introduction

The Session Initiation Protocol (SIP) [3] Events framework [2] 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. Packages have been defined for user presence [11], watcher information [12], and message waiting indicators [14], amongst others. Here, we define an event package for SIP conferences. This package provides the conference notification service as outlined in the SIP conferencing framework [15]. 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 broken into several general categories:

General State: A small amount of general conference state is provided, primarily for the purposes of bootstrapping access to other conference services, such as media and conference policy controls.

Membership State: The members of the conference, and their state within the conference.

Dialog State: The state of the dialogs for users connected to the conference.

Media State: Information about what media users in the conference are receiving.

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](#) [[1](#)] 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 [15]. The focus has sufficient information about the state of the conference to inform subscribers about it.

This section provides the details for defining a SIP Events package, as specified by [2].

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 [2].

3.2 Event Package Parameters

Two Event header field parameters are defined for this package. The first is "recurse". The parameter has no values. When present, it informs the server that it should check whether any participants in the conference are themselves a focus, and if so, generate a subscription to their conference state with the "recurse" attribute. The focus can determine whether a participant is a focus based on the presence of the "isfocus" capability indication in the Contact header field provided by the participant [4]. The users reported in notifications from this recursed subscription are reported to the original subscriber. The result is that the list of users reported to the subscriber represents the complete user list even when cascaded conferences are used.

The second Event header field parameter is "type". Its value is a comma separated list of the types of conference information that are desired. This specification defines four values - "general", "membership", "dialog", and "basic-media". When any one of these is present, it means that the corresponding conference information is desired. The specific components of a conference information document corresponding to each of these values are described below. Additional values for this parameter MAY be defined by extensions to this package. Such extensions are anticipated to handle membership and media policy state.

Both of these parameters MUST have the same value in a initial SUBSCRIBE request and any refreshes of the resulting subscription. In other words, their values are fixed for the duration of the subscription. The default meaning of the "type" parameter, when not present, is that the focus should send all information it has about

the conference. When present, the type parameter MUST have at least one value.

The BNF for these parameters is:

```
recurse      = "recurse"
type         = "type" EQUAL SWS DQUOTE conf-info *(", " conf-info)
              DQUOTE
              ;; EQUAL, SWS, DQUOTE from RFC3261
conf-info    = "general" | "membership" | "dialog" | "basic-media" |
              token
```

Example:

```
Event: conference;recurse;type="membership,general"
```

[3.3](#) SUBSCRIBE Bodies

A SUBSCRIBE for a dialog 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 dialog 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, where that change is in a piece of information that the subscriber has indicated (using the "type" Event header field parameter) an interest in receiving.
- 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.

[3.4](#) 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" [[2](#)].

[3.5](#) NOTIFY Bodies

As described in [RFC 3265](#) [2], 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.

[3.6](#) 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.

[3.7](#) 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 types requested by the subscriber. For "membership", these changes generally occur when a new participant joins, a participant leaves, and a dial-out attempt succeeds or fails. For "dialog", these changes occur when a dialog is created, terminated, or updated through a target refresh request. For "basic-media", these changes occur when the media types receive by a participant change. For "general", these changes occur when the conference policy protocol URIs change.

[3.8](#) 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 dialog 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.

[3.9](#) 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.

[3.10](#) 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.

[3.11](#) 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 [5] 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 [6], using the namespace identifier 'ietf' defined by [7] and extended by [8]. This URN is:

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

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

4.1 Structure of the Format

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

version: This 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 representable using a 32 bit integer.

state: This 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 attribute contains the conference URI that identifies the conference being described in the document.

The "conference-info" element has zero or more "conf-service" elements, which provide URIs for access conferencing services, such as media policy and conference policy control. This is followed by zero or more "user" sub-elements which contain information on the users in the conference.

4.1.1 Conferencing Service Element

The "conf-service" element contains a URI that can be used to access some additional conferencing service. The element contains the following attributes:

id: This attribute provides a unique identifier (unique within the context of the subscription) for the service. The attribute is mandatory.

type: This attribute contains a string which indicates the type of service. Defined values are "conf-policy", to indicate that the URI is for manipulating the conference policy [17][18], "media-policy", to indicate that the URI is for manipulating the media policy [9], and "floor-control", to indicate that the URI is for access to floor control services [19]. The attribute is mandatory.

There MUST only be one conf-service for each type of service. Additional service types may be defined in the future.

OPEN ISSUE: Once a protocol becomes solidified, we will need to add some additional content here. For example, if XCAP [16] is used, the AUID should be provided here. We may want to move to a model where each service type is a unique XML element; that would allow for the schema to impose the constraint on a single URI for each service type.

The "conf-service" element is only provided in notifications if the subscriber included the value of "general" in its "type" Event header field parameter.

4.1.2 User Element

Each user element has zero or one "status" elements, indicating their status in the conference, zero or one "dialog" elements, indicating their dialog information, and zero or one "media-streams" elements, indicating their media reception information.

The user element has one mandatory attribute, "uri" that indicates the URI for the user in the conference. This is a logical identifier, not a machine specific one (i.e., its taken from the authenticated identity of the participant). 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.

4.1.3 Status

The status element contains the status of the user in the conference. The following statuses are defined:

active: The user is in an active dialog with the focus.

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

booted: The user was sent a BYE by the conference host, booting them out of the conference.

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.

The "status" element is only provided in notifications if the subscriber included the value of "membership" in its "type" Event header field parameter.

[4.1.4](#) Dialog

The dialog element is defined in [\[13\]](#). It is presented from the viewpoint of the focus. The "dialog" element is only provided in notifications if the subscriber included the value of "dialog" in its "type" Event header field parameter.

[4.1.5](#) Media Streams

The "media-streams" element indicates the media streams that the user is currently connected to. Here, "connected to" implies that a user has a media line in their SDP [\[20\]](#) which is associated with a media stream managed by the focus (see Section 2.1 of [\[9\]](#)). With this definition, a user is connected to a media stream even if they are not sending any media.

The "media-streams" element has zero or more "media-stream" sub-elements. The value of the "media-stream" element is an identifier, unique within the conference, which identifies the media stream that a user is connected to [\[9\]](#). The "media-stream" element also has a mandatory "media-type" attribute which identifies the media type (audio, video, message and application) of the media stream.

The "media-streams" element is only provided in notifications if the subscriber included the value of "basic-media" in its "type" Event header field parameter.

OPEN ISSUE: This needs to be aligned with the final media policy mechanisms done in xcon. If we wish this draft to proceed independently, we should probably remove any notion of media information.

4.2 Constructing Coherent State

The conference information subscriber maintains a table for the list of users in the conference. The table contains a row for each user. Each row is indexed by a URI, present in the "uri" attribute of the "user" element. The contents of each row contain the state of that user as conveyed in the document. The subscriber also maintains a table for the service URIs in the conference. This table contains a row for each service type. Each row is indexed by a token, present in the "id" attribute of the "conf-service" element. The contents of the row contain the URI and type of that service.

Both tables are associated with a 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.

The 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 contents of both tables are flushed. They are repopulated from the document. A new row in the user table is created for each "user" element, and a new row in the conferencing services table is created for each "conf-service" element. 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 tables. For each "user" element in the document, the subscriber checks to see whether a row exists for that user in the user table. This check is done by comparing the URI in the "uri" attribute of the "user" element with the URI associated with the row. If the user doesn't exist in the table, a row is added, and its state is set to the information from that "user" element. If the user does exist, its state is updated to be the information from that "user" element. If a row is updated or created, such that its state is now booted, failed or departed, that entry MAY be removed from the table at any time.

Similarly, for each each "conf-service" element in the document, the

subscriber checks to see whether a row exists for that service in the service table. This check is done by comparing the id in the "id" attribute of the "conf-service" element with the id associated with the row. If the service doesn't exist in the table, a row is added, and its URI and type are set to be the information from the "conf-service" element. Since it is not expected that this information will change during the lifetime of the conference, there is no way to indicate removal of a service.

OPEN ISSUE: Is this really the right way to bootstrap these URIs? The information really is static, and placing it in an event package will result in a waste of bandwidth during full-state updates.

[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:di="urn:ietf:params:xml:ns:dialog-info"
  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"/>
  <!-- This import brings in the dialog-info element dialog-->
  <xs:import namespace="urn:ietf:params:xml:ns:dialog-info"
    schemaLocation="dialog-info.xsd"/>
  <xs:element name="conference-info">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="conf-service" type="tns:conf-serviceType"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="user" type="user-type"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="xs:nonNegativeInteger"
        use="required"/>
      <xs:attribute name="state" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="full"/>

```



```
        <xs:enumeration value="partial"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="entity" type="xs:anyURI" use="required"/>
</xs:complexType>
</xs:element>
<xs:complexType name="user-type">
  <xs:sequence>
    <xs:element name="status" type="tns:status-type" minOccurs="0"/>
    <xs:element ref="di:dialog" minOccurs="0"/>
    <xs:element name="media-streams" minOccurs="0">
      <xs:complexType name="media-status-type">
        <xs:sequence>
          <xs:element name="media-stream"
            type="tns:media-stream-type"
            minOccurs="0" maxOccurs="unbounded"/>
          <xs:any namespace="##other" processContents="lax"
            minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </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:complexType>
<xs:complexType name="media-stream-type">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="media-type" type="tns:mimetypes"
        use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="mimetypes">
  <xs:restriction base="xs:string">
    <xs:enumeration value="audio"/>
    <xs:enumeration value="video"/>
    <xs:enumeration value="message"/>
    <xs:enumeration value="application"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="conf-serviceType">
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attribute name="id" type="xs:string" use="required"/>
      <xs:attribute name="type" type="tns:typeType" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```



```
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="typeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="conf-policy"/>
    <xs:enumeration value="media-policy"/>
    <xs:enumeration value="floor-control"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="active"/>
    <xs:enumeration value="departed"/>
    <xs:enumeration value="booted"/>
    <xs:enumeration value="failed"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

4.4 Example

The following is an example conference information document:

```
<conference-info version="0" state="full" entity="sip:conf233@example.com">
  <user uri="sip:jdrosen@example.com" display-name="Jonathan Rosenberg">
    <status>active</status>
    <media-streams>
      <media-stream media-type="audio">8hha7</media-stream>
    </media-streams>
  </user>
  <user uri="sip:hgs@example.com" display-name="Henning Schulzrinne">
    <status>active</status>
  </user>
</conference-info>
```

This document describes a conference with two users, both of which are active.

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.

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](#) [2]. 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](#) [10].

Encoding considerations: Same as encoding considerations of application/xml as specified in [RFC 3023](#) [10].

Security considerations: See [Section 10 of RFC 3023](#) [10] 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 [8].

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>Dialog Information Namespace</title>
</head>
<body>
  <h1>Namespace for Dialog 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 [\[8\]](#).

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 and Sean Olson for their comments.

Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [3] 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.
- [4] Rosenberg, J., "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [draft-ietf-sip-callee-caps-00](#) (work in progress), June 2003.
- [5] Bray, T., Paoli, J., Sperberg-McQueen, C. and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C REC REC-xml-20001006, October 2000.
- [6] Moats, R., "URN Syntax", [RFC 2141](#), May 1997.
- [7] Moats, R., "A URN Namespace for IETF Documents", [RFC 2648](#), August 1999.
- [8] Mealling, M., "The IETF XML Registry", [draft-mealling-iana-xmlns-registry-05](#) (work in progress), June 2003.
- [9] Mahy, R. and N. Ismail, "Media Policy Manipulation in the Conference Policy Control Protocol", [draft-mahy-xcon-media-policy-control-00](#) (work in progress), June 2003.
- [10] Murata, M., St. Laurent, S. and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.

Informative References

- [11] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", [draft-ietf-simple-presence-10](#) (work in progress), January 2003.
- [12] Rosenberg, J., "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)", [draft-ietf-simple-wininfo-package-05](#) (work in progress), January 2003.
- [13] Rosenberg, J. and H. Schulzrinne, "An INVITE Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", [draft-ietf-sipping-dialog-package-01](#) (work in progress), March 2003.
- [14] Mahy, R., "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", [draft-ietf-sipping-mwi-03](#) (work in progress), July 2003.
- [15] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol", [draft-ietf-sipping-conferencing-framework-00](#) (work in progress), May 2003.
- [16] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", [draft-ietf-simple-xcap-00](#) (work in progress), June 2003.
- [17] Koskelainen, P. and H. Khartabil, "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Conference Policy Manipulation", [draft-koskelainen-xcon-xcap-cpcp-usage-00](#) (work in progress), June 2003.
- [18] Levin, O., "Conference Policy Control Protocol for Centralized Conferencing", [draft-levin-xcon-cpcp-00](#) (work in progress), June 2003.
- [19] Wu, X., "Use of Session Initiation Protocol (SIP) and Simple Object Access Protocol (SOAP) for Conference Floor Control Protocol (SOAP) for Conference Floor Control", [draft-wu-sipping-floor-control-04](#) (work in progress), March 2003.
- [20] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", [RFC 2327](#), April 1998.

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

EMail: schulzrinne@cs.columbia.edu
URI: <http://www.cs.columbia.edu/~hgs>

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in [BCP-11](#). Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

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