

MMUSIC Working Group
Internet-Draft
Expires: June 30, 2006

Y. Nomura
Fujitsu Labs.
H. Asaeda
Keio University
H. Schulzrinne
Columbia University
December 26, 2005

**SIP Event Notification for Internet Media Guides
draft-nomura-mmusic-img-notify-02**

Status of this Memo

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 becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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

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

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

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

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document specifies an update notification protocol for Internet Media Guides (IMGs) using SIP event notification. The mechanism achieves the timely delivery of IMGs and avoids that IMG receivers have to poll for updates.

Table of Contents

1	Introduction	3
2	Terminology	3
3	Overview of Protocol Operations	4
3.1	Mapping of IMG Operations to SIP Event Notification .	5
3.2	SIP Event Notification	5
3.3	IMG to SIP URI Mapping	6
4	Protocol Details	6
4.1	Initialize	6
4.2	Update Notification	7
4.3	Session Keep Alive	8
4.4	Polling metadata status	8
4.5	Confirming the status of IMG receiver	8
4.6	Timer Expiration	8
4.7	Invalid update notification	9
5	IMG Envelope	9
6	Example Message Flow	9
6.1	Example MIME Header	9
6.2	Example Flows	10
7	Security Considerations	13
8	IANA Considerations	13
9	Normative References	13
10	Informative References	13
11	Authors' Addresses	14

1 Introduction

Internet Media Guides (IMGs) [2] [3] provide and deliver structured collections of multimedia descriptions expressed using SDP [4], SDPng [5] or other description formats. They are used to describe sets of multimedia services (e.g. television program schedules, content delivery schedules) and refer to other networked resources including web pages.

IMG metadata may be delivered to a potentially large audience, who use it to join a subset of the sessions described, and who may need to be notified of changes to the IMG metadata. Since content and its structure described by IMG metadata changes as time elapses, an IMG receiver needs to be notified of changes so that IMG metadata and content do not become stale.

This document defines an update notification protocol for IMGs using SIP Event Notification framework [6], which satisfies the IMG framework [2] and matches the IMG requirements [3].

The authors assume that SIP event is not a mechanism just for a presence service, but applicable for many services such as IMGs. As the IMG framework defines the IMG operations, an update notification mechanism is necessary for scalable delivery. In addition, SIP event satisfies IMG requirements with minimum implementation cost for an unicast transport protocol, thus it provides necessary and sufficient mechanism for the IMG update notification.

2 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

Internet Media Guide (IMG): IMG is a generic term to describe the formation, delivery and use of IMG metadata. The definition of the IMG is intentionally left imprecise [3].

IMG Element: The smallest atomic element of metadata that can be transmitted separately by IMG operations and referenced individually from other IMG elements [3].

IMG Metadata: A set of metadata consisting of one or more IMG elements. IMG metadata describes the features of multimedia content used to enable selection of and access to media sessions containing content. For example, metadata may consist of the URI, title, airtime, bandwidth needed, file size, text summary, genre and access restrictions [3].

IMG Delivery: The process of exchanging IMG metadata both in terms of large scale and atomic data transfers [3].

IMG Transport Session: An association between an IMG sender and one or more IMG receivers within the scope of an IMG transport protocol. An IMG transport session involves a time bound series of IMG transport protocol interactions that provide delivery of IMG metadata from the IMG sender to the IMG receiver(s) [3].

IMG Sender: An IMG sender is a logical entity that sends IMG metadata to one or more IMG receivers [3].

IMG Receiver: An IMG receiver is a logical entity that receives IMG metadata from an IMG sender [3].

3 Overview of Protocol Operations

The IMG framework [2] defines two abstract operations, the IMG SUBSCRIBE and IMG NOTIFY, to notify updates of IMG metadata from an IMG sender to an IMG receiver. Thus, the IMG sender has an logical entity, IMG notifier, and the IMG receiver has a IMG subscriber. Both the IMG sender and IMG receiver need a bi-directional transport to fulfill the operation.

The IMG subscriber initiates an IMG transport session to an IMG notifier by sending an IMG SUBSCRIBE message. When IMG notifier receiving the IMG SUBSCRIBE message, it will reply an IMG NOTIFY for the IMG SUBSCRIBE. The IMG notifier sends the IMG NOTIFY message when IMG metadata is stale and should be updated in the IMG subscriber.

Figure 1 shows the protocol operations between the IMG subscriber and IMG notifier.

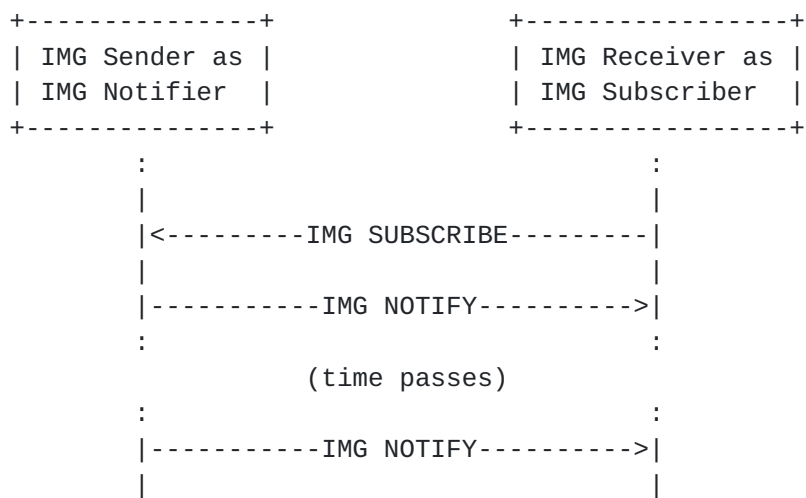


Figure 1: Update Notification Sequence Example

3.1 Mapping of IMG Operations to SIP Event Notification

There are simple mapping rules between IMG operations and SIP Event Notification mechanisms. A SIP SUBSCRIBE is an instance of the IMG SUBSCRIBE and a SIP NOTIFY is an instance of the IMG NOTIFY.

3.2 SIP Event Notification

SIP [7] is a text-based request/response protocol. A SIP request consists of a request-line, header field and message body like HTTP. A SIP response also consists of a status-line, header field and body.

SIP Event has the same feature as SIP but SIP Event needs only two messages, SUBSCRIBE and NOTIFY, to notify an event to subscribers. Since the protocol intends to do this, it does not require SIP messages to establish a session.

Fig. 2 shows basic protocol operations in SIP Event. Protocol details are shown in [section 4](#).

IMG Subscriber	IMG Notifier
-----SUBSCRIBE----->	Request state subscription
<-----200-----	Acknowledge subscription
<-----NOTIFY-----	Return current state information
-----200----->	
<-----NOTIFY-----	Return current state information
-----200----->	

Fig. 2 SIP Event Notification Sequence

3.3 IMG to SIP URI Mapping

An IMG envelope may include an IMG URN [8] which can be used to initiate an IMG transport session for the update notification. If the IMG URN can be used to the SIP URI for the update notification, the IMG receiver may try to resolve the IMG URN to the SIP URI. This mechanism will define in another document.

In other case, the IMG subscriber may obtain the SIP URI by the IMG notifier given some a priori knowledge.

This IMG URN may also indicate the URI for a original IMG metadata which provides a self-contained set of metadata for one media object or service, i.e., it does not need additional information from any other IMG metadata.

In general, the SIP URI has the following format.

```
sip:initial-version@host
```

"initial-version" indicates the original IMG metadata which is used to an initial version of IMG metadata in the succeeding update notification session. The IMG notifier provides this parameter to identify which version of IMG metadata the IMG subscriber has.

"host" means the location of the IMG notifier.

4 Protocol Details

4.1 Initialize

An IMG receiver MUST send a SUBSCRIBE request to an IMG sender in order to prepare to receive a subsequent update notification from the IMG sender.

The SUBSCRIBE request includes identity information in its headers defined by SIP Events framework. As defined in SIP, To, From, Call-ID, Event and Contact header can be used to identify a session between an IMG receiver and sender. Each parameter in the headers depends on the implementations.

In this phase, the SUBSCRIBE request MAY contain a body indicating what metadata status will be subscribed.

The name of this package is "img". This header also appears in NOTIFY requests.

If the SUBSCRIBE request is not related to existing sessions and the IMG sender can authenticate the request successfully, the IMG sender sends a 200 (OK) response to the IMG receiver. If the IMG sender can't authenticate or accept the request, the IMG sender sends a 4xx response and does not send a NOTIFY request.

An IMG sender MUST authenticate all subscription requests. This authentication can be done using any of the mechanisms defined in SIP and SIP Events.

After sending the SUBSCRIBE response, the IMG sender sends a NOTIFY request immediately to the same IMG receiver. The request contains an URI indicating a metadata location or metadata. Metadata indicated by URI or contained in the body MUST describe the latest full state when the NOTIFY request is sent. "Content-Type:" header in the NOTIFY request MUST indicate a data type of the body.

The body in NOTIFY request MUST contains a version number and a timestamp. The version number increases by exactly one for each NOTIFY message as defined in SIP Event. The time stamp indicates latest modified time of metadata status.

When the IMG receiver receives the request, the IMG receiver tries to retrieve IMG metadata specified in the body. If the IMG receiver gets it successfully, it sends 200 (OK) response to the IMG sender. If not, the IMG receiver sends 4xx response.

[4.2 Update Notification](#)

When metadata changes and it affects an existing subscription, IMG sender sends a NOTIFY request on a session related to the metadata status. The request body SHOULD contain metadata delta location or metadata delta. The format of metadata delta is discussed in [section 6](#).

When the IMG receiver receives the request, the IMG receiver tries to

obtain delta information specified in the body. If the IMG receiver successfully gets it and updates metadata, it sends 200 (OK) response to the IMG sender. If not, the IMG receiver sends 4xx response.

4.3 Session Keep Alive

At any time before a subscription expires, the IMG receiver may send a SUBSCRIBE request to the IMG sender. The IMG sender sends a SUBSCRIBE response and a NOTIFY request with delta information to communicate up-to-date metadata status.

If the IMG receiver receives the NOTIFY message which includes the same timestamp as the previous NOTIFY message, the IMG receiver does not have to obtain delta information. In this case, an IMG sender sends it just for a confirmation of the metadata status as an immediate response for a SUBSCRIBE request.

If the timestamp is updated, the IMG receiver MUST obtain new delta in the same way as receiving an update notification.

This mechanism can be applicable not only to refresh the timer but also to confirm the current metadata status.

4.4 Polling metadata status

If an IMG receiver does not want to receive an update notification, an IMG receiver may poll metadata status to send a SUBSCRIBE with an "Expires" of 0.

4.5 Confirming the status of IMG receiver

If an IMG sender needs to confirm the current status of an IMG receiver which subscribes IMG metadata status, it may send the NOTIFY request including a body which indicates a current delta information and wait a NOTIFY response from the IMG receiver. When the IMG sender receives the response with 200 (OK), the IMG sender confirms that the IMG receiver's status is up-to-date. If not, the IMG sender can decide that the IMG receiver has stale metadata.

4.6 Timer Expiration

Expiration time depends on the system. If the system requires short duration to guarantee metadata coherency, it should be a small value. In some system, a connection between an IMG receiver and IMG sender is not persistent but sporadic. In this case, IMG receiver may require the long expiration time such as 1 day or 1 week.

If an IMG receiver receives NOTIFY request on the session which

already has been terminated because the timer has expired, the IMG receiver SHOULD try to subscribe it again.

4.7 Invalid update notification

If an IMG receiver receives invalid NOTIFY message such as a discontinuous version number or CSeq, the IMG receiver SHOULD close the session and restart the session from Initialize step. As a result, the IMG receiver can obtain latest full metadata states.

5 IMG Envelope

The IMG envelope would reference or carry some application-specific metadata, and the envelope would support the maintenance of the application-specific metadata, which may also serve the metadata relationships determined by the data model(s) used. The IMG envelope is independent from the IMG transport protocol.

There is no standard format for the IMG envelope yet. However, XML based format [9] and MIME based format are candidates for it. [9] defines essential description format for IMG envelope parameters such as version, reference, and validation using XML.

On the other hand, the MIME based format has to be defined to support these parameters to meet the IMG requirements. When it will successfully reuse the existing MIME headers or define new headers, the cost to implement the IMG envelope may be less than xml based format.

In either case, the IMG envelope will be independent from an IMG transport protocol. Therefore, this document does not assume any non-standard IMG envelope format.

6 Example Message Flow

6.1 Example MIME Header

This section uses example MIME based description for IMG envelope in order to explain how the protocol works. As mentioned in [Section 5](#), the standard IMG envelope format is still developing. Consequently, this example does not mean this protocol is subject to the MIME based IMG envelope.

This section introduces four example MIME headers.

```
Content-Type: application/xml
```

Content-Type indicates a format of the IMG metadata description. If IMG metadata is described by an xml based

format, application/xml is used. If SDP describes IMG

metadata, application/sdp is appropriate.

Content-ID: major.minor

The IMG Envelope supports delta information between initial IMG metadata and update IMG metadata. Initial IMG metadata has certain version number described by "major" number. Update IMG metadata is also described by "minor" number.

Expires: "valid until"

IMG metadata may have period of validity. The MIME "Expires" header can be applicable to this purpose. IMG metadata may require "valid from" information. In this case, new MIME header will be defined to support this period. This example just uses "Expires" header. This example assume that this field value is defined by the [RFC 1123\[10\]](#)-date format.

Content-Location: "metadata URI"

"Metadata URI" identifies original IMG metadata. For instance, it may be URN as a name space or URL providing IMG metadata.

[6.2](#) Example Flows

When an IMG receiver needs to subscribe IMG metadata, it sends subscribe message to an IMG Sender (F1). If the sender accept this request, it sends the reply message (F2). After that message, the sender sends initial IMG metadata to the receiver (F3) and the receiver acknowledges the initial IMG metadata (F4).

After a while, when metadata changes, the sender notifies this to the receiver (F5) and the receiver acknowledges it (F6).

IMG Receiver	IMG Sender
F1 SUBSCRIBE	
----->	
F2 200 OK	
<-----	
F3 NOTIFY	
<-----	
F4 200 OK	
----->	
	<-- Update metadata
F5 NOTIFY	
<-----	
F6 200 OK	
----->	

F1 SUBSCRIBE receiver.com->sender.com

```

SUBSCRIBE sip:img@sender.com SIP/2.0
Via: SIP/2.0/TCP host1.receiver.com;branch=z9hG4bKnashds7
To: <sip:img@sender.com>
From: <sip:img@receiver.com>;tag=xfg9
Call-ID: 7070@host1.receiver.com
CSeq: 15024 SUBSCRIBE
Max-Forwards: 70
Event: img
Contact: <sip:metadata@host.receiver.com>
Expires: 300
Content-Length: 0

```

F2 200 OK sender.com->receiver.com

```

SIP/2.0 200 OK
Via: SIP/2.0/TCP host1.receiver.com;branch=z9hG4bKnashds7
;received=192.0.2.2
To: <sip:img@sender.com>;tag=ffd2
From: <sip:img@receiver.com>;tag=xfg9
Call-ID: 7070@host1.receiver.com
CSeq: 15024 SUBSCRIBE
Event: img
Expires: 300
Contact: <sip:metadata@host2.sender.com>
Content-Length: 0

```


F3 NOTIFY sender.com-> receiver.com

NOTIFY metadata@host.receiver.com SIP/2.0
Via: SIP/2.0/TCP host2.sender.com;branch=z9hG4bKna998sk
From: <sip:img@sender.com>;tag=ffd2
To: <sip:img@receiver.com>;tag=xfg9
Call-ID: 7070@host1.receiver.com
Event: img
Subscription-State: active;expires=599
Max-Forwards: 70
CSeq: 3487 NOTIFY
Content-Type: application/xml
Content-ID: 2798.8208
Expires: Mon, 25 Jul 2005 08:03:55 GMT
Content-Location: <http://sender.com/ID/2798>
Content-Length: ...

F4 200 OK receiver.com-> sender.com

SIP/2.0 200 OK
Via: SIP/2.0/TCP host2.sender.com;branch=z9hG4bKna998sk
;received=192.0.2.2
From: <sip:img@receiver.com>;tag=ffd2
To: <sip:img@sender.com>;tag=xfg9
Call-ID: 7070@host1.receiver.com
CSeq: 3487 NOTIFY

F5 NOTIFY sender.com -> receiver.com

NOTIFY sip:metadata@host.receiver.com SIP/2.0
Via: SIP/2.0/TCP host2.sender.com;branch=z9hG4bKna998s1
From: <sip:img@sender.com>;tag=ffd2
To: <sip:img@receiver.com>;tag=xfg9
Call-ID: 7070@host1.receiver.com
CSeq: 3488 NOTIFY
Event: img
Subscription-State: active;expires=543
Content-Type: application/xml
Content-ID: 2798.8344
Expires: Mon, 25 Jul 2005 14:00:00 GMT
Content-Location: <http://sender.com/ID/2798>
Content-Length: ...

F6 200 OK receiver.com-> sender.com

SIP/2.0 200 OK
Via: SIP/2.0/UDP notifier.example.com;branch=z9hG4bKna998s1
;received=192.0.2.2
From: <sip:img@receiver.com>;tag=ffd2
To: <sip:img@sender.com>;tag=xfg9
Call-ID: 7070@host1.receiver.com
CSeq: 3488 NOTIFY
Content-Length: 0

7 Security Considerations

TBD

8 IANA Considerations

TBD

9 Normative References

[1] S. Bradner, "Key words for use in RFCs to indicate requirement levels," [RFC 2119](#), Internet Engineering Task Force, March 1997.

10 Informative References

[2] Y. Nomura, R. Walsh, J-P. Luoma, H. Asaeda, and H. Schulzrinne, "A framework for the usage of Internet media guides," Internet Draft [draft-ietf-mmusic-img-framework-09](#), Internet Engineering Task Force, December 2005. Work in progress.

[3] Y. Nomura, R. Walsh, J-P. Luoma, J. Ott, and H. Schulzrinne, "Requirements for Internet Media Guides," Internet Draft [draft-ietf-mmusic-img-req-08](#), Internet Engineering Task Force, December 2005. Work in progress.

[4] M. Handley and V. Jacobson, "SDP: session description protocol," [RFC 2327](#), Internet Engineering Task Force, Apr. 1998.

[5] D. Kutscher, J. Ott, and C. Bormann, "Session description and capability negotiation," Internet Draft [draft-ietf-mmusic-sdpng-07](#), Internet Engineering Task Force, Oct. 2003. Work in progress.

[6] A. B. Roach, "Session initiation protocol (sip)-specific event notification," [RFC 3265](#), Internet Engineering Task Force, June 2002.

[7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol," [RFC 3261](#), Internet Engineering Task Force, June 2002.

[8] J. Greifenberg, "Identifiers for Internet Media Guides (IMG)," Internet Draft [draft-greifenberg-mmusic-img-urn-01](#), Internet Engineering Task Force, December 2005.

[9] R. Walsh, J-P. Luoma, J. Peltotalo, S. Peltotalo, and J. Greifenberg, "The IMG Envelope," Internet Draft [draft-walsh-mmusic-img-envelope-03](#), Internet Engineering Task Force, Jun. 2005. Work in progress.

[10] R. Braden., "Requirements for Internet Hosts -- Application and Support", [RFC 1123](#), Internet Engineering Task Force, Oct. 1989.

11 Authors' Addresses

Yuji Nomura
Fujitsu Laboratories Ltd.
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki 211-8588
Japan
Email: nom@flab.fujitsu.co.jp

Hitoshi Asaeda
Graduate School of Media and Governance
Keio University
5322 Endo, Fujisawa-shi, Kanagawa-ken, 252-8520
Japan
Email: asaeda@wide.ad.jp

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
Email: schulzrinne@cs.columbia.edu

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 (2005). 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.

Acknowledgement

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

