

**A Message Summary and Message Waiting Indication Event Package for
the Session Initiation Protocol (SIP)
draft-ietf-sipping-mwi-03.txt**

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/lid-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 27, 2003.

Copyright Notice

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

Abstract

This document describes a SIP event package to carry message waiting status and message summaries from a messaging system to an interested User Agent.

Table of Contents

1.	Conventions	3
2.	Background and Appropriateness	3
3.	Event Package Formal Definition	4
3.1	Event Package Name	4
3.2	Event Package Parameters	4
3.3	SUBSCRIBE Bodies	4
3.4	Subscription Duration	4
3.5	NOTIFY Bodies	4
3.6	Subscriber generation of SUBSCRIBE requests	6
3.7	Notifier processing of SUBSCRIBE requests	6
3.8	Notifier generation of NOTIFY requests	7
3.9	Subscriber processing of NOTIFY requests	7
3.10	Handling of Forked Requests	7
3.11	Rate of notifications	7
3.12	State Agents and Lists	8
3.13	Behavior of a Proxy Server	8
4.	Examples of Usage	8
4.1	Example Message Flow	8
4.2	Example Usage with Callee Capabilities	14
5.	Formal Syntax	14
5.1	New event-package definition	14
5.2	Body Format Syntax	14
6.	Security Considerations	15
7.	IANA Considerations	15
7.1	SIP Event Package Registration for message-summary	15
7.2	MIME Registration for application/simple-message-summary	16
8.	Revision history	16
8.1	Changes from draft-ietf-sipping-mwi-01 and -02	16
8.2	Changes from draft-ietf-sipping-mwi-00	17
8.3	Changes from draft-mahy-sipping-mwi-00	17
8.4	Changes from draft-mahy-sip-mwi-01	17
8.5	Changes from draft-mahy-sip-mwi-00	18
9.	Contributors	18
10.	Acknowledgments	18
	Normative References	19
	Informational References	19
	Author's Address	20
	Intellectual Property and Copyright Statements	21

1. Conventions

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](#) [3].

2. Background and Appropriateness

Messaging Waiting Indication is a common feature of telephone networks. It typically involves an audible or visible indication that messages are waiting, such as playing a special dial tone (which in telephone networks is called message-waiting dial tone), lighting a light or indicator on the phone, displaying icons or text, or some combination.

Message-waiting dial tone is similar to but distinct from stutter dial tone. Both are defined in GR-506 [11].

The methods in the SIP [1] base specification were only designed to solve the problem of session initiation for multimedia sessions, and rendezvous. Since Message Waiting Indication is really status information orthogonal to a session, it was not clear how an IP telephone acting as a SIP User Agent would implement comparable functionality. Members of the telephony community viewed this as a shortcoming of SIP.

Users want the useful parts of the functionality they have using traditional analog, mobile, and PBX telephones. It is also desirable to provide comparable functionality in a flexible way that allows for more customization and new features. SIP Specific Event Notification ([RFC 3265](#) -- SIP Events) [2] is an appropriate mechanism to use in this environment, as it preserves the user mobility and rendezvous features which SIP provides.

Using SIP-Specific Event Notification, A Subscriber User Agent (typically an IP phone or SIP software User Agent) subscribes to the status of their messages. A SIP User Agent acting on behalf of the user's messaging system then notifies the Subscriber whenever the messaging account's messages have changed. (This Notifier could be composed with a User Agent that provides a real-time media interface to send or receive messages, or it could be a standalone entity.) The Notifier sends message summary information in the body of a NOTIFY, encoded in a new MIME type defined later in this document. A User Agent can also explicitly fetch the current status.

A SIP User Agent MAY subscribe to multiple accounts (distinguished by the Request URI). Multiple SIP User Agents MAY subscribe to the same account.

Before any subscriptions or notifications are sent, each interested User Agent must be made aware of its messaging notifier(s). This MAY be manually configured on interested User Agents, manually configured on an appropriate SIP Proxy, or dynamically discovered based on registered callee capabilities [4]. (For more information on usage with callee capabilities, see [Section 4.2](#))

[3. Event Package Formal Definition](#)

[3.1 Event Package Name](#)

This document defines a SIP Event Package as defined in [RFC 3265](#) [2]. The event-package token name for this package is:

"message-summary"

[3.2 Event Package Parameters](#)

This package does not define any event package parameters.

[3.3 SUBSCRIBE Bodies](#)

This package does not define any SUBSCRIBE bodies.

[3.4 Subscription Duration](#)

Subscriptions to this event package MAY range from minutes to weeks. Subscriptions in hours or days are more typical and are RECOMMENDED. The default subscription duration for this event package is one hour.

[3.5 NOTIFY Bodies](#)

A simple text-based format is proposed to prevent an undue burden on low-end user agents, for example, inexpensive IP phones with no display. Although this format is text-based, it is intended for machine consumption only.

A future extension MAY define other NOTIFY bodies. If no "Accept" header is present in the SUBSCRIBE, the body type defined in this document MUST be assumed.

The format specified in this proposal attempts to separate orthogonal attributes of messages as much as possible. Messages are separated by message-context-class (for example: voice-message, fax-message, pager-message, multimedia-message, text-message, and none); by message status (new and old); and by urgent and non-urgent type.

The text format begins with a simple status line, and optionally a summary line per message-context-class. Message-context-classes are defined in [6]. For each message-context-class, the total number of new and old messages is reported in the new and old fields.

In some cases, detailed message summaries are not available. The status line allows messaging systems or messaging gateways to provide the traditional boolean message waiting notification.

Messages-Waiting: yes

If the Request-URI or To header in a message-summary subscription corresponds to a group or collection of individual messaging accounts, the notifier MUST specify to which account the message-summary body corresponds. Note that the account URI MUST NOT be delimited with angle brackets (" $<$ " and " $>$ ").

Message-Account: sip:alice@example.com

In the example that follows, more than boolean message summary information is available to the User Agent. There are two new and four old fax messages.

Fax-Message: 2/4

After the summary, the format can optionally list a summary count of urgent messages. In the next example there are one new and three old voice messages, none of the new messages are urgent, but one of the old messages is. All counters have a maximum value of 4,294,967,295 ($(2^{32}) - 1$). Notifiers MUST NOT generate a request with a larger value. Subscribers MUST treat a larger value as $2^{32}-1$.

Voice-Message: 1/3 (0/1)

Optionally, after the summary counts, the messaging systems MAY append [RFC 2822](#) [9]-style message headers, which further describe newly added messages. Message headers MUST NOT be included in an initial NOTIFY, as new messages could be essentially unbounded in size. Message headers included in subsequent notifications MUST only correspond to messages added since the previous notification for that subscription. A messaging system which includes message headers in a NOTIFY, MUST provide an administrator configurable mechanism for selecting which headers are sent. Likely headers for inclusion include To, From, Date, Subject, and Message-ID. Note that the formatting of these headers in this body is identical to that of SIP extension-headers, not the (similar) format defined in [RFC 2822](#).

Implementations which generate large notifications are reminded to

follow the message size restrictions for unreliable transports articulated in [Section 18.1.1](#) of SIP.

Mapping local message state to new/old message status and urgency is an implementation issue of the messaging system. However, the messaging notifier **MUST NOT** consider a message "old" merely because it generated a notification, as this could prevent another subscription from accurately receiving message-summary notifications. Likewise, the messaging system **MAY** use any suitable algorithm to determine that a message is "urgent".

Messaging systems **MAY** use any algorithm for determining the appropriate message-context-class for a specific message. Systems which use Internet Mail **SHOULD** use the contents of the Message-Context header [6] (defined in [RFC 3458](#)) if present as a hint to make a context determination. Note that a composed messaging system does not need to support a given context in order to generate notifications identified with that context.

3.6 Subscriber generation of SUBSCRIBE requests

Subscriber User Agents will typically **SUBSCRIBE** to message summary information for a period of hours or days, and automatically attempt to re-**SUBSCRIBE** well before the subscription is completely expired. If re-subscription fails, the Subscriber **SHOULD** periodically retry again until a subscription is successful, taking care to backoff to avoid network congestion. If a subscription has expired, new re-subscriptions **MUST** use a new Call-ID.

The Subscriber **SHOULD** **SUBSCRIBE** to that user's message summaries whenever a new user becomes associated with the device (a new login). The Subscriber **MAY** also explicitly fetch the current status at any time. The subscriber **SHOULD** renew its subscription immediately after a reboot, or when the subscriber's network connectivity has just been re-established.

The Subscriber **MUST** be prepared to receive and process a **NOTIFY** with new state immediately after sending a new **SUBSCRIBE**, a **SUBSCRIBE** renewal, an **unsubscribe**, or a **fetch**; or at any time during the subscription.

When a user de-registers from a device (logoff, power down of a mobile device, etc.), subscribers **SHOULD** **unsubscribe** by sending a **SUBSCRIBE** message with an **Expires** header of zero.

3.7 Notifier processing of SUBSCRIBE requests

When a SIP Messaging System receives **SUBSCRIBE** messages with the

message-summary event-type, it SHOULD authenticate the subscription request. If authentication is successful, the Notifier MAY limit the duration of the subscription to an administrator defined amount of time as described in SIP Events.

3.8 Notifier generation of NOTIFY requests

Immediately after a subscription is accepted, the Notifier MUST send a NOTIFY with the current message summary information. This allows the Subscriber to resynchronize its state. This initial synchronization NOTIFY MUST NOT include the optional [RFC 2822](#) [9]-style message headers.

When the status of the messages changes sufficiently for a messaging account to change the number of new or old messages, the Notifier SHOULD send a NOTIFY message to all active subscribers to that account. NOTIFY messages sent to subscribers of a group or alias, MUST contain the message account name in the notification body.

A Messaging System MAY send a NOTIFY with an "Expires" header of "0" and a "Subscription-State" header of "terminated" before a graceful shutdown.

3.9 Subscriber processing of NOTIFY requests

Upon receipt of a valid NOTIFY request, the subscriber SHOULD immediately render the message status and summary information to the end user in an implementation specific way.

The Subscriber MUST be prepared to receive NOTIFYs from different Contacts corresponding to the same SUBSCRIBE. (the SUBSCRIBE may have been forked).

3.10 Handling of Forked Requests

Forked requests are allowed for this event type and may install multiple subscriptions. The Subscriber MAY render multiple summaries which correspond to the same account directly to the user, or MAY merge them as described below.

If any of the "Messages-Waiting" status lines report "yes", then the merged state is "yes"; otherwise the merged state is "no".

The Subscriber MAY merge summary lines in an implementation-specific way if all notifications contain at least one msg-summary line.

3.11 Rate of notifications

A Notifier MAY choose to hold NOTIFY requests in "quarantine" for a short administrator-defined period (seconds or minutes) when the message status is changing rapidly. Requests in the quarantine which become invalid are replaced by newer notifications, thus reducing the total volume of notifications. This behavior is encouraged for implementations with heavy interactive use. Note that timely notification which results in a change of overall state (messages waiting or not), and notification of newly added messages is probably more significant to the end user than a notification of newly deleted messages which do not affect the overall message waiting state (e.g. there are still new messages).

Notifiers SHOULD NOT generate NOTIFY requests more frequently than once per second.

3.12 State Agents and Lists

A Subscriber MAY use an "alias" or "group" in the Request-URI of a subscription if that name is significant to the messaging system. Implementers MAY create a service which consolidates and summarizes NOTIFYS from many Contacts. This document does not preclude implementations from building state agents which support this event package. One way to implement such a service is with the event list extension [[10](#)].

3.13 Behavior of a Proxy Server

There are no additional requirements on a SIP Proxy, other than to transparently forward the SUBSCRIBE and NOTIFY methods as required in SIP. However, Proxies SHOULD allow non-SIP URLs. Proxies and Redirect servers SHOULD be able to direct the SUBSCRIBE request to an appropriate messaging notifier User Agent.

4. Examples of Usage

4.1 Example Message Flow

The examples shown below are for informational purposes only. For a normative description of the event package, please see sections [3](#) and [5](#) of this document.

In the example call flow below, Alice's IP phone subscribes to the status of Alice's messages. Via headers are omitted for clarity.

Subscriber	Notifier
A1: SUBSCRIBE (new)	
----->	


```

| A2: 200 OK |
|<-----|
| A3: NOTIFY (sync) |
|<-----|
| A4: 200 OK |
|----->|
| |
| A5: NOTIFY (change) |
|<-----|
| A6: 200 OK |
|----->|
| |
| A7: (re)SUBSCRIBE |
|----->|
| A8: 200 OK |
|<-----|
| A9: NOTIFY (sync) |
|<-----|
| A10: 200 OK |
|----->|
| |
| A11: (un)SUBSCRIBE |
|----->|
| A12: 200 OK |
|<-----|
| A13: NOTIFY (sync) |
|<-----|
| A14: 200 OK |
|----->|

```

A1: Subscriber (Alice's phone) ->
 Notifier (Alice's voicemail gateway)
 Subscribe to Alice's message summary status for 1 day.

```

SUBSCRIBE sip:alice@vmail.example.com SIP/2.0
To: <sip:alice@example.com>
From: <sip:alice@example.com>;tag=78923
Date: Mon, 10 Jul 2000 03:55:06 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 4 SUBSCRIBE
Contact: <sip:alice@alice-phone.example.com>

```


Event: message-summary
Expires: 86400
Accept: application/simple-message-summary
Content-Length: 0

A2: Notifier -> Subscriber

SIP/2.0 200 OK
To: <sip:alice@example.com>;tag=4442
From: <sip:alice@example.com>;tag=78923
Date: Mon, 10 Jul 2000 03:55:07 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 4 SUBSCRIBE
Expires: 86400
Content-Length: 0

A3: Notifier -> Subscriber
(immediate synchronization of current state:
2 new and 8 old [2 urgent] messages)

NOTIFY sip:alice@alice-phone.example.com SIP/2.0
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 03:55:07 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 20 NOTIFY
Contact: <sip:alice@vmail.example.com>
Event: message-summary
Subscription-State: active
Content-Type: application/simple-message-summary
Content-Length: 99

Messages-Waiting: yes
Message-Account: sip:alice@vmail.example.com
Voice-Message: 2/8 (0/2)

A4: Subscriber -> Notifier

SIP/2.0 200 OK
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 03:55:08 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 20 NOTIFY
Content-Length: 0

A5: Notifier -> Subscriber
This is a notification of new messages.

Some headers from each of the new messages are appended.

NOTIFY sip:alice@alice-phone.example.com SIP/2.0
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 04:28:53 GMT
Contact: <sip:alice@vmail.example.com>
Call-ID: 1349882@alice-phone.example.com
CSeq: 31 NOTIFY
Event: message-summary
Subscription-State: active
Content-Type: application/simple-message-summary
Content-Length: 503

Messages-Waiting: yes
Message-Account: sip:alice@vmail.example.com
Voice-Message: 4/8 (1/2)

To: <alice@atlanta.com>
From: <bob@biloxi.com>
Subject: carpool tomorrow?
Date: Sun, 09 Jul 2000 21:23:01 -0700
Priority: normal
Message-ID: 13784434989@vmail.example.com

To: <alice@example.com>
From: <cathy-the-bob@example.com>
Subject: HELP! at home ill, present for me please
Date: Sun, 09 Jul 2000 21:25:12 -0700
Priority: urgent
Message-ID: 13684434990@vmail.example.com

A6: Subscriber -> Notifier

SIP/2.0 200 OK
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 04:28:53 GMT
Call-ID: 1349882@alice-phone.example.com
CSeq: 31 NOTIFY
Content-Length: 0

A7: Subscriber -> Notifier
Refresh subscription.

SUBSCRIBE sip:alice@vmail.example.com SIP/2.0
To: <sip:alice@example.com>;tag=4442
From: <sip:alice@example.com>;tag=78923

Date: Mon, 10 Jul 2000 15:55:06 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 8 SUBSCRIBE
Contact: <sip:alice@alice-phone.example.com>
Event: message-summary
Expires: 86400
Accept: application/simple-message-summary
Content-Length: 0

A8: Notifier -> Subscriber

SIP/2.0 200 OK
To: <sip:alice@example.com>;tag=4442
From: <sip:alice@example.com>;tag=78923
Date: Mon, 10 Jul 2000 15:55:07 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 8 SUBSCRIBE
Contact: <sip:alice@alice-phone.example.com>
Expires: 86400
Content-Length: 0

A9: Notifier -> Subscriber
(immediate synchronization of current state)

NOTIFY sip:alice@alice-phone.example.com SIP/2.0
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 15:55:07 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 47 NOTIFY
Contact: <sip:alice@vmail.example.com>
Event: message-summary
Subscription-State: active
Content-Type: application/simple-message-summary
Content-Length: 99

Messages-Waiting: yes
Message-Account: sip:alice@vmail.example.com
Voice-Message: 4/8 (1/2)

A10: Subscriber -> Notifier

SIP/2.0 200 OK
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 15:55:08 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 47 NOTIFY

Contact: <sip:alice@vmail.example.com>

A11: Subscriber -> Notifier

Un-subscribe after "alice" logs out.

SUBSCRIBE sip:alice@vmail.example.com SIP/2.0
To: <sip:alice@example.com>;tag=4442
From: <sip:alice@example.com>;tag=78923
Date: Mon, 10 Jul 2000 19:35:06 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 17 SUBSCRIBE
Contact: <sip:alice@alice-phone.example.com>
Event: message-summary
Expires: 0
Accept: application/simple-message-summary
Content-Length: 0

A12: Notifier -> Subscriber

SIP/2.0 200 OK
To: <sip:alice@example.com>;tag=4442
From: <sip:alice@example.com>;tag=78923
Date: Mon, 10 Jul 2000 19:35:07 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 17 SUBSCRIBE
Contact: <sip:alice@alice-phone.example.com>
Expires: 0
Content-Length: 0

A13: Notifier -> Subscriber

(immediate synchronization of current state,
which the subscriber can now ignore)

NOTIFY sip:alice@alice-phone.example.com SIP/2.0
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 19:35:07 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 56 NOTIFY
Contact: <sip:alice@vmail.example.com>
Event: message-summary
Subscription-State: terminated;reason=timeout
Content-Type: application/simple-message-summary
Content-Length: 99

Messages-Waiting: yes

Message-Account: sip:alice@vmail.example.com

Voice-Message: 4/8 (1/2)

A10: Subscriber -> Notifier

```
SIP/2.0 200 OK
To: <sip:alice@example.com>;tag=78923
From: <sip:alice@example.com>;tag=4442
Date: Mon, 10 Jul 2000 19:35:08 GMT
Call-Id: 1349882@alice-phone.example.com
CSeq: 56 NOTIFY
Event: message-summary
Content-Length: 0
```

[4.2](#) Example Usage with Callee Capabilities

The use of callee capabilities is optional but encouraged. If callee capabilities is used, a messaging notifier MAY REGISTER a Contact with an appropriate methods and events tag as shown in the example below. To further distinguish itself, the messaging notifier MAY also REGISTER as a Contact with the feature="msgserver" tag. An example of this kind of registration follows below.

```
REGISTER sip:sip3-sj.example.com SIP/2.0
To: <sip:alice@example.com>
From: <sip:alice@example.com>;tag=4442
...
Contact: <sip:alice@vm13-sj.example.com>
        ;msgserver="true";methods="SUBSCRIBE"
        ;events="message-summary"
```

[5](#). Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in [RFC-2234](#) [5].

[5.1](#) New event-package definition

This document defines a new event-package with the package name:

message-summary

[5.2](#) Body Format Syntax

The formal syntax for application/simple-message-summary is below:


```
message-summary = msg-status-line CRLF
                  [msg-account CRLF]
                  [*(msg-summary-line CRLF)]
                  [ *opt-msg-headers ]

msg-status-line  = "Messages-Waiting" HCOLON msg-status
msg-status      = "yes" / "no"
msg-account     = "Message-Account" HCOLON Account-URI
Account-URI     = SIP-URI / SIPS-URI / absoluteURI

msg-summary-line = message-context-class HCOLON newmsgs SLASH oldmsgs
                  [ LPAREN new-urgentmsgs SLASH old-urgentmsgs RPAREN ]

opt-msg-headers = CRLF 1*(extension-header CRLF)

newmsgs = msgcount
oldmsgs = msgcount
new-urgentmsgs = msgcount
old-urgentmsgs = msgcount
msgcount = 1*DIGIT ; MUST NOT exceed 2^32-1
```

6. Security Considerations

Message summaries and optional message bodies contain information which is typically very privacy sensitive. At minimum, subscriptions to this event package SHOULD be authenticated and properly authorized. Furthermore, notifications SHOULD be encrypted and integrity protected using either end-to-end mechanisms, or the hop-by-hop protection afforded messages sent to SIPS URIs.

Additional and privacy security considerations are discussed in detail in SIP [1] and SIP Events [2].

7. IANA Considerations

7.1 SIP Event Package Registration for message-summary

Package name: message-summary

Type: package

Contact: [Mahy]

Published Specification: This document.

7.2 MIME Registration for application/simple-message-summary

MIME media type name: application

MIME subtype name: simple-message-summary

Required parameters: none.

Optional parameters: none.

Encoding considerations: This type is only defined for transfer via SIP [[1](#)].

Security considerations: See the "Security Considerations" section in this document.

Interoperability considerations: none

Published specification: This document.

Applications which use this media: The simple-message-summary application subtype supports the exchange of message waiting and message summary information in SIP networks.

Additional information:

1. Magic number(s): N/A
2. File extension(s): N/A
3. Macintosh file type code: N/A

8. Revision history

** Note to the RFC editor: please remove this entire section upon publication. **

8.1 Changes from [draft-ietf-sipping-mwi-01](#) and -02

1. Updated the caller-preference section (now the callee capabilities section) to reflect the split of these drafts and the new tag ;msgserver="true".
2. Added some text in the overview to further clarify how message notifiers can be composed/decomposed with media processing.
3. Add a pointer to the event-list extension.

8.2 Changes from [draft-ietf-sipping-mwi-00](#)

1. Replaced the "media types" concept with message contexts. This is a better semantic match than what was in the draft before, and also controls extensibility and change control in a single document. The list of valid message-context-classes are voice-message, fax-message, pager-message, multimedia-message, text-message, and none.
2. Completely updated the syntax to follow that of SIP instead of the previously more restrictive (and somewhat arbitrary) syntax. The SIP syntax adds line folding, for example. The optional message-headers borrow the "extension-header" syntax and explicit whitespace separators defined in SIP (ex: HCOLON, SLASH).
3. Added a Message-Account field in the body format to provide the specific account name which corresponds to the notification when forking or state agents are used with group aliases (or collections).
4. Changed caller preferences example to exclude methods="SUBSCRIBE" in the SUBSCRIBE request (removed redundant information).
5. Changed examples to be consistent with IESG recommendations

8.3 Changes from [draft-mahy-sipping-mwi-00](#)

1. Updated references and split into normative and informational
2. Removed normative behavior now specified in Events
3. Updated to address the event package sections now specified in Events.
4. Added the Subscription-State header field to the examples and removed the Event header field from responses.
5. Removed redundant BNF
6. Simplified text on how to choose the media type. For Internet Mail, this now references the Message-Context header.

8.4 Changes from [draft-mahy-sip-mwi-01](#)

1. This document is now formatted as a SIP Event Package as defined in [Section 4 of RFC 3265](#) (SIP Events) [2].

2. The event-package name is now "message-summary", to allow for other bodies to extend the package.
3. The "urgent" token was missing from the BNF.

8.5 Changes from [draft-mahy-sip-mwi-00](#)

This draft greatly simplifies and shortens the -00 version.

1. The generic behavior of SUBSCRIBE/NOTIFY is now greatly clarified in SIP Events [2] and made consistent with PINT and SIP for presence. This message waiting draft is now consistent with SIP Events.
2. The XML format has been removed due to lack of immediate interest. At a future date, similar functionality may be added as another body definition with an appropriate MIME type.
3. An IANA Considerations section was added to register the new "application/simple-message-summary" MIME type and the "simple-message-summary" SIP event package.
4. The "flag-list" was removed due to lack of interest and to encourage simplicity.
5. Due to synchronization issues, and the recommendation of the VPIM Working Group, support for message count "deltas" was removed.
6. The Messages-Waiting line in the body is now mandatory.
7. This version of the draft clarifies the role of caller preferences as optional but encouraged.
8. A set of SMTP-like headers from the triggering messages may now optionally follow the message summaries, provided that the resulting NOTIFY on UDP fits in a single datagram.

9. Contributors

Ilya Slain came up with the initial format of the text body contained in this document. He was previously listed as a co-author, however, he is no longer reachable.

10. Acknowledgments

Thanks to Dan Wing, Dave Oran, Bill Foster, Steve Levy, Denise

Caballero-McCann, Jeff Michel, Priti Patil, Satyender Khatter, Bich Nguyen, Manoj Bhatia, David Williams, and Bryan Byerly of Cisco; Jonathan Rosenberg and Adam Roach of Dynamicsoft; Eric Burger of Snowshore; Nir Chen of iComverse, and Eric Tremblay of Mediatrix.

Normative References

- [1] 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.
- [2] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [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] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.
- [6] Burger, E., Candell, E., Eliot, C. and G. Klyne, "Message Context for Internet Mail", [RFC 3458](#), January 2003.

Informational References

- [7] Vaudreuil, G. and G. Parsons, "Voice Profile for Internet Mail - version 2", [RFC 2421](#), September 1998.
- [8] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.
- [9] Resnick, P., "Internet Message Format", [RFC 2822](#), April 2001.
- [10] Rosenberg, J., Roach, A. and B. Campbell, "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", [draft-ietf-simple-event-list-04](#) (work in progress), June 2003.
- [11] Telcordia, "GR-506: Signaling for Analog Interfaces, Issue 1, Revision 1", Nov 1996.

Author's Address

Rohan Mahy
Cisco Systems, Inc.
101 Cooper Street
Santa Cruz, CA 95060
USA

EMail: rohan@cisco.com

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.