

Registration Event Package Extension for
Session Initiation Protocol (SIP)
Globally Routable User Agent URIs (GRUUs)

Abstract

RFC 3680 defines a Session Initiation Protocol (SIP) event package for registration state. This package allows a watcher to learn about information stored by a SIP registrar, including its registered contact.

However, the registered contact is frequently unreachable and thus not useful for watchers. The Globally Routable User Agent URI (GRUU), defined in RFC 5627, is a URI that is capable of reaching a particular contact. However this URI is not included in the document format defined in RFC 3680. This specification defines an extension to the registration event package to include GRUUs assigned by the registrar.

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Description	4
4. Notifier Processing of SUBSCRIBE Requests	4
5. Notifier Generation of NOTIFY Requests	4
6. Subscriber Processing of NOTIFY Requests	5
6.1. Managing Temporary GRUU Lifetime	5
7. Sample reginfo Document	7
8. Examples	7
8.1. Example: Welcome Notice	8
8.2. Example: Implicit Registration	8
9. XML Schema Definition	11
10. IANA Considerations	12
10.1. URN Sub-Namespace Registration	12
10.2. XML Schema Registration	13
11. Security Considerations	13
12. Acknowledgements	14
13. References	14
13.1. Normative References	14
13.2. Informative References	14

1. Introduction

RFC 3680 [2] defines a Session Initiation Protocol (SIP) [5] event package for registration state. This package allows a watcher to learn about information stored by a SIP registrar, including the registered contacts.

However, a registered contact is frequently unreachable from hosts outside of the domain of the User Agent (UA). It is commonly a private address, or, when it is a public address, access to it may be blocked by firewalls.

The Globally Routable User Agent URI (GRUU), defined in RFC 5627 [3], is a URI that reaches a particular UA instance, but is reachable by any host on the Internet. GRUUs assigned by the registrar represent additional registration state. However, GRUUs assigned by the registrar are not included in the notifications provided by RFC 3680. For many applications of the registration event package, a GRUU is needed, and not the registered contact.

For example, the Welcome Notices example in [2] will only operate correctly if the contact address in the registration event notification is reachable by the sender of the welcome notice. When the registering device is using the GRUU extension, it is likely that the registered contact address will not be globally addressable, and a GRUU should be used as the target address for the MESSAGE.

Another case where this feature may be helpful is within the Third Generation Partnership Project (3GPP) IP Multimedia Subsystem (IMS). IMS employs a technique where a REGISTER of a contact address to one Address of Record (AOR) causes the implicit registration of the same contact to other associated AORs. If GRUUs are requested and obtained as part of the registration request, then additional GRUUs will also be needed for the implicit registrations. While assigning the additional GRUUs is straightforward, informing the registering UA of them is not. In IMS, UAs typically subscribe to the registration event, and subscriptions to the registration event for an AOR result in notifications, each containing the registration state of all the associated AORs. The proposed extension provides a way to easily deliver the GRUUs for the associated AORs.

As specified in RFC 5627 [3], temporary GRUUs are invalidated when contact address bindings for the corresponding AOR and instance ID are not refreshed, or when a registration to the AOR and instance ID is performed with a new Call-ID. A UA cannot always determine with certainty which temporary GRUUs are valid based solely on the response to the REGISTER requests it has issued, or from

notifications according to RFC 3680 [2]. The extension defined in this document provides sufficient information for a UA to determine which temporary GRUUs are valid.

The registration event package has provisions for including extension elements within the <contact> element. This document defines new elements that may be used in that context to deliver the public and temporary GRUUs corresponding to the contact.

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]

3. Description

Two new elements (<pub-gruu> and <temp-gruu>) are defined, each of which contains a GRUU. The <temp-gruu> element also identifies the oldest temporary GRUU that is currently valid.

These optional elements may be included within the body of a NOTIFY for the registration event package when GRUUs are associated with the contact. The contact URI and the GRUUs are then all available to the watcher.

4. Notifier Processing of SUBSCRIBE Requests

Unchanged from RFC 3680 [2].

5. Notifier Generation of NOTIFY Requests

A notifier for the registration event package [2] SHOULD include the <pub-gruu> element when a contact has an instance ID and a public GRUU is associated with the combination of the AOR and the instance ID. When present, the <pub-gruu> element MUST be positioned as a child of the <contact> element.

A notifier for the registration event package [2] MAY include the <temp-gruu> element when a contact has an instance ID and a temporary GRUU is associated with the combination of the AOR and the instance ID. This element SHOULD be included if the subscriber is also authorized to register to the AOR. This element SHOULD NOT be included if the subscriber is not authorized to register to the AOR, unless there is an explicitly configured policy directing that it be included. When present, the <temp-gruu> element MUST be positioned as a child of the <contact> element.

Note that it is possible for multiple registered contacts to share the same instance ID. In such a case, each <contact> element will have child <pub-gruu> and <temp-gruu> elements, which are identical to the corresponding child elements in those other <contact> elements that share the same instance ID. Since a particular contact cannot be associated with more than one instance ID, a <contact> element will never have more than one <pub-gruu> and one <temp-gruu> child element.

If the notifier includes the <pub-gruu> element, it MUST populate the element with the public GRUU that is associated with the instance ID and AOR of the registered contact.

If the notifier includes the <temp-gruu> element, it MUST populate the element with the most recently assigned temporary GRUU that is associated with the instance ID and AOR of the registered contact. It MUST also populate the element with a "cseq" attribute corresponding to the first (oldest) currently active temporary GRUU that is associated with the instance ID and AOR of the registered contact. The value of the "cseq" attribute is set to the value of the CSeq header field of the REGISTER request that caused that first temporary GRUU to be assigned.

6. Subscriber Processing of NOTIFY Requests

When a subscriber receives a registration event notification with a <contact> containing a <pub-gruu>, it MAY associate the public GRUU with the corresponding AOR and instance ID. Any previously received public GRUU for the same AOR and instance ID MUST be discarded. (It will no longer function.)

When a subscriber receives a registration event notification with a <contact> containing a <temp-gruu>, it MAY associate the temporary GRUU, together with the "callid" and "cseq" attributes, with the corresponding AOR and instance ID.

Subscribers that are unaware of this extension will, as required by [2], ignore the <pub-gruu> and <temp-gruu> elements.

6.1. Managing Temporary GRUU Lifetime

Section 4.2 of RFC 5627 [3] gives guidance for developers of UAs on how to ensure that only valid temporary GRUUs are retained and used by the UA. A UA cannot always determine with certainty which temporary GRUUs are valid based solely on the information contained in responses to the REGISTER requests it has issued or from the information contained in notifications that conform solely to RFC 3680 [2]. The extension defined in this document provides sufficient

added information for a UA to determine which temporary GRUUs are valid. The extension to RFC 3680 defined in this document provides added information to help with that process. The following are steps that the UA MAY take to ensure it only retains valid GRUUs:

- o The UA should subscribe to the registration event package for the AOR it is registering.
- o When a UA receives a 2xx response to a REGISTER request, it may extract and retain temporary GRUUs from the response for future use, as long as they remain valid. Appropriate GRUUs to retain are those corresponding to the contact address and instance ID it has registered. (Typically, the UA will register only one contact address, and so receive at most one temporary GRUU.)
- o The UA may add the temporary GRUU to the set of valid temporary GRUUs associated with the AOR. (Note, in this case AOR is the To-address of the REGISTER request.) To aid in tracking validity, the UA should also associate a "callid" attribute and "cseq" attribute with the temporary GRUU, with values obtained respectively from the Call-ID and CSeq values of the REGISTER response containing the temporary GRUU.
- o If the UA receives a registration event notification with an AOR (that it supports) and a <contact>, for a contact address and instance ID (that it has registered and that contains a <temp-gruu>), it may update its set of valid temporary GRUUs associated with the AOR, as follows:
 - * It may add the temporary GRUU to the set. To aid in tracking validity, the UA should associate the "callid" and "cseq" attributes of the <contact> with the GRUU in the set.
 - * It should remove any temporary GRUUs with a "callid" attribute value different from that in the value of the "callid" attribute of the <contact>, or with a "cseq" attribute with value less than the value of the "first-cseq" attribute of the <temp-gruu>.
- o If the UA receives a registration event notification with an AOR that it supports, and there are no <contact> entries for its instance ID, then it should discard all the temporary GRUUs it has saved for that AOR.

7. Sample reginfo Document

Note: This example and others in the following section are indented for readability by the addition of a fixed amount of whitespace to the beginning of each line. This whitespace is not part of the example. The conventions of [7] are used to describe representation of long message lines.

The following is an example registration information document that includes the new element:

```
<?xml version="1.0"?>
  <reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
    xmlns:gr="urn:ietf:params:xml:ns:gruinfo"
    version="0" state="full">
    <registration aor="sip:user@example.com" id="as9"
      state="active">
      <contact id="76" state="active" event="registered"
        duration-registered="36001" expires="3599"
        callid="1j9FpLxk3uxtm8tn@192.0.2.1" cseq="54321"
        q="0.8">
        <uri>sip:user@192.0.2.1</uri>
      <allOneLine>
        <unknown-param name="+sip.instance">
          "&lt;urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6&gt;"
        </unknown-param>
      </allOneLine>
    </allOneLine>
      <gr:pub-gruu uri="sip:user@example.com"
        ;gr=hha9s8d-999a"/>
    </allOneLine>
    <allOneLine>
      <gr:temp-gruu uri="sip:8ffkas08af7fasklzi9@example.com"
        ;gr" first-cseq="54301"/>
    </allOneLine>
      </contact>
    </registration>
  </reginfo>
```

8. Examples

Note: In the following examples, the SIP messages have been simplified, removing headers that are not pertinent to the example.

When the value of the Content-Length header field is "...", this means that the value should be the computed length of the body.

8.1. Example: Welcome Notice

Consider the Welcome Notices example in [2]. When the application server receives a notification of a new registration containing the reginfo shown in Section 7, it should address messages using the contained public GRUU as follows:

```
MESSAGE sip:user@example.com;gr=hha9s8d-999a SIP/2.0
To: <sip:user@example.com>
From: "SIPland Notifier" <sip:notifier@example.com>;tag=7xy8
Content-Type: text/plain
Content-Length: ...
```

```
Welcome to SIPland!
Blah, blah, blah.
```

8.2. Example: Implicit Registration

In a 3GPP IMS setting, a UA may send a single register message, requesting assignment of GRUUs, as follows:

```
REGISTER sip:example.net SIP/2.0
From: <sip:user_aor_1@example.net>;tag=5ab4
To: <sip:user_aor_1@example.net>
Call-ID: faif9a@ua.example.com
CSeq: 23001 REGISTER
Contact: <sip:ua.example.com>
        ;expires=3600
        ;+sip.instance="urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
Supported: path, gruu
Content-Length: 0
```

The response reports success of the registration and returns the GRUUs assigned for the combination of AOR, instance ID, and Contact. It also indicates (via the P-Associated-URI header [6]) that there are two other associated AORs that may have been implicitly registered using the same contact. Each of those implicitly registered AORs will have unique GRUUs assigned. The REGISTER response will not include those GRUUs; it will only include the GRUUs for the AOR and instance ID explicitly included in the registration.

```
SIP/2.0 200 OK
From: <sip:user_aor_1@example.net>;tag=5ab4
To: <sip:user_aor_1@example.net>;tag=373392
Call-ID: faif9a@ua.example.com
CSeq: 23001 REGISTER
Path: <sip:proxy.example.net;lr>
Service-Route: <sip:proxy.example.net;lr>
```

```
Contact: <sip:ua.example.com>
;expires=3600
;+sip.instance="urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6"
;pub-gruu="sip:user_aor_1@example.net;gr=hha9s8d-999a"
;temp-gruu="sip:8ffkas08af7fasklzi9@example.net;gr"
P-Associated-URI: <sip:user_aor_2@example.net>,
<sip:+358504821437@example.net;user=phone>
Content-Length: 0
```

The UA then subscribes to the registration event package as follows:

```
SUBSCRIBE sip:user_aor_1@example.net SIP/2.0
From: <sip:user_aor_1@example.net>;tag=27182
To: <sip:user_aor_1@example.net>
Call-ID: gbjg0b@ua.example.com
CSeq: 45001 SUBSCRIBE
Route: <sip:proxy.example.net;lr>
Event: reg
Expires: 3600
Accept: application/reginfo+xml
Contact: <sip:user_aor_1@example.net;gr=hha9s8d-999a>
Content-Length: 0
```

(The successful response to the subscription is not shown.) Once the subscription is established, an initial notification is sent giving registration status. In IMS deployments, the response includes, in addition to the status for the requested URI, the status for the other associated URIs.

```
NOTIFY sip:user_aor_1@example.net;gr=hha9s8d-999a SIP/2.0
From: <sip:user_aor_1@example.net>;tag=27182
To: <sip:user_aor_1@example.net>;tag=262281
Call-ID: gbjg0b@ua.example.com
CSeq: 633 NOTIFY
Subscription-State: active;expires=3600
Event: reg
Content-Type: application/reginfo+xml
Contact: <sip:registrar.example.net>
Content-Length: ...
```

```
<?xml version="1.0"?>
<reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
  xmlns:gr="urn:ietf:params:xml:ns:gruinfo"
  version="1" state="full">
  <registration aor="sip:user_aor_1@example.net" id="a7"
    state="active">
    <contact id="92" state="active" event="registered"
      duration-registered="1" expires="3599">
```

```

        callid="faif9a@ua.example.com" cseq="23001">
        <uri>
            sip:ua.example.com
        </uri>
<allOneLine>
    <unknown-param name="+sip.instance">
"&lt;urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6&gt;"
</unknown-param>
</allOneLine>
<allOneLine>
    <gr:pub-gruu uri="sip:user_aor_1@example.net
;gr=hha9s8d-999a"/>
</allOneLine>
<allOneLine>
    <gr:temp-gruu uri="sip:8ffkas08af7fasklzi9@example.net
;gr" first-cseq="54301"/>
</allOneLine>
    </contact>
</registration>
<registration aor="sip:user_aor_2@example.net" id="a8"
state="active">
    <contact id="93" state="active" event="created"
duration-registered="1" expires="3599"
callid="faif9a@ua.example.com" cseq="23001">
        <uri>
            sip:ua.example.com
        </uri>
<allOneLine>
    <unknown-param name="+sip.instance">
"&lt;urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6&gt;"
</unknown-param>
</allOneLine>
<allOneLine>
    <gr:pub-gruu uri="sip:user_aor_2@example.net
;gr=hha9s8d-999b"/>
</allOneLine>
<allOneLine>
    <gr:temp-gruu uri="sip:07hcovy36vp6vngvbia@example.net
;gr" first-cseq="54301"/>
</allOneLine>
    </contact>
</registration>
<registration
aor="sip:+358504821437@example.net;user=phone"
id="a9"
state="active">
    <contact id="94" state="active" event="created"
duration-registered="1" expires="3599"

```

```

        callid="faif9a@ua.example.com" cseq="23001">
        <uri>
            sip:ua.example.com
        </uri>
    <allOneLine>
        <unknown-param name="+sip.instance">
            "&lt;urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6&gt;"
        </unknown-param>
    </allOneLine>
    <allOneLine>
        <gr:pub-gruu uri="sip:+358504821437@example.net
;user=phone;gr=hha9s8d-999c"/>
    </allOneLine>
    <allOneLine>
        <gr:temp-gruu uri="sip:h99egjbv17fe8ibvlka@example.net
;gr" first-cseq="54301"/>
    </allOneLine>
    </contact>
</registration>
</reginfo>

```

The status indicates that the associated URIs all have the same contact registered. It also includes the unique GRUUs that have been assigned to each. The UA may then retain those GRUUs for use when establishing dialogs using the corresponding AORs.

9. XML Schema Definition

The <pub-gruu> and <temp-gruu> elements are defined within a new XML namespace URI. This namespace is "urn:ietf:params:xml:ns:gruuinfo". The schema for these elements is:

```

<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema targetNamespace="urn:ietf:params:xml:ns:gruuinfo"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="urn:ietf:params:xml:ns:gruuinfo">
    <xs:complexType name="pubGruu">
      <xs:attribute name="uri" type="xs:anyURI"
        use="required"/>
    </xs:complexType>
    <xs:complexType name="tempGruu">
      <xs:complexContent>
        <xs:extension base="tns:pubGruu">
          <xs:attribute name="first-cseq"
            type="xs:unsignedLong"
            use="required"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:schema>

```

```
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="pub-gruu" type="tns:pubGruu"/>
<xs:element name="temp-gruu" type="tns:tempGruu"/>
</xs:schema>
```

10. IANA Considerations

There are two IANA considerations associated with this specification.

10.1. URN Sub-Namespace Registration

This section registers a new XML namespace, per the guidelines in [4].

URI: The URI for this namespace is `urn:ietf:params:xml:ns:gruinfo`

Registrant Contact: IETF, SIPING working group, <sipping@ietf.org>, Paul Kyzivat <pkyzivat@cisco.com>

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>Reg Information GRUU Extension Namespace</title>
</head>
<body>
  <h1>Namespace for Reg Information GRUU Extension</h1>
  <h2>urn:ietf:params:xml:ns:gruinfo</h2>
  <p>See <a href="http://www.rfc-editor.org/rfc/rfc5628.txt">
    RFC5628</a>.</p>
</body>
</html>
END
```

10.2. XML Schema Registration

This section registers an XML schema per the procedures in [4].

URI: urn:ietf:params:xml:schema:gruuinfo.

Registrant Contact: IETF, SIPING working group, <sipping@ietf.org>, Paul Kyzivat <pkyzivat@cisco.com>

The XML for this schema can be found in Section 9.

11. Security Considerations

Security considerations for the registration event package are discussed in RFC 3680 [2], and those considerations apply here.

If a contact address obtained via subscription to the registration event package is not reachable by the subscriber, then its disclosure may arguably be considered a minimal security risk. In that case, the inclusion of a GRUU may be considered to increase the risk by providing a reachable address. On the other hand, requests addressed to a GRUU are always first processed by the servicing proxy before they reach the intended User Agent. The proxy may control access as desired, just as it may for the AOR. For instance, the proxy servicing a GRUU may accept requests from senders whose identity appears on a white list, and reject other requests. In this respect, disclosing a GRUU presents no more risk than disclosing the AOR.

Temporary GRUUs have an additional security consideration. The intent of the temporary GRUU is to provide a contact address that cannot be correlated to the identity of the calling party. The recipient of a call using a temporary GRUU may guess the identity of the calling party and then attempt to obtain the temporary GRUUs assigned to that caller to confirm the conjecture. Two possible approaches to obtaining the temporary GRUUs are:

- o Send a REGISTER request to a conjectured caller.
- o Send a SUBSCRIBE request for the registration event package to the conjectured caller.

Typically, REGISTER is restricted to devices or users that are authorized to originate and receive calls with the AOR. Anonymity among users of the same AOR is hard to achieve and typically unnecessary. It is recommended (see Section 5) that the authorization policy for the registration event package permit only those subscribers who are authorized to register to the AOR to receive temporary GRUUs. With this policy, the confidentiality of

the temporary GRUU will be the same with and without the registration event package. User Agents that use a temporary GRUU should note that confidentiality does not extend to parties that are permitted to register to the AOR or to obtain the temporary GRUU when subscribing to the registration event package.

12. Acknowledgements

The author would like to thank Jonathan Rosenberg for help with this document, and Jari Urpalainen for assistance with the XML.

13. References

13.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Registrations", RFC 3680, March 2004.
- [3] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", RFC 5627, October 2009.
- [4] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

13.2. Informative References

- [5] 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.
- [6] Garcia-Martin, M., Henrikson, E., and D. Mills, "Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)", RFC 3455, January 2003.
- [7] Sparks, R., Hawrylyshen, A., Johnston, A., Rosenberg, J., and H. Schulzrinne, "Session Initiation Protocol (SIP) Torture Test Messages", RFC 4475, May 2006.

Author's Address

Paul H. Kyzivat
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719
USA

EMail: pkyzivat@cisco.com

