    Extensions to the Session Initiation Protocol (SIP) User Agent Profile
     Delivery Change Notification Event Package for the Extensible Markup
         Language Language Configuration Access Protocol (XCAP)
                      draft-ietf-sip-xcap-config-00.txt

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/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 April 17, 2007.

Copyright Notice

Abstract

   The SIP User Agent profile delivery framework describes how a User
   Agent can retrieve its data using a variety of protocols and defines
   a SIP event package for notifications of changes to those profiles.
   This document extends that event package to support XCAP (XML
   Configuration Access Protocol).

Table of Contents

1. Introduction

The SIP [RFC3261] User Agent profile delivery framework [I-D.ietf-sipping-config-framework] describes how a User Agent (UA) can retrieve its data using a variety of protocols. The framework also defines a SIP event package [RFC3265] for notifications of changes to those profiles. This document extends that event package to support XCAP (XML Configuration Access Protocol) [I-D.ietf-simple-xcap].

XCAP is a usage of HTTP (HyperText Transfer Protocol) [RFC2616] which defines structure of the HTTP URI (Uniform Resource Identifier) to represent a specific document hierarchy. XCAP URIs consist of the document root, the Application Unique ID (AUID), the XCAP User Identifier (XUI), plus additional optional elements for selecting XML nodes within an XML document. The mandatory components point to a specific XCAP document. For example:

```
http://xcap.example.com/root/resource-lists/users/joe
|-------------v-------------||------v------||---v---|
       document root              AUID        XUI
```

This document extends the UA profile change event package with two new Event header parameters. These allow a UA to subscribe to change notification for a specific XCAP AUID or document.

2. Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [RFC2119].

3. New Event Package Parameters

This document extends the event package defined in Section 7 of [I-D.ietf-sipping-config-framework] with the following two new parameters for the Event header: "document" and "auid" and a new profile type: "application" for the "profile-type" parameter. These new parameters are for use in both SUBSCRIBE and NOTIFY requests. The motivation for these new parameters is in support of XCAP, but they may be used with other suitable protocols.

The "document" parameter is used to specify a relative URI for a specific profile document that the user agent wishes to retrieve and for which it wishes to receive change notification. It can be used with any of the profile-types. The document parameter is useful for profile content like XCAP [I-D.ietf-simple-xcap] where there is a

well defined URI schema and the user agent knows the specific content
that it wants. This provides a filtering mechanism to restrict the
content to be retrieved and for which change notification is to be
received. (The size of the content is important in limited bandwidth
environments.) The "document" parameter value syntax is a quoted
string. The values for the "document" parameter are defined as part
of the profile data format, which is out of scope for this document.
To use the "document" parameter, the profile data format, must also
define a URI path schema. For more details on the use of this
package and the "document" parameter with XCAP see Section 4. The
"document" parameter MAY be set in SUBSCRIBE requests for any of the
profile types. It is ignored in all other messages. In the
following ABNF EQUAL and quoted-string are defined in [RFC3261].

Document   = "document" EQUAL quoted-string

The "auid" parameter MAY be set when the "profile-type" parameter
value is "application". The "auid" indicates that the user agent
wishes to retrieve the profile data or URI and change notification
for the application profile data for the specific application
indicated in the value of the "auid" parameter. Like the "document"
parameter, the "auid" parameter provides a filtering mechanism on the
profile content. The "auid" parameter value is a quoted string. The
values for the "auid" parameter are defined as part of the profile
data format to be used with XCAP (see [I-D.ietf-simple-xcap]), which
is out of scope for this document. The "auid" parameter has meaning
only in SUBSCRIBE requests when the "profile-type" Event header
parameter is set to "application". When used with XCAP it is not
necessary to set both the "document" and "auid" parameters in a
SUBSCRIBE request as the document path will also include the
application auid. The "auid" parameter is ignored if it conflicts
with the parameter "document" path. The "auid" parameter is ignored
in all other messages.

AUID   = "auid" EQUAL quoted-string

The profile-type Event header parameter is extended to have the
additional profile type "application". Specifying "application" type
profile indicates the desire for the profile data (URI when content
indirection is used) and change notification of the profile content
for the user's applications. Specifying the "application" type
profile also implies the use of XCAP. If the profile-type is
"application" in the SUBSCRIBE request and the profile delivery
server does not support XCAP, a 439 Invalid Event Parameter Value
MUST be sent and profile-type=application MUST be added to the
Invalid-Parameters-Values response header field as described in
[I-D.dpetrie-sip-event-param-err]. The SUBSCRIBE request URI SHOULD
be discovered and constructed in the same way the "user" type

profiles described in [Section 7](#) of [I-D.ietf-sipping-config-framework].

    profile-types = "device" / "user" / "application" / "local-network"

The following is a SUBSCRIBE request Event header example:
Event: ua-profile;profile-type="application";
    document="user-aor/";
    vendor="premier";model="trs8000";version="5.5"

The following table shows the use of Event header parameters in
SUBSCRIBE requests for the four profile types:

```
profile-type || device | user | application | local-network
=========================================================
vendor        ||   m    |  m   |     m       |      m
model         ||   m    |  m   |     m       |      m
version       ||   m    |  m   |     m       |      m
network-user  ||   s    |      |             |      s
effective-by  ||        |      |             |
document      ||        |      |     o       |
auid          ||        |      |     o       |
```

m - mandatory
s - SHOULD be provided
o - optional
Non-specified means that the parameter has no meaning and
should be ignored.

The following table shows the use of Event header parameters in
NOTIFY requests for the four profile types:

```
profile-type || device | user | application | local-network
=========================================================
vendor        ||        |      |             |
model         ||        |      |             |
version       ||        |      |             |
network-user  ||   s    |      |             |      s
effective-by  ||   o    |  o   |     o       |      o
document      ||        |      |    o/m      |
auid          ||        |      |    o/m      |
```

o/m - mandatory if provided in the SUBSCRIBE request

4. Relationship of XCAP with the Data Model

The UA profile delivery framework [I-D.ietf-sipping-config-framework]
describes a rough data model with profile types that can correspond
to profile information related to the local-network, devices, users,
and applications. XCAP MAY be used as the profile delivery and
management mechanism for the UA profile delivery framework. Because
XCAP defines a specific hierarchy for how documents are organized, it
is necessary to discuss how that organization relates to the data
model described in the profile delivery framework.

When a user or device enrolls with a SUBSCRIBE request, the request
URI will contain identifying information for that user or device.
This identity is mapped to an XCAP User ID (XUID) based on an
implementation specific mapping. The "profile-type" along with the
"auid" Event header parameters specify the specific XCAP application
usage.

In particular, when the Event header parameter "profile-type" is
"application", the "auid" MAY be included to contain the XCAP
Application Unique ID (AUID). When the "profile-type" is
"application", but the "auid" parameter is absent, this specifies
that the user wishes to SUBSCRIBE to all documents for all
application usages associated with the user in the request-uri. This
provides a convenient way for a single subscription to be used to
obtain all application data. The XCAP root is determined by a local
mapping.

The "profile-type", "document" and "auid" can be thought of as
filters for determining which profile(s) are desired. When the
profile-type is "application" and the "document" and "auid"
parameters are not present, the XCAP operation to perform is to
select all AUID in the home and global paths. When the "auid" and/or
"document" parameters are present, the operation is to further filter
the profiles by the "auid" and/or "document" parameter values for the
global and user scope. If the filtering operations result in no
profiles being selected, a 404 response SHOULD be sent to the
SUBSCRIBE request.

Furthermore, when the "document" attribute is present and used with
XCAP, it identifies a specific document that is being requested. The
"document" attribute specifies a relative path from the XCAP root
[I-D.ietf-simple-xcap]. That is the "document" attribute is an XCAP
Document Selector expressed as a relative path to the XCAP root. The
"document" attribute MUST refer to a whole document.
   The "document" cannot refer to an XCAP query or path that results
   in a partial document. The reason for this is that changes which
   occur and get notified require the context of the full document to

be unambiguous. For example if the "document" path refered to an
element of an list, deleting the element from the list entirely is
indistiguishable from moving the order in which the element occurs
in the list.

5. Example Usage

For example, consider a phone with an instance ID of
urn:uuid:00000000-0000-0000-0000-0003968cf920. To obtain its device
profile, it generates a SUBSCRIBE that contains the following
Request-Line and Event header: (Note that line folding of the
Request-URI is illegal in SIP. The Request URI is shown broken
across the first 3-lines here only due to formatting limitations of
IETF documents.)


```
SUBSCRIBE
 sip:urn%3auuid%3a00000000-0000-0000-0000-0003968cf920@example.com
 SIP/2.0
Event: ua-profile;profile-type=device;Vendor="vendor2"
 ;Model="1";Version="2.2.2"
```

If the profile data is stored in an XCAP server, the profile delivery
server maps the "device" profile to an application usage and document
selector based on local policy. The user ID, in the case of a device
profile, could be the device ID which is identified in the user part
of the SUBSCRIBE URI. Assume the XCAP server uses an XCAP root
directory of: http://xcap.example.com/root. Local policy provides a
mapping for the AUID "vendor2-device-data" based upon the "vendor"
parameter and a document called "index" within the user directory,
the corresponding HTTP URI for the document would be: (Note that this
URI is only one line; it is split across lines due to formatting
limitations of IETF documents.)


```
http://xcap.example.com/root/vendor2-device-data/
 urn%3auuid%3a00000000-0000-0000-0000-0003968cf920/index
```

The returned user profile would typically specify the user identity
(as a SIP AOR) and his or her application-usages. From that, the
device can enroll to learn about its application data. To learn
about all of the data, the UA sends a subscription with the
application profile-type and no AUID:


```
SUBSCRIBE sip:alice@example.com SIP/2.0
Event: ua-profile;profile-type=application;Vendor="vendor2";
```

  Model="1";Version="2.2.2"

   The server maps the SIP Request URI to an XUI (alice, for example)
   and the xcap root based on local policy. If there are two AUIDs,
   "resource-lists" [I-D.ietf-simple-xcap-list-usage] and "rls-services"
   [I-D.ietf-simple-xcap-list-usage], this would result in a
   subscription to all documents within:

   http://xcap.example.com/root/rls-services/alice
   http://xcap.example.com/root/resource-lists/alice

   The user would not be subscribed to the global data for these two
   application usages, since that data is not important for users.

   However, the user/device could be made aware that it needs to
   subscribe to a specific document. In that case, its subscribe would
   look like:

   SUBSCRIBE sip:user-aor@example.com SIP/2.0
   Event: ua-profile;profile-type=application;auid="resource-lists";
    Vendor="vendor2";Model="1";Version="2.2.2"

   this would result in a subscription to the single global document for
   resource-lists.

   In some cases, these subscriptions are to a multiplicity of
   documents. In that case, the notification format will need to be one
   which can indicate what document has changed. This includes content
   indirection, but also the xcap diff format [I-D.ietf-simple-xcap-
   diff].

6. **Use of the XCAP Diff Format with ua-profile Event Package**

   The XCAP diff format [I-D.ietf-simple-xcap-diff] is meant to be used
   with an event package for the purposes of indicating changes in a
   document. This section provides guidelines for its usage with the
   us-profile or any event package defined for that purpose.

   Upon receipt of an initial SUBSCRIBE request, the client may have a
   cached version of some documents. However, the server does not know
   which instances of each document (where each instance is identified
   by an etag) the client currently posessses, if any. Indeed, upon
   initial startup, the client will not have any documents. The initial
   NOTIFY in this case MUST include a <document> element for each
   document associated with the subscription. The "previous-etag"

attribute MUST be absent, and the "new-etag" attribute MUST be
present and contain the entity tag for the current version of that
document resource. An XCAP diff document structured this way is
called a "reference" XCAP diff document. It establishes the baseline
etags and document URIs for the documents covered by the
subscription.

Upon receipt of this document, the client can determine whether its
local instance documents, if any, match the etags in the XCAP diff
document. If they do not match, the client SHOULD perform a
conditional GET for each document. The document URI is constructed
by appending the XCAP root in the "xcap-root" attribute of the <xcap-
diff> element to the escape coded "doc-selector" from each <document>
element. The request is made conditional by including an If-Match
header field, with the value of the etag from each <document>
element. So long as the documents haven't changed between the NOTIFY
and the GET, the client will obtain the reference versions that the
server will use for subsequent notifications.

If the conditional GET should fail, the client SHOULD wait for the
next NOTIFY and retry the conditional GET with the etag from the new
NOTIFY.
   This is because the the document was changed between the time that
   the NOTIFY was sent and the time that the GET was received. In
   this situation there should be another NOTIFY, for the change that
   occurred, with the newer etag that the client can use to try
   again.

Once the client has obtained the versions of the documents identified
in the reference XML diff, it can process NOTIFY requests on that
subscription. To process the NOTIFY requests, it makes sure that its
current version matches the version in the "previous-etag" attribute
of the <document> element. If not, the client can then fetch the
updated document from the server. If they do match, the client has
the most current version.

7. IANA Considerations

This specification registers two new Event header parameters and
updates the corresponding event package as defined in [RFC3265]. The
following information required for this registration:

   Package Name: ua-profile

      Published Document: RFC XXXX (Note to RFC Editor: Please fill in
      XXXX with the RFC number of this specification).
      Person to Contact: Daniel Petrie dan.ietf AT SIPez DOT com
      Additional SIP Event header parameters: document, auid
      The document and auid parameters do not have predefined values.
   The following table illustrates the additions to the IANA SIP Header
   Field Parameters and Parameter Values:

                                             Predefined
      Header Field   Parameter Name   Values   Reference
      --------------------------- --------------- ---------   ---------
      Event   document   No   [RFCXXXX]
      Event   auid   No   [RFCXXXX]

8. Security Considerations

   Profiles may contain sensitive data such as user credentials and
   personal information. The security considerations of this document
   are identical to those of the framework [I-D.ietf-sipping-config-
   framework]. Implementors should also carefully read the security
   considerations of XCAP [I-D.ietf-simple-xcap] as well.

   Subscribers implementing this specification MUST implement either
   HTTP or HTTPS. Subscribers also MUST implement the hash verification
   scheme described in SIP content indirection [I-D.ietf-sip-content-
   indirect-mech]. SIP profile delivery servers MUST implement both
   HTTP and HTTPS, and SHOULD implement a SIP Authentication Service as
   described in the SIP Identity mechanism [I-D.ietf-sip-identity]. All
   SIP entities are already required to implement SIP Digest
   authentication [RFC3261].

9. Acknowledgements

   Thanks to Rohan Mahy for editorial work on this document.

10. Change History

   [[RFC Editor: Please remove this entire section upon publication as
   an RFC.]]

10.1. Changes from draft-ietf-sipping-xcap-config-00.txt

Clarified constraint that "document" path MUST refer to a whole
document.
Moved "application" profile type definition here from:
draft-ietf-sipping-config-framework-08.
Defined concept of filtering using Event header parameters.
Cleaned up IANA section.
Profile delivery server SHOULD send 438 Invalid Event Parameter
Value for profile-type=application if XCAP is not supported. Also
reference draft-petrie-sip-event-param-err.
Moved section on xcap-diff usage in an event package from xcap-
diff to this document.

11. Normative References

[I-D.dpetrie-sip-event-param-err]
          Petrie, D., "Session Initiation Protocol Response Code for
          Invalid Event Parameter Values",
          draft-petrie-sip-event-param-err-00 (work in progress),
          October 2006.

[I-D.ietf-simple-xcap]
          Rosenberg, J., "The Extensible Markup Language (XML)
          Configuration Access Protocol (XCAP)",
          draft-ietf-simple-xcap-11 (work in progress), May 2006.

[I-D.ietf-simple-xcap-diff]
          Rosenberg, J., "An Extensible Markup Language (XML)
          Document Format for Indicating A Change in XML
          Configuration Access Protocol (XCAP) Resources",
          draft-ietf-simple-xcap-diff-03 (work in progress),
          October 2006.

[I-D.ietf-simple-xcap-list-usage]
          Rosenberg, J., "Extensible Markup Language (XML) Formats
          for Representing Resource Lists",
          draft-ietf-simple-xcap-list-usage-05 (work in progress),
          February 2005.

[I-D.ietf-sip-content-indirect-mech]
          Burger, E., "A Mechanism for Content Indirection in
          Session Initiation Protocol (SIP) Messages",
          draft-ietf-sip-content-indirect-mech-05 (work in
          progress), October 2004.

[I-D.ietf-sip-identity]
          Peterson, J. and C. Jennings, "Enhancements for
          Authenticated Identity Management in the Session
          Initiation Protocol (SIP)", draft-ietf-sip-identity-06

              (work in progress), October 2005.

   [I-D.ietf-sipping-config-framework]
              Petrie, D., "A Framework for Session Initiation Protocol
              User Agent Profile Delivery",
              draft-ietf-sipping-config-framework-09 (work in progress),
              October 2006.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
              Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext
              Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

   [RFC3261] 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.

   [RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific
              Event Notification", RFC 3265, June 2002.

Author's Address

    Daniel Petrie
    SIPez LLC.
    34 Robbins Rd
    Arlington, MA 02476
    US

    Phone: "+1 617 273 4000
    Email: dan.ietf AT SIPez DOT com
    URI: http://www.SIPez.com/index.html?id=xcap-config