

SIMPLE
Internet-Draft
Intended status: Standards Track
Expires: February 18, 2008

J. Rosenberg
Cisco
J. Urpalainen
Nokia
August 17, 2007

An Extensible Markup Language (XML) Document Format for Indicating A
Change in XML Configuration Access Protocol (XCAP) Resources
draft-ietf-simple-xcap-diff-06

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/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 February 18, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This specification defines a document format that can be used to indicate that a change has occurred in a document managed by the Extensible Markup Language (XML) Configuration Access Protocol (XCAP). This format indicates the document that has changed and its former and new entity tags. It also can indicate the specific change that was made in the document, using an XML patch format. This

Internet-Draft

XCAP Diff Format

August 2007

format allows also indications of element and attribute content of an XML document. XCAP diff documents can be delivered to clients using a number of means, including a Session Initiation Protocol (SIP) event package.

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Structure of an XCAP Diff Document	4
4.	XML Schema	7
5.	Example Document	10
6.	Security Considerations	10
7.	IANA Considerations	11
7.1.	application/xcap-diff+xml MIME Type	11
7.2.	URN Sub-Namespace Registration for urn:ietf:params:xml:ns:xcap-diff	12
7.3.	Schema Registration	12
8.	Acknowledgments	13
9.	References	13
9.1.	Normative References	13
9.2.	Informative References	14
	Authors' Addresses	14
	Intellectual Property and Copyright Statements	15

Internet-Draft

XCAP Diff Format

August 2007

1. Introduction

The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) [9] is a protocol that allows clients to manipulate XML documents stored on a server. These XML documents serve as configuration information for application protocols. As an example, resource list [13] subscriptions (also known as presence lists) allow a client to have a single SIP subscription to a list of users, where the list is maintained on a server. The server will obtain presence for those users and report it back to the client. This application requires the server, called a Resource List Server (RLS), to have access to the list of presentities. This list needs to be manipulated by clients so they can add and remove their friends as they desire.

Complexities arise when multiple clients attempt to simultaneously manipulate a document, such as a presence list. Frequently, a client will keep a copy of the current list in memory, so it can render it to users. However, if another client modifies the document, the cached version becomes stale. This modification event must be made known to all clients which have cached copies of the document, so that they can fetch the most recent one.

To deal with this problem, clients can use a Session Initiation Protocol (SIP) [11] event package [12] to subscribe to change events in XCAP documents. This notification needs to indicate the specific resource that changed, and how it changed. One solution for the format of such a change notification would be a content indirection object [15]. Though content indirection can tell a client that a document has changed, it provides it with MIME Content-ID indicating the new version of the document. The MIME Content-ID is not the same as the entity tag, which is used by XCAP for document versioning. As such, a client cannot easily ascertain whether an indication of a change in a document is due to a change it just made, or due to a change another client made at around the same time. Furthermore, content indirections don't indicate how a document changed; they

would only be able to indicate that it did change.

To resolve these problems, this document defines a data format which can convey the fact that an XML document managed by XCAP has changed. This data format is an XML document format, called an XCAP diff document. This format can indicate that a document has changed, and provide its previous and new entity tags. It can also optionally include a set of patch operations [[10](#)], which indicate how to transform the document from the version prior to the change, to the version after it. XML element and attribute content of XCAP documents can also be delivered with this format.

XML documents that are equivalent for the purposes of many applications may differ in their physical representation. Similar to XCAP, the canonical form with comments [[1](#)] of an XML document determines the logical equivalence when this format is used to patch XML documents.

[2](#). Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [[8](#)] and indicate requirement levels for compliant implementations.

This specification also defines the following additional terms:

Document: When the term document is used without the "XCAP diff" in front of it, it refers to the XCAP document resource about whom the XCAP diff document is reporting a change.

XCAP diff document: The XML document defined by this specification that reports on a set of changes in an XCAP document resource.

Server: Typically an XCAP server, this is a protocol entity that generates XCAP diff documents based on its knowledge of a set of XCAP documents.

Client: Typically an XCAP client and SIP User Agent (UA), the client consumes XCAP diff documents in order to reconstruct the document

stored on the server.

3. Structure of an XCAP Diff Document

An XCAP diff document is an XML [2] document that MUST be well-formed and SHOULD be valid. XCAP diff documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying XCAP diff documents and document fragments. The namespace URI for elements defined by this specification is a URN [4], using the namespace identifier 'ietf' defined by [6] and extended by [7]. This URN is:

urn:ietf:params:xml:ns:xcap-diff

An XCAP diff document begins with the root element tag <xcap-diff>. This element has a single mandatory attribute, "xcap-root". The value of this attribute is the XCAP root URI for the documents in which the changes have taken place. A single XCAP diff document can

only represent changes in documents within the same XCAP root. The content of the <xcap-diff> element is an unordered sequence of <document>, <element> and <attribute> elements followed by any number of elements from other namespaces for the purposes of extensibility. Any such unknown elements MUST be ignored by the client. Each <document> element specifies changes in a specific document within the XCAP root. It has one mandatory attribute, "sel", and a two optional attributes, "new-etag" and "previous-etag". The "sel" attribute of the <document> element identifies the specific document within the XCAP root for which changes are indicated. Its content MUST be a relative path reference, with the base URI being equal to the XCAP root URI. The "new-etag" attribute provides the entity tag (ETag) for the document after the application of the changes, assuming the document exists after those changes. The "previous-etag" attribute provides an identifier for the document instance prior to the change. If the change being reported is the removal of a document, the "previous-etag" MUST only be included and the "new-etag" attribute will not be present. The "new-etag" attribute MUST only exist alone when the document either exists or it was just created (no patch included). Both attributes are present when a patch (or series of XCAP operations) has been applied to the resource. Also both attributes MAY be used to indicate an ETag

change without any document modifications (patches).

The "previous-etag" and "new-etag" need not have been sequentially assigned ETags at the server. An XCAP diff document can indicate changes that have occurred over a series of XCAP operations. The only requirement then is that, the sequence of events, when executed serially, will result in the transformation of the document with the ETag "previous-etag" to the one whose ETag is "new-etag". Also the series of operations do not have to be the same exact series of operations that occurred at the server.

Each <document> element contains either a sequence of patching instructions or an indication that the body hasn't semantically changed. The latter means that the document has been assigned a new ETag but its content is unchanged and it is indicated by the <body-not-changed> element. Patching instructions are described by the <add>, <replace> and <remove> elements. These elements use the corresponding add, replace and remove types defined in [10], and define a set of patch operations that can be applied to transform the document. See [10] for instructions on how this transformation is effected. The <document> element can also contain elements from other namespaces for the purposes of extensibility. Any unknown elements MUST be ignored.

Figure 1 shows <document> element content and how corresponding resource or metadata changes. An external document retrieval means in practice an HTTP GET requests for a relevant resource.

previous- etag	new- etag	<add> <replace> <remove>	<body- not- changed>	XCAP resource/ metadata change
xxx	yyy	*	-	resource patched, patch included
xxx	yyy	-	-	resource patched, external document

				retrieval
xxx	yyy	-	*	only ETag changed
-	yyy	-	-	resource created or exists, external document retrieval
xxx	-	-	-	resource removed

Figure 1: <document> element content / corresponding resource changes

Each <element> element indicates the existing element content of an XCAP document. It has one mandatory attribute, "sel", and one optional attribute, "exists". The "sel" attribute of the <element> element identifies an XML element of an XCAP document. It is a percent encoded relative URI following XCAP conventions when selecting elements. The XCAP Node Selector MUST always locate a unique node, the "exists" attribute thus shows whether an element exists or not in the XCAP document. When the "exists" attribute is absent from the <element> element, the indicated element still exists in the XCAP document. The located result element exists as a child element of the <element> element. It should be noted, that only the full content of an element is shown if it exists, there are no conventions for patching these elements. In a corner case where the content of this element cannot be presented for some reason, although it exists in the XCAP document, the <element> element MUST not have any child nodes.

As the result XML element is typically namespace qualified, all

needed namespace declarations MUST exist within the <xml-diff> document. The possible local namespace declarations within the result element exist unmodified as in the source document, similar to XCAP conventions. Other namespace references MUST be resolved from the context of the <element> or its parent elements. The prefixes of qualified names (QName) [3] of XML nodes also remain as they exist originally in the source XCAP document.

Each <attribute> element indicates the existing attribute content of an XCAP document. It has one mandatory attribute, "sel", and one optional attribute, "exists". The "sel" attribute of the <attribute> element identifies an XML attribute of an XCAP document. It is a percent encoded relative URI following XCAP conventions when selecting attributes. The "exists" attribute indicates whether an attribute exists or not in the XCAP document. When the "exists" attribute is absent from the <attribute> element, the indicated attribute exists in the XCAP document. The child text node of the <attribute> element indicates the value of the located attribute.

[4.](#) XML Schema

The XML Schema for the XCAP diff format.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:xcap-diff"
  targetNamespace="urn:ietf:params:xml:ns:xcap-diff"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- include patch-ops -->
  <xs:include
    schemaLocation="urn:ietf:params:xml:ns:xcap-diff:patch-ops"/>

  <!-- document root -->
  <xs:element name="xcap-diff">
    <xs:complexType>
      <xs:sequence minOccurs="0">

        <xs:sequence minOccurs="0" maxOccurs="unbounded">
          <xs:choice>
            <xs:element name="document" type="documentType"/>
            <xs:element name="element" type="elementType"/>
            <xs:element name="attribute" type="attributeType"/>
          </xs:choice>
        </xs:sequence>


```

```
<xs:any namespace="##other" processContents="lax">
```



```

        minOccurs="0" maxOccurs="unbounded"/>

    </xs:sequence>
    <xs:attribute name="xcap-root" type="xs:anyURI" use="required"/>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
</xs:element>

<!-- xcap document type -->
<xs:complexType name="documentType">

    <xs:choice minOccurs="0">
        <xs:element name="body-not-changed" type="emptyType"/>

        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:choice>

                <xs:element name="add">
                    <xs:complexType>
                        <xs:complexContent>
                            <xs:extension base="add">
                                <xs:anyAttribute processContents="lax"/>
                            </xs:extension>
                        </xs:complexContent>
                    </xs:complexType>
                </xs:element>

                <xs:element name="remove">
                    <xs:complexType>
                        <xs:complexContent>
                            <xs:extension base="remove">
                                <xs:anyAttribute processContents="lax"/>
                            </xs:extension>
                        </xs:complexContent>
                    </xs:complexType>
                </xs:element>

                <xs:element name="replace">
                    <xs:complexType>
                        <xs:complexContent>
                            <xs:extension base="replace">
                                <xs:anyAttribute processContents="lax"/>
                            </xs:extension>
                        </xs:complexContent>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:choice>
</xs:complexType>

```

```
    <xs:any namespace="##other" processContents="lax"/>
  </xs:choice>
</xs:sequence>

</xs:choice>

<xs:attribute name="sel" type="xs:anyURI" use="required"/>
<xs:attribute name="new-etag" type="xs:string" use="optional"/>
<xs:attribute name="previous-etag" type="xs:string" use="optional"/>
<xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>

<!-- xcap element type -->
<xs:complexType name="elementType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:sequence>
        <xs:any processContents="lax" namespace="##any"
          minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="sel" type="xs:string"
        use="required"/>
      <xs:attribute name="exists" type="xs:boolean"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<!-- xcap attribute type -->
<xs:complexType name="attributeType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="sel" type="xs:string"
        use="required"/>

      <xs:attribute name="exists" type="xs:boolean"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- empty type -->
<xs:complexType name="emptyType"/>

</xs:schema>
```

5. Example Document

The following is an example of a document compliant to the schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
           xcap-root="http://xcap.example.com/root/">

  <document new-etag="7ahggs"
            sel="resource-lists/users/sip:joe@example.com/coworkers"
            previous-etag="8a77f8d"/>

  <d:element sel="rls-services/users/sip:joe@example.com/index/~~
/*/service%5b@uri='sip:marketing@example.com'%5d"
            xmlns:d="urn:ietf:params:xml:ns:xcap-diff"
            xmlns="urn:ietf:params:xml:ns:rls-services"
            xmlns:rl="urn:ietf:params:xml:ns:resource-lists"
            ><service uri="sip:marketing@example.com">
      <list name="marketing">
        <rl:entry uri="sip:joe@example.com"/>
        <rl:entry uri="sip:sudhir@example.com"/>
      </list>
      <packages>
        <package>presence</package>
      </packages>
    </service></d:element>

  <attribute
    sel="rls-services/users/sip:joe@example.com/index/~~/*/service/@uri"
    >sip:marketing@example.com</attribute>

</xcap-diff>
```

This indicates that the document with URI "http://xcap.example.com/root/resource-lists/users/sip:joe@example.com/coworkers" has changed. Its previous entity tag is "8a77f8d" and its new one is "7ahggs" but actual changes are not shown. The <service> element exists in the rls-services "index" document and its full content is shown. Note

that the <service> element is attached with a default namespace declaration within the original document. Similarly, a "uri" attribute content is shown from the same "index" document as an illustrative example.

[6.](#) Security Considerations

XCAP diff documents can include changes from one document to another.

As a consequence, if the document itself is sensitive and requires confidentiality, integrity or authentication, then the same applies to the XCAP diff format. Therefore, protocols which transport XCAP diff documents must provide sufficient security capabilities for transporting the document itself.

[7.](#) IANA Considerations

There are several IANA considerations associated with this specification.

[7.1.](#) application/xcap-diff+xml MIME Type

MIME media type name: application

MIME subtype name: xcap-diff+xml

Mandatory parameters: none

Optional parameters: Same as charset parameter application/xml as specified in [RFC 3023](#) [5].

Encoding considerations: Same as encoding considerations of application/xml as specified in [RFC 3023](#) [5].

Security considerations: See [Section 10 of RFC 3023](#) [5] and [Section 6](#) of RFCXXXX [[NOTE TO RFC-EDITOR/IANA: Please replace XXXX with the RFC number of this specification.]].

Interoperability considerations: none.

Published specification: This document.

Applications which use this media type: This document type has been used to support manipulation of resource lists [\[14\]](#) using XCAP.

Additional Information:

Magic Number: None

File Extension: .xdf

Macintosh file type code: "TEXT"

Rosenberg & Urpalainen Expires February 18, 2008

[Page 11]

Internet-Draft

XCAP Diff Format

August 2007

Personal and email address for further information: Jonathan Rosenberg, jdrosen@jdrosen.net

Intended usage: COMMON

Author/Change controller: The IETF.

[7.2.](#) URN Sub-Namespace Registration for `urn:ietf:params:xml:ns:xcap-diff`

This section registers a new XML namespace, as per the guidelines in [\[7\]](#)

URI: The URI for this namespace is
`urn:ietf:params:xml:ns:xcap-diff`.

Registrant Contact: IETF, SIMPLE working group, (simple@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
```

```

        "http://www.w3.org/TR/xhtml1-basic/xhtml1-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>XCAP Diff Namespace</title>
</head>
<body>
  <h1>Namespace for XCAP Diff</h1>
  <h2>urn:ietf:params:xml:ns:xcap-diff</h2>
  <p>See <a href="[URL of published RFC]">RFCXXXX[[NOTE
TO IANA/RFC-EDITOR: Please replace XXXX with the RFC number of this
specification.]]</a>.</p>
</body>
</html>
END

```

[7.3.](#) Schema Registration

This section registers a new XML schema per the procedures in [\[7\]](#).

URI: urn:ietf:params:xml:ns:xcap-diff

Registrant Contact: IETF, SIMPLE working group, (simple@ietf.org),
Jonathan Rosenberg (jdrosen@jdrosen.net).

The XML for this schema can be found as the sole content of
[Section 4](#).

[8.](#) Acknowledgments

The authors would like to thank Pavel Dostal, Jeroen van Bommel,
Martin Hynar and Anders Lindgren for their valuable comments.

[9.](#) References

[9.1.](#) Normative References

- [1] "Canonical XML 1.0", W3C Recommendation REC-xml-c14n-20010315 , March 2001.
- [2] "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation REC-xml-20060816 , August 2006.
- [3] "Namespaces in XML (Second Edition)", W3C Recommendation REC-xml-names-20060816 , August 2006.
- [4] Moats, R., "URN syntax", [RFC 2141](#), May 1997.
- [5] Murata, M., "XML media types", [RFC 3023](#), January 2001.
- [6] Moats, R., "A URN namespace for IETF documents", [RFC 2648](#), Aug. 1999.
- [7] Mealling, M., "The IETF XML Registry", [RFC 3688](#), [BCP 81](#), January 2004.
- [8] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [9] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", [RFC 4825](#), May 2007.
- [10] Urpalainen, J., "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors", [draft-ietf-simple-xml-patch-ops-03](#), August 2007.

[9.2.](#) Informative References

- [11] 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.
- [12] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [13] Roach, A., Campbell, B., and J. Rosenberg, "A Session Initiation Protocol (SIP) Event Notification Extension for

Resource Lists", [RFC 4662](#), August 2006.

- [14] Rosenberg, J., "Extensible Markup Language (XML) Formats for Representing Resource Lists", [RFC 4826](#), May 2007.
- [15] Burger, E., Ed., "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", [RFC 4483](#), May 2006.

Authors' Addresses

Jonathan Rosenberg
Cisco
Edison, NJ
US

Email: jdrosen@cisco.com
URI: <http://www.jdrosen.net>

Jari Urpalainen
Nokia
Itamerenkatu 11-13
Helsinki 00180
Finland

Phone: +358 7180 37686
Email: jari.urpalainen@nokia.com

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.

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, THE IETF TRUST 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.

Intellectual Property

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.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).