

SIMPLE
Internet-Draft
Expires: April 25, 2006

J. Rosenberg
Cisco Systems
October 22, 2005

An Extensible Markup Language (XML) Document Format for Indicating A
Change in XML Configuration Access Protocol (XCAP) Resources
draft-ietf-simple-xcap-diff-02

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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

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

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/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 April 25, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

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. XCAP diff documents can be delivered to clients using a number of means, including the Session Initiation Protocol (SIP) event package for configuration data. By subscribing to this event package, clients can learn about document changes made

Internet-Draft

XCAP Diff Format

October 2005

by other clients. The XCAP diff format is extensible, so that additional information, such as a description of the actual change, can be included.

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Structure of an XCAP Diff Document	4
4.	XML Schema	6
5.	Example Document	6
6.	Usage with the Config Framework	7
7.	Security Considerations	8
8.	IANA Considerations	8
8.1	application/xcap-diff+xml MIME Type	8
8.2	URN Sub-Namespace Registration for urn:ietf:params:xml:ns:xcap-diff	9
8.3	Schema Registration	10
9.	References	10
9.1	Normative References	10
9.2	Informative References	11
	Author's Address	12
	Intellectual Property and Copyright Statements	13

1. Introduction

The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) [8] 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 [12] 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 the Session Initiation Protocol (SIP) [10] event package [11] for subscribing to changes in configuration and profile information [9], including application data that resides on an XCAP server. With that package, a user gets notified that a particular document has changed. This notification can include the full content of the new document, or it can be a content indirection [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.

In addition, when an XCAP client inserts a new element or attribute into an existing document, the client has no way to know whether the insertion was done against its cached version of the document. The reasons for this are described in [Section 7.10](#) of XCAP. To help a client ascertain whether this has occurred after performing the insertion, the XCAP response needs to contain a document which indicates the entity tags before and after the document was modified.

To resolve these problems, this document defines a data format which can convey the fact that an XML document 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. This specification also explains how this format is used in conjunction with the configuration profile framework.

[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](#) [7] 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) that acts as a subscriber to the configuration event package, this is a protocol entity that 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 [3], using the namespace identifier 'ietf' defined by [5] and extended by [6]. 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 a sequence of <document> elements. Each <document> element specifies changes in a specific document within the XCAP root. It has one mandatory attribute, "doc-selector", and a three optional attributes, "new-etag", "previous-etag" and "hash". The "doc-selector" 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 etag for the document after the application of the changes, assuming the document exists after those changes. If the change being reported is the deletion of the document, the "new-etag" attribute will not be present. A server MUST include the "new-etag" unless the document does not exist subsequent to the changes reported in the XCAP diff document. The "previous-etag" attribute provides an identifier for the document instance prior to the change. If the document did not exist prior to the change (that is, the change was the creation of the document), the "previous-etag" is not present.

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 optional "hash" attribute provides an HMAC of the document instance whose etag is "new-etag", once that document is represented in canonical form. To compute this value, the server MUST apply the mandatory XML canonicalization defined in the Canonical XML 1.0 [1] specification, and then computes an HMAC [13] using SHA1 over this canonical document, with a key whose value is 0x2238a. The result is the value of the "hash" attribute. This attribute is optional, and a server MAY elect not to include it. Even if present, a client MAY elect to ignore it.

This contents of the <document> element are extensible, and can include elements from other namespaces. It is anticipated that extensions would be defined that allow the actual change in the document to be reported.

[4.](#) XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:xcap-diff"
  xmlns="urn:ietf:params:xml:ns:xcap-diff"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="document">
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="doc-selector" type="xs:anyURI" use="required"/>
      <xs:attribute name="new-etag" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    <xs:attribute name="previous-etag" type="xs:string" use="optional"/>
    <xs:attribute name="hash" type="xs:string" use="optional"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
</xs:element>
<xs:element name="xcap-diff">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="document"/>
    </xs:sequence>
    <xs:attribute name="xcap-root" type="xs:anyURI" use="required"/>
  </xs:complexType>
</xs:element>
</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"
    doc-selector="resource-lists/users/joe/coworkers"
    previous-etag="8a77f8d"/>
</xcap-diff>

```

This indicates that the document with URI `http://xcap.example.com/root/resource-lists/users/joe/coworkers` has changed. Its previous entity tag is `8a77f8d` and its new one is

7ahggs.

6. Usage with the Config Framework

The framework for user agent profile delivery [9] defines an event package which can be used to subscribe to user, device, application or local-network data that defines the configuration of a client. This data can be present in an XCAP server. Normally, content

indirection [15] will be used as the NOTIFY body format, to indicate the specific document that has changed, and should be re-fetched. However, if the client includes an Accept header field including the MIME type "application/xcap-diff+xml", the server has the option of returning documents in this format instead.

When the client performs an initial subscription, the rules in [9] are used to select the set of documents which the subscription applies to. Upon initial subscription, the server does not know which instances of each document (where each instance is identified by an etag) the client currently possesses, if any. Indeed, upon 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 generate a SUBSCRIBE refresh request to trigger a new NOTIFY. The server will always generate a "reference" XML diff document on receipt of a SUBSCRIBE refresh. This establishes a new set of baseline etags, and the client can then attempt to do another fetch. It is anticipated that future extensions to the profile delivery framework will allow a client to include, in its SUBSCRIBE request, an indicator of the current version of the documents it holds. That would obviate the

need for a potentially never-ending stream of SUBSCRIBE/GET sequences

should the documents be rapidly changing, for some reason.

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.](#) Security Considerations

XCAP diff documents are not very sensitive; they only contain entity tags and the URI for documents. An attacker that is able to examine such a document cannot access or modify the referenced document unless it has also managed to attack XCAP itself. Thus, there is no requirement for message confidentiality. However, an attacker that can modify XCAP diff documents in transit could fool a client into thinking that a document hasn't changed, when it has, or vice-versa. Therefore, protocols which transport XCAP Diff documents SHOULD provide message integrity.

[8.](#) IANA Considerations

There are several IANA considerations associated with this specification.

[8.1](#) 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](#) [4].

Encoding considerations: Same as encoding considerations of application/xml as specified in [RFC 3023](#) [4].

Security considerations: See [Section 10 of RFC 3023](#) [4] and [Section 7](#) 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"

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

Intended usage: COMMON

Author/Change controller: The IETF.

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

This section registers a new XML namespace, as per the guidelines in [\[6\]](#)

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:

Internet-Draft

XCAP Diff Format

October 2005

```
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>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
```

[8.3](#) Schema Registration

This section registers a new XML schema per the procedures in [\[6\]](#).

URI: urn:ietf:params:xml:schema: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](#).

[9](#). References

[9.1](#) Normative References

- [1] Boyer, J., "Canonical XML Version 1.0", W3C REC REC-xml-c14n-

20010315, March 2001.

- [2] Bray, T., Paoli, J., Sperberg-McQueen, C., and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C FirstEdition REC-xml-20001006, October 2000.
- [3] Moats, R., "URN Syntax", [RFC 2141](#), May 1997.
- [4] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types",

Rosenberg

Expires April 25, 2006

[Page 10]

Internet-Draft

XCAP Diff Format

October 2005

[RFC 3023](#), January 2001.

- [5] Moats, R., "A URN Namespace for IETF Documents", [RFC 2648](#), August 1999.
- [6] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), January 2004.
- [7] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [8] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", [draft-ietf-simple-xcap-07](#) (work in progress), June 2005.
- [9] Petrie, D., "A Framework for Session Initiation Protocol User Agent Profile Delivery", [draft-ietf-sipping-config-framework-07](#) (work in progress), July 2005.

[9.2](#) Informative References

- [10] 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.
- [11] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [12] Roach, A., Rosenberg, J., and B. Campbell, "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", [draft-ietf-simple-event-list-07](#) (work in progress), January 2005.

- [13] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), February 1997.
- [14] Rosenberg, J., "Extensible Markup Language (XML) Formats for Representing Resource Lists", [draft-ietf-simple-xcap-list-usage-05](#) (work in progress), February 2005.
- [15] 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.

Rosenberg

Expires April 25, 2006

[Page 11]

Internet-Draft

XCAP Diff Format

October 2005

Author's Address

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07054
US

Phone: +1 973 952-5000
Email: jdrosen@cisco.com
URI: <http://www.jdrosen.net>

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

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