

SIMPLE
Internet-Draft
Expires: August 15, 2005

J. Rosenberg
Cisco Systems
February 14, 2005

An Extensible Markup Language (XML) Document Format for Indicating
Changes in XML Configuration Access Protocol (XCAP) Resources
draft-ietf-simple-xcap-diff-00

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of [section 3 of RFC 3667](#). 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 become aware will be disclosed, in accordance with [RFC 3668](#).

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 August 15, 2005.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This specification defines a document format that can be used to describe the differences between versions of resources managed by the Extensible Markup Language (XML) Configuration Access Protocol (XCAP). Documents of this format 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 by other

Internet-Draft

XCAP Diff Format

February 2005

clients.

Table of Contents

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

1. Introduction

The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) [7] 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 information 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 [8], 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]. However, in both cases, the transfer of the entire document is ultimately required. This may require a lot of bandwidth, particularly for wireless devices with large documents (such as a resource list [12] with hundreds of users listed).

To resolve this problem, this document defines a data format which

can convey changes in XML documents managed by an XCAP server. This data format is an XML document format, called an XCAP diff document. The XCAP diff document is based on a generic format for XML patch operations [9]. This specification also explains how this format is used in conjunction with the configuration profile framework.

2. Structure of an XCAP Diff Document

An XCAP diff document is an XML [2] document that MUST be well-formed and SHOULD be valid. XML-change documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying xml-change 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 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 three mandatory attributes - "new-etag", "previous-etag" and "doc-selector", and a single optional attribute, "hash". The "doc-selector" identifies the specific document within the XCAP root for which changes are included. Its content MUST be a relative path reference, with the base URL being equal to the XCAP root URL. The "previous-etag" and "new-etag" provide identifiers for the document instance before the change, and then after the change. These need not have been sequentially assigned etags at the server. An XCAP diff document can describe changes that have occurred over a series of XCAP operations.

The optional "hash" attribute provides an HMAC of the new document, represented in canonical form. See [Section 5](#) for details on how this value is computed. This attribute is optional, and a server can elect not to include it.

Each <document> element is followed by a series of operations, which if followed by the client, will convert the document whose etag is "previous-etag" into the one whose etag is "new-etag". These are the three XML patch operations, <add>, <remove> and <replace> defined in [9].

It is possible for the list of instructions for a <document> to be empty. In that case, the entity tag in the "new-etag" may equal the entity tag in the "previous-etag". These entity tags may differ in the event that the document has changed entity tags, but its content has not been altered.

[3.](#) XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:xcap-diff"
  xmlns:patch="urn:ietf:params:xml:ns:xml-patch"
  xmlns:tns="urn:ietf:params:xml:ns:xcap-diff"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="urn:ietf:params:xml:ns:xml-patch"/>
  <xs:element name="xcap-diff">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="document" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
              <xs:choice>
                <xs:element ref="patch:add"/>
                <xs:element ref="patch:remove"/>
                <xs:element ref="patch:replace"/>
              </xs:choice>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    </xs:sequence>
    <xs:attribute name="doc-selector" type="xs:anyURI" use="required"/>
    <xs:attribute name="new-etag" type="xs:string" use="required"/>
    <xs:attribute name="previous-etag" type="xs:string" use="required"/>
    <xs:attribute name="hash" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="xcap-root" type="xs:anyURI" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

[4.](#) 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"
  xmlns:patch="urn:ietf:params:xml:ns:xml-patch"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:rl="urn:ietf:params:xml:ns:resource-lists"
  xcap-root="http://xcap.example.com/root">
  <document doc-selector="resource-lists/users/joe/friends"
    new-etag="7hahsd" previous-etag="7hahsd"/>
  <document doc-selector="resource-lists/users/joe/coworkers"
    new-etag="ffds66a" previous-etag="xkkkaisu">
    <patch:add parent="rl:resource-lists/rl:list[@name='&quot;l1&quot;']"
      sel="*[1]">
      <rl:entry uri="sip:new-worker@example.com"/>
    </patch:add>
  </document>
</xcap-diff>

```

</document>
</xcap-diff>

5. Usage with the Config Framework

The framework for user agent profile delivery [8] defines an event package which can be used to subscribe to user, device, application or local-network data. 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 [8] are used to select the set of documents which the subscription applies to. Upon initial subscription, the server does not know which instance (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 content of each of those <document> elements MUST be empty. The "previous-etag" and "new-etag" attributes MUST be identical, and contain the entity tag for the current version of that resource. An XML diff document structured this way is called a "reference" XML 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. It then follows the list of instructions, in order, for that <document> as defined in [9].

Once the client has finished applying the instructions to the document, it should end up with the same document the server has. To verify this, the client applies the mandatory XML canonicalization defined in the Canonical XML 1.0 [1] specification, and computes an HMAC [13] using SHA1 over this canonical document, with a key whose value is 0x2238a. The resulting string is compared with the "hash" attribute of the <document> element. If they match, the client can be sure that it has the most up to date version. If they don't match, the client MUST flush its current version of the document from memory. It can then obtain a new XML diff reference by sending a SUBSCRIBE refresh request on the dialog.

Of course, this mechanism for computing the most current document from the hash is optional. A client can elect to ignore the information on what changed and simply fetch the most recent document every time it gets a change indication where the new version is not the same as the one cached by the client. Furthermore, the server may elect to not send the hash, in which case this check cannot be made.

[6.](#) Security Considerations

XCAP diff documents contain the same information in the documents whose differences they describe. As such, the security considerations associated with those documents apply to XCAP diff documents.

[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](#) [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 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"

Internet-Draft

XCAP Diff Format

February 2005

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 [\[6\]](#)

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:

```
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
```

[7.3](#) Schema Registration

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

Rosenberg

Expires August 15, 2005

[Page 9]

Internet-Draft

XCAP Diff Format

February 2005

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 3](#).

[8](#). References

[8.1](#) 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", [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] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", [draft-ietf-simple-xcap-05](#) (work in progress), November 2004.
- [8] Petrie, D., "A Framework for Session Initiation Protocol User

Agent Profile Delivery", [draft-ietf-sipping-config-framework-05](#) (work in progress), November 2004.

- [9] Urpalainen, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Patch Operations", Internet Draft [draft-urpalainen-simple-xcap-patch-ops-00.txt](#), February 2005.

[8.2](#) Informative References

- [10] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP:

Rosenberg Expires August 15, 2005 [Page 10]

Internet-Draft XCAP Diff Format February 2005

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-04](#) (work in progress), October 2004.
- [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.

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>

Rosenberg

Expires August 15, 2005

[Page 11]

Internet-Draft

XCAP Diff Format

February 2005

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.