XML Document Management (XDM) Specification

Candidate Version 1.0 – 15 Apr 2005

Open Mobile Alliance

OMA-TS-XDM_Core-V1_0-20050415-C

Contents

Figures

1. Scope

This document specifies common protocols, data access conventions, common data application usages and two entities that are needed to provide XDM services to other enablers. Such enablers can utilize this specification to support any required application-specific usages.

2. References

2.1 Normative References

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL:http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt

[RFC2617] "HTTP Authentication: Basic and Digest Access Authentication", Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, RFC 2617, June 1999. URL:http://www.ietf.org/rfc/rfc2617.txt

[RFC2616] "Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding, June 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

[RFC2818] "HTTP Over TLS", Rescorla, E., RFC 2818, May 2000. URL: http://www.ietf.org/rfc/rfc2818.txt

[RFC3265] "Session Initiation Protocol (SIP)-Specific Event Notification", A. B. Roach, June 2002. URL: http://www.ietf.org/rfc/rfc3265.txt

[RFC3986] "Uniform Resource Identifier (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, January 2005, http://www.ietf.org/rfc/rfc3986.txt

[XDM RD] "XML Document Management Requirements", Version 1,0, Open Mobile Alliance™, OMA-RD-XDM-V1_0-, URL:http://ww.openmobilealliance.org/

[XCAP] "The Extensible Markup Language (XML) Configuration Access protocol (XCAP)", J. Rosenberg, February 7, 2005, URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-06.txt
Note: Work in progress

[XCAP_Diff] "An Extensible Markup Language (XML) Document Format for Indicating Changes in XML Configuration Access Protocol (XCAP) Resources", J. Rosenberg, July 18, 2004, URL: http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-package-02.txt
Note: Work in progress

[SIP_UA_Prof] "A Framework for Session Initiation Protocol User Agent Profile Delivery", D. Petrie, October 2004. URL: http://www.ietf.org/internet-drafts/draft-ietf-sipping-config-framework-05.txt
Note: Work in progress

[INDIRMECH] "Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", E. Burger, Ed., October 2004.
URL: http://www.ietf.org/internet-drafts/draft-ietf-sip-content-indirect-mech-05.txt

Note: Work in progress

[3GPP TS 33.222] 3GPP TS 33.222 "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) (Release 6)", URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.222/

[OMA-SyncML-DMStdObj-V1-1-2] "SyncML Device Management Standardized Objects", version 1.1.2, Open Mobile Alliance™, OMA-SyncML-DMStdObj-V1_1_2, URL:http://www.openmobilealliance.org/.

[OMA-SyncML-DMTND-V1-1-2] "SyncML Device Management Tree and Description", version 1.1.2, Open Mobile Alliance™, OMA-SyncML-DMTND-V1_1_2, URL:http://www.openmobilealliance.org/.

[OMA DM] OMA Device Management, V1.1.2 (based on SyncML DM), OMA-DM-V1_1_2
URL:http://www.openmobilealliance.com/

[Provisioning Content] OMA – Provisioning Content V1.1, OMA-DM_ProvCont-V1_2_0
URL:http://www.openmobilealliance.com/

[3GPP TS 24.109] 3GPP TS 24.109 "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details (Release 6)" . URL: http://www.3gpp.org/ftp/Specs/archive/24_series/24.109/

[3GPP TS 23.228] 3GPP TS 23.228 "IP Multimedia Subsystem (IMS); Stage 2 (Release 6)". URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.228/

[3GPP TS 24.229] 3GPP TS 24.229 "IP Multimedia Call Control Protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP)"; Stage 2 (Release 6)". URL: http://www.3gpp.org/ftp/Specs/archive/24_series/24.229/

[3GPP2 X.S0013.4] 3GPP2 X.S0013-004-0 "IP Multimedia Call Control Protocol Based on SIP and SDP Stage 3". URL: http://www.3gpp2.org/Public_html/specs/

[3GPP2 X.S0013-002] 3GPP2 X.S0013-002-0 "All-IP Core Network Multimedia Domain; IP Multimedia Subsystem - Stage 2", version 1.0, December 2003. URL: http://www.3gpp2.org

[3GPP TS 33.141] 3GPP TS 33.141 "Presence service; Security"; (Release 6)". URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.141/

2.2 Informative References

[XDMAD] "XML Document Management Architecture", Version 1.0, Open Mobile Alliance™. OMA-AD-XDM-V1_0, URL:http://www.openmobilealliance.org/.

[Shared_XDM] "Shared XDM Specification", Version 1.0, Open Mobile Alliance™, OMA-TS-XDM_Shared-V1_0, URL:http://www.openmobilealliance.org/.

[PoC_XDM] "PoC XDM Specification", Version 1.0, Open Mobile Alliance™, OMA-TS-POC_XDM-V1_0, URL:http://www.openmobilealliance.org/.

[RLS_XDM] "Resource List Service (RLS) XDM Specification", Version 1.0, Open Mobile Alliance™, OMA-TS-PRESENCE_SIMPLE_RLS_XDM-V1_0, URL:http://www.openmobilealliance.org/.

[Presence_XDM] "Presence XDM Specification", Version 1.0, Open Mobile Alliance™, OMA-TS-PRESENCE_SIMPLE_XDM-V1_0, URL:http://www.openmobilealliance.org/.

[COMMONPOL] "A Document Format for Expressing Privacy Preferences", H. Schulzrinne, J. Morris, H. Tschofenig, J. Cuellar, J. Polk, J. Rosenberg, October. 2004, URL:http://www.ietf.org/internet-drafts/draft-ietf-geopriv-common-policy-03.txt

Note: work in progress.

[RFC3040] "Internet Web Replication and Caching Taxonomy", I. Cooper, I. Melve, G. Tomlinson, January 2001, URL:http://www.ietf.org/rfc/rfc3040.txt.

3. Terminology and Conventions

3.1 Conventions

The key words “SHALL”, “SHALL NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Application Unique ID (AUID)	A unique identifier that differentiates XCAP resources accessed by one application from XCAP resources accessed by another. [Source: XCAP]
Document URL	The HTTP URL containing the XCAP root and document selector, resulting in the selection of a specific document. (Source: [XCAP])
Global document	A document placed under the XCAP global tree that applies to all users of that application usage.
Global tree	A URL that represents the parent for all global documents for a particular application usage within a particular XCAP root. (Source: [XCAP])
Node URL	The HTTP URL containing the XCAP root, document selector, path separator and node selector, resulting in the selection of a specific XML node. (Source: [XCAP])
Primary Principal	The principal who has full access rights (e.g., read, write, delete) for a given document, including the right to delegate some of these rights to other principals. (Source: [XDM RD])
Reverse Proxy	A reverse proxy is a web server system that is capable of serving web pages sourced from other web servers (AS), making these pages look like they originated at the reverse proxy. (Source: [3GPP TS 33.222])
XCAPApplication Usage	Detailed information on the interaction of an application with an XCAP server. (Source: [XCAP])
XCAP Client	An HTTP client that understands how to follow the naming and validation constraints defined in this specification. (Source: [XCAP])
XCAP Root	A context that includes all of the documents across all application usages and users that are managed by a server. [Source: XCAP]
XCAP Root URL	An HTTP URI that represents the XCAP root. Although a valid URI, the XCAP Root URI does not correspond to an actual resource. [Source: XCAP]
XCAP Server	An HTTP server that understands how to follow the naming and validation constraints defined in this specification. (Source: [XCAP])
XCAP User Identifier (XUI)	The XUI is a string, valid as a path element in an HTTP URI, that is associated with each user served by the XCAP server. [Source: XCAP]

3.3 Abbreviations

OMA	Open Mobile Alliance
OMNA	OMA Naming Authority
TLS	Transport Layer Security
XCAP	XML Configuration Access Protocol
XDM	XML Document Management
XML	Extensible Markup Language
AUID	Application Unique ID

XUI	XCAP User Identifier
URI	Uniform Resource Identifier
GAA	Generic Authentication Architecture
HTTP	Hyper Text Transfer Protocol
UE	User Equipment
AS	Application Server
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
MMD	MultiMedia Domain

4. Introduction

Various OMA enablers such as, Presence, Push to Talk Over Cellular (PoC), Instant Messaging (IM), etc. need support for access to and manipulation of certain information that are needed by these enablers. Such information is expressed as XML documents and stored in various document repositories in the network where such documents can be located, accessed and manipulated (created, changed, deleted) by authorised principals.

This specification defines the common protocol for access and manipulation of such XML documents by authorized principals. This specification reuses the IETF XML Configuration Access Protocol (XCAP).

XCAP defines:

- A convention for describing elements and attributes of an XML document as a HTTP resource, i.e., accessible via an HTTP URI
- A technique for using HTTP GET, PUT and DELETE methods for various document manipulation operations (e.g., retrieving/adding/deleting elements/attributes, etc.)
- The concept and structure of an XCAP Application Usage by which service or enabler specific documents can be described
- A default authorization policy for accessing and manipulating documents

This specification also defines a technique by which changes to such XML documents can be conveyed to an XCAP Client. This reuses an IETF-defined SIP event package by which an XDM Client subscribes to changes to all documents that it owns.

Common, reusable as well as enabler-specific document formats and associated XCAP application usages are described in separate specifications (e.g., [Shared_XDM] [PoC_XDM] [Presence_XDM] and [RLS_XDM]) that make use of the XCAP protocol specified here for their document management.

5. Description of Functional Elements

5.1 XDM Client

The XDM Client SHALL support the XDM Client procedures described in section 6.1, and the XCAP application usages described in Section 6.7.

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDM Client MAY be implemented in a UE or an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002] respectively.

5.2 Aggregation Proxy

The Aggregation Proxy is the contact point for the XDM Client implemented in an UE to access XML documents stored in any XDMS.

The Aggregation Proxy SHALL act as an HTTP Proxy defined in [RFC2616] with the following clarifications. The Aggregation Proxy:

1. SHALL be configured as an HTTP reverse proxy (see [RFC3040]);
2. SHALL support authenticating the XDM Client; in case the GAA is used according to [3GPP TS 33.222], the mutual authentication SHALL be supported; or SHALL assert the XDM Client identity by inserting the X-XCAP-Asserted-Identity extension header to the HTTP requests after a successful HTTP Digest Authentication as defined in Section 6.3.2, in case the GAA is not used.
3. SHALL forward the XCAP requests to the corresponding XDM Server, and forward the response back to the XDM Client;
4. SHALL protect the XCAP traffic by enabling TLS transport security mechanism. The TLS resumption procedure SHALL be used as specified in [RFC2818].

When realized with 3GPP IMS or 3GPP2 MMD networks, the Aggregation Proxy SHALL act as an Authentication Proxy defined in [3GPP TS 33.222] with the following clarifications. The Aggregation Proxy: SHALL check whether an XDM Client identity has been inserted in X-3GPP-Intended-Identity header of HTTP request.

- If the X-3GPP-Intended-Identity is included , the Aggregation Proxy SHALL check the value in the header is allowed to be used by the authenticated identity.
- If the X-3GPP-Intended-Identity is not included, the Aggregation Proxy SHALL insert the authenticated identity in the X-3GPP-Asserted-Identity header of the HTTP request.

6. Description of procedures

6.1 Procedures at the XDM Client

An XDM Client is an entity that accesses a XCAP resource in an XML Document Management Server (XDMS). Such XCAP resources correspond to elements and attributes of an XML document. An XCAP resource is identified via an HTTP URI following the conventions for constructing URIs in [XCAP].

6.1.1 Document Management

6.1.1.1 XDM URI Construction

An HTTP URI represents each element and attribute of an XML document in a XDM respository. The rules for constructing such URIs SHALL follow the rules described in [XCAP] Section 6 with the clarifications given in this sub-clause.

Therefore, for example, a generic XCAP URI would be of the form [XCAP Root URL]/[AUID]/users/[XUI]/…..(See Appendix B for examples.)

The DNS lookup of the hostname of [XCAP Root URL] SHALL resolve to the address of the Aggregation Proxy. The path segment corresponding to the XUI SHALL be a Public SIP URI of form sip: user@domain, identifying the document owner.

6.1.1.2 XDM Operations

An XDM Client manipulates an XML document by invoking certain HTTP operations (defined in sub-sections below) on the XDM resource identified in the Request-URI of the HTTP header.

The client SHALL construct the Request-URI based on its knowledge of the application usage governing that XML document.

An XDM client MAY implement the conditional operations of [XCAP] section 7.10.

An XDM client MAY support HTTP compression using content encoding. If the XDM client utilizes HTTP compression, it SHALL set the “Accept-Encoding” header as defined in [RFC2616].

6.1.1.2.1 Create or Replace a Document

Creating or replacing an XML document SHALL follow the procedures described in [XCAP] Section 7.1.

6.1.1.2.2 Delete a Document

Deleting an XML document SHALL follow the procedures described in [XCAP] Section 7.2.

6.1.1.2.3 Retrieve a Document

Retrieving an XML document SHALL follow the procedures described in [XCAP] Section 7.3.

6.1.1.2.4 Create or Replace an Element

Creating or replacing an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.4.

6.1.1.2.5 Delete an Element

Deleting an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.5.

6.1.1.2.6 Retrieve an Element

Retrieving an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.6.

6.1.1.2.7 Create or Replace an Attribute

Creating or replacing an attribute of an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.7.

6.1.1.2.8 Delete an Attribute

Deleting an attribute of an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.8.

6.1.1.2.9 Retrieve an Attribute

Retrieving an attribute of an element in an XML document SHALL follow the procedures described in [XCAP] Section 7.9.

6.1.2 Subscribing to changes in the XML documents

6.1.2.1 Initial subscription

If the XDM Client subscribes to changes in XML documents, then it SHALL be done by sending a SUBSCRIBE request according to [RFC3265] and [SIP_UA_Prof] with the clarifications given in this sub-clause.

The XDM Client

1. SHALL set the Request-URI to the public SIP URI identifying the document owner, or to the SIP URI identifying the service instance (e.g. PoC Group URI, or Presence List URI);
2. SHALL include value “application” in the “profile-name” Event header parameter;
3. SHALL include the AUID to be watched in the “app-id” Event header parameter;
4. In case the public SIP URI identifying the owner of the document is set as Request-URI, then the “document” Event header parameter MAY be set to specify the document to be watched.

   In case the service instance SIP URI is set as Request-URI, then the “document” Event header parameter SHALL be set to specify the relevant document stored in the “global” tree for this service instance SIP URI.

Note: For example, if the Request-URI SIP URI identifying the service instance is “sip:my_friends@example.com”, the document parameter has to be set to “document= global/group/list-service[@uri=sip:my_friends@example.com]” for a PoC group.

5. SHALL include an Accept header to indicate acceptable content-type for notifications. The Accept header
   a. MAY include the value “application/xcap-diff+xml” to indicate support for partial XML updates described in [XCAP_Diff];
   b. MAY include the value “message/external-body” to indicate support for content indirection described in [INDIRMECH];
6. SHALL send the SUBSCRIBE request towards the SIP/IP Core according to the procedures of the SIP/IP core.

Note: The XDM Client is not able to subscribe for changes in multiple documents stored under different AUIDs in a single subscription. This functionality has been postponed for a future release.

The responses to the SUBSCRIBE request SHALL be handled in accordance with [RFC3265], [SIP_UA_Prof], and the procedures of the SIP/IP core.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, an UE acting as the XDM Client SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in chapter 5.1 in [3GPP TS 24.229] / [3GPP2 X.S0013.4] and an AS acting as the XDM Client SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in chapter 5.7.3 [3GPP TS 24.229] / [3GPP2 X.S0013.4] with the clarifications given in the respective sub clauses.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, and the XDM Client resides in an Application Server (e.g. PoC Server) the mechanisms of the "Application Server acting as originating User Agent" SHALL be applied as defined in [3GPP TS 24.229] / [3GPP2 X.P0013.4] section 5.7.3 and setting its public SIP URI in the "P-Asserted-Identity" header.

6.1.2.2 NOTIFY processing

Upon receiving an incoming NOTIFY request that is part of the same dialog as the previously sent SUBSCRIBE request the XDM Client

1. SHALL handle the request according to [RFC3265], [SIP_UA_Prof], and the procedures of the SIP/IP core;
2. SHOULD update the stored XML document based on the information in the NOTIFY request.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, the XDM Client SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in [3GPP TS 24.229] / [3GPP2 X.S0013.4] with the clarifications given in this sub-clause.

6.2 Procedures at the XDM Server

A XDM Server (XDMS) is a HTTP origin server that manipulates XML resources according to the conventions described in [XCAP].

When the SIP/IP Core network corresponds with 3GPP IMS or 3GPP2 MMD networks, the XDM Server SHALL be implemented in an AS as defined in [3GPP TS 23.228] and [3GPP2 X.S0013-002] respectively.

6.2.1 Document Management

An XDM Server receiving an HTTP request targeted at an XCAP resource identified by the HTTP Request-URI follows the following procedures based on the method requested.

An XDM server SHALL conform to [XCAP] section 8.5 for the management of Etags.

An XDM server SHALL implement the conditional operations of [XCAP] section 7.10.

If the XDM Server implements parallel processing of requests, it SHALL ensure the integrity of the resulting document.

6.2.1.1 POST handling

HTTP POST requests targeted at an XDM resource SHALL be rejected with an HTTP 405 "Method not allowed" response as described in [XCAP] Section 8.1.

6.2.1.2 PUT handling

HTTP PUT requests targeted at an XDM resource SHALL be processed as described in [XCAP] Section 8.2.

6.2.1.3 GET handling

HTTP GET requests targeted at an XDM resource SHALL be processed as described in [XCAP] Section 8.3.

6.2.1.4 DELETE handling

HTTP DELETE requests targeted at an XDM resource SHALL be processed as described in [XCAP] Section 8.4.

6.2.2 Subscriptions to changes in the XML documents

6.2.2.1 Initial subscription

Upon receiving a SUBSCRIBE request for the "sip-profile" event defined in [SIP_UA_Prof] the XDM Server performs the following steps:

1. SHALL return the SIP “501 Not Implemented” error response, if the “sip-profile” event is not supported. Otherwise perform the following steps.
2. SHALL use the Request-URI
    a. as an XUI identifying the owner of the document in case the “document” event header parameter is defining a document in the “users” tree or is not set;
    b. as a service instance SIP URI (e.g. PoC group) in case the “document” event header parameter is defining a document in the “global” tree;
3. SHALL perform the necessary authorization checks on the originator. When the SIP/IP Core corresponds to 3GPP/3GPP2 IMS the XDM Server SHALL use the "P-Asserted-Identity" as defined in [3GPP TS 24.229] / [3GPP2 X.S0013.4] to ensure that this particular XDM Client is authorized to track the document changes. If the authorization check fails, the XDM Server SHALL return the SIP "403 Forbidden" error response.
    a. By default, the initial creator of the document in the "users" tree is the primary principal of that document and SHALL be authorized to subscribe to the “sip-profile” event package as described in Section 6.4.2.
    b. Other principals (e.g. XDMCs and Application Servers) identified by their "P-Asserted-Identity" headers MAY be authorised to subscribe based on “local configuration policy”.

Note: An example of such a “local configuration policy” is the authorization of subscriptions to a service instance SIP URI (e.g. PoC group URI), in which the members of that particular group, represented by that service instance SIP URI, are authorised to subscribe for changes.

4. SHALL create a subscription to changes of XML data identified by Event header parameters as described in [SIP_UA_Prof];
5. SHALL send a SIP “200 OK” in accordance with [RFC3265], [SIP_UA_Prof], and the procedures of the SIP/IP core.
6. SHALL generate and send an initial NOTIFY request as specified in sub-clause 6.2.2.2 “Generating a NOTIFY request”.

When a change in the subscribed document occurs, the XDM Server SHOULD generate and send a NOTIFY request as specified in sub-clause 6.2.2.2 “Generating a NOTIFY request”.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, the XDM Server SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in [3GPP TS 24.229] / [3GPP2 X.S0013.4] with the clarifications given in this sub-clause.

6.2.2.2 Generating a NOTIFY request

The XDM Server SHALL generate a NOTIFY request as described in the [RFC3265] and [SIP_UA_Prof] with the clarifications given in this sub-clause.

The XDM Server

1. SHALL check content-types accepted by the XDM Client as indicated in the SUBSCRIBE request (see sub-clause 6.1.2.1);
    a. if both indirect and directly supplied content are acceptable, the XDM Server MAY include either alternative;
    b. if acceptable for the XDM Client, the XDM Server SHALL include an “application/xcap-diff+xml” body as defined in [XCAP_Diff];
    c. if acceptable for the XDM Client, the XDM Server SHALL include a “message/external-body” body as defined in [SIP_UA_Prof] and [INDIRMECH];

2. SHALL send the NOTIFY request towards the SIP/IP Core according to the procedures of the SIP/IP core.

The responses to the NOTIFY request SHALL be handled in accordance with [RFC3265], [SIP_UA_Prof], and the procedures of the SIP/IP core.

When the SIP/IP Core corresponds with 3GPP/3GPP2 IMS, the XDM Server SHALL use 3GPP/3GPP2 IMS requirements, mechanisms and procedures as defined in [3GPP TS 24.229] / [3GPP2 X.S0013.4] with the clarifications given in this sub-clause.

6.3 Procedures at the Aggregation Proxy

The Aggregation Proxy performs security procedures, as well as the request forwarding procedure for HTTP traffic. The first function is covered in section 6.3.1 and 6.3.2, and the request forwarding procedure is covered in section 6.3.3.

6.3.1 Authentication

The authentication function SHALL be performed over XDM-3 reference point (see [XDMAD]). The initial HTTP request from XDM Client SHALL be interrogated by the Aggregation Proxy using the HTTP Digest mechanism as specified in [RFC2617].

The Aggregation Proxy SHALL fulfill the functions described in sub-clause 6.4.1.

6.3.2 XDM Client identity assertion

When the 3GPP GAA is not present the Aggregation Proxy SHALL perform the following:

1. Insert the X-XCAP-Asserted-Identity extension header to the HTTP requests after a successful HTTP Digest Authentication ;
2. Populate the X-XCAP-Asserted-Identity with the public SIP URI in quotation marks (“”) provided by the “username” field in the HTTP Digest Authorization header.
3. Ensure that only one instance of the X-XCAP-Asserted-Identity header exists in the HTTP Requests before forwarding it. In cases where there are multiple instances, the Aggregation Proxy SHALL remove all previous instances of this header and insert its own provided that the XDM Client authentication with the Aggregation Proxy was successful

When realized in 3GPP IMS and the GAA is present, the procedures described [3GPP TS 24.109] SHALL be followed with the following clarifications:

1. The Aggregation Proxy SHALL check whether an XDM Client identity has been inserted in X-3GPP-Intended-Identity header of HTTP request. If so, the Aggregation Proxy SHALL check the value in the header is equal to the authenticated identity.
2. If the X-3GPP-Intended-Identity is not included, the Aggregation Proxy SHALL insert authenticated identity in the X-3GPP-Asserted-Identity header of the HTTP request.

6.3.3 XCAP request forwarding

6.3.3.1 General

Upon receiving an XCAP request targeted to the Aggregation Proxy, the Aggregation Proxy:

1. SHALL act as an HTTP reverse proxy;
2. SHALL forward the XCAP request to the corresponding XDM Server based on the HTTP Request URI.

The response to the XCAP request SHALL be sent back to the originator.

6.3.3.2 XCAP Server Capabilities retrieval

Upon receiving an XCAP GET request for the "xcap-caps" AUID (described in section 6.7.1), the Aggregation Proxy:

1. SHALL act as an HTTP reverse proxy;
2. SHALL obtain XCAP Server Capabilities from all XDM Servers that serve the request originator. To perform this operation the Aggregation Proxy SHALL:
    a. forward the XCAP request to all XDM Servers that serve the request originator;
    b. if the target XDM Server responded with HTTP "200 OK" response, collect the <auid>, <extension> and <namespace> elements.
3. SHALL return the HTTP "200 OK" response with the "application/xcap-caps+xml" body including all received <auid>, <extension> and <namespace> elements.

Upon receiving of other HTTP request for an "xcap-caps" document, the Aggregation Proxy shall respond with an HTTP "405 Method Not Allowed" response.

6.3.3.3 XCAP Directory retrieval

Upon receiving an XCAP GET request for the "org.openmobilealliance.xcap-directory" AUID (described in section 6.7.2), the Aggregation Proxy:

1. SHALL act as an HTTP reverse proxy;
2. SHALL obtain the requested XCAP Directory from all XDM Servers that serve the request originator. To perform this operation the Aggregation Proxy SHALL:
    a. forward the XCAP request to all XDM Servers that serve the request originator;
    b. if the target XDM Server responded with HTTP "200 OK" response, collect the <folder> elements.
3. SHALL return the HTTP "200 OK" response with the "application/oma-directory+xml" body including all received <folder> elements.

Upon receiving of other HTTP request for an "org.openmobilealliance.xcap-directory" document, the Aggregation Proxy shall respond with an HTTP "405 Method Not Allowed" response.

6.3.4 Compression

The Aggregation Proxy MAY support compression using content encoding.

If the Aggregation Proxy supports compression it SHALL follow the procedures defined in [RFC2616].

6.4 Security Procedures

6.4.1 Authentication

The XDM-3 reference point (see [XDMAD]) SHALL provide mutual authentication.

For a 3GPP/3GPP2 realisation, the XDM-3 corresponds to the Ut interface. In this case the authentication between theXDM Client and the Aggregation Proxy SHALL be performed according to [3GPP TS 33.141] / [3GPP2 X.S0027-003-0].

If the Generic Authentication Architecture (GAA) as defined in [3GPP TS 33.222] is not used, the XDM Client and the Aggregation Proxy (see [XDMAD]) SHALL support the HTTP Digest mechanism for client authentication.

The HTTP Digest authentication SHALL conform to [RFC2617] with the following clarifications:

- The HTTP server ("401 Unauthorized") SHALL be used;

- the "rspauth" parameter MAY be used to provide mutual authentication;
- the "username" parameter SHALL contain the SIP URI identifying the user (the public user identity);
- the "qop" header shall be set to "auth-int".

The XDM Client and the Aggregation Proxy SHALL support HTTP over Transport Layer Security (TLS) as specified in [RFC2818] for server authentication over the XDM-3 interface.

For a 3GPP/3GPP2 realization, the reference points between the Aggregation Proxy and any XDMS or an Application Server and any XDMS uses the security mechanisms defined in 3GPP/3GPP2 that are out of scope of this specification.

6.4.2 Integrity and Confidentiality protection

The XDM Client and the Aggregation Proxy SHALL support the TLS as specified in [RFC2246] with the following clarifications:

- The following cipher suites SHALL be supported:
  - TLS_RSA_WITH_3DES_EDE_CBC_SHA
  - TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- other cipher suites defined in [RFC2246] MAY be supported.

When the SIP/IP Core corresponds with 3GPP IMS, the XDM Client and the Aggregation Proxy SHALL support the TLS version and profile as specified in clause 5.3 of [3GPP TS 33.222].

6.4.3 Authorization

The XDMS SHALL check that the identity of the requesting XDMC has been granted access rights to perform the requested operations. Application usages MAY define their own policies for accessing different XCAP resources (e.g. global documents).

The XDMS SHALL use the information in the X-XCAP-Asserted-Identity header provided by the Aggregation Proxy to determine the identity of the XDM Client.

When realized in 3GPP IMS and the GAA is present, the identity of the requesting XDMC is obtained from the X-3GPP-Asserted-Identity or the X-3GPP-Intended-Identity..

By default, the initial creator of a document is its primary principal. The primary principal SHALL have permission to perform all operations defined in Sections 6.1.1 and 6.1.2. In this release, it will not be possible to change the primary principal. Additionally, it will not be possible to assign permissions to access or manipulate a document to anyone except for the primary principal or trusted applications.

Any application usage defining the use of any global documents SHALL specify the authorization policy associated with the use of such documents.

6.5 Error cases

If the Aggregation Proxy or XDM server receives an HTTP request targeted at an XCAP resource whose application usage is not recognized or understood, the Aggregation Proxy or XDM Server SHALL reject the request with an HTTP 404 (Not Found) response.

Additional validation constraints might be applied which may result in a HTTP 409 Response. An HTTP 409 error response SHALL include a document in the HTTP body that conforms to that defined in [XCAP] Section 9.

For additional details of the handling of those, see [XCAP] Section 8.2.5.

Other specifications MAY define the value of the "phrase" attribute, which contains text for rendering to a human user, that is optionally present in an error element identifying an error condition.

An HTTP error response SHALL be sent to the XDMC after several failed responses to a challenge. The exact count of challenges is decided by local implementation policy.

6.6 Common Extensions

6.6.1 Lists defined in Shared XDMS

Various entities may wish to refer to a shared URI list stored in the [Shared_XDMS]. The "external" element provides the means to make such references, in a similar manner across different entities.

The "external" element SHALL contain either an XCAP document URI pointing to a "resource-lists" document or an XCAP node URI pointing to a "list" element within a "resource-lists" document.

NOTE: There is an <external-list> element defined in section 6.6.2. It points to an external list against which the rules are specified according to [COMMONPOL]. See section 6.6.2.3 for the detail.

Entities that utilize the "external" element will resolve it to a set of entities according to the following rules:

- If the external-list element contains a document URI, then it SHALL resolve to all the entities listed in all the lists within that resource document.
- If the list contains a node URI, then it SHALL resolve only to entities within the specific list that is pointed to.

Any node resolving an <external> SHALL NOT fetch the contents of the referenced <external> if it has already done so.

6.6.2 Authorization Rules

Every authorization policy based on [COMMONPOL] SHALL support the extensions defined in this sub-clause.

6.6.2.1 Structure

Every rule in an authorization policy document SHALL support the following extensions to [COMMONPOL]:

- The "identity" condition element (as defined in section 6.6.2.2, which is different from [COMMONPOL])
- the "external-list" condition element (as defined in section 6.6.2.2);
- the "other-identity" condition element.

If present in any rule, the "external-list" element allows for matching those identities that are part of a URI List (as defined in section 6.6.2.2).

If present in any rule, the "other-identity" element, which is empty, matches all identities that are not referenced in any rule. It allows for specifying a default policy.

It is RECOMMENDED that each rule be based on a single condition.

6.6.2.2 XML Schema

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

  <!-- OMA specific "conditions" child elements -->    <xs:element name="other-identity"
substitutionGroup="cr:condition"/>
<xs:element name="identity" substitutionGroup="cr:condition">
    <xs:complexType>
```

```
        <xs:sequence>
         <xs:element name="entry" type="entType" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="anonymous" type="xs:string"
               minOccurs="0" maxOccurs="1">
            <xs:sequence minOccurs="0">
              <xs:element name="domain" type="xs:string"/>
            </xs:sequence>
                  </xs:element>
          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

     <xs:element name="external-list" substitutionGroup="cr:condition">
       <xs:complexType>
         <xs:sequence>
           <xs:element name="entry" type="anchorType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
       </xs:complexType>
     </xs:element>

   <xs:complexType name="entType">
     <xs:attribute name="id" type="xs:anyURI"/>
     <xs:anyAttribute namespace="##any"/>
   </xs:complexType>

   <xs:complexType name="anchorType">
     <xs:attribute name="anc" type="xs:anyURI"/>
     <xs:anyAttribute namespace="##any"/>
   </xs:complexType>

 </xs:schema>
```

Editor note: It must be clarified how to register the OMA specific namespace.

6.6.2.3 Combining Permissions

When evaluating any authorization policy document based on [COMMONPOL] together with the extensions described in section 6.6.1.2 against a URI value, the following rules for combining permissions from the different <rule>s that are applicable SHALL be as follows:

a Those rules matching the URI value against the <identity> element SHALL take precedence over those rules based on matching it against an <external-list> or an <other-identity>. That is, if there are applicable rules based on <identity> matches, only these shall be used for the evaluation of the combined permission.

b Those rules matching the URI value against an <other-identity> element SHALL be used for the evaluation of the combined permission only if there are no applicable rules based on matching against an <identity> or <external-list> element

The evaluation of the combined permission SHALL be based on [COMMONPOL] Section 10.

6.7 Common Application Usage

6.7.1 XCAP Server Capabilities

Every XDM server SHALL support the Application Usage "xcap-caps", which defines the capabilities of the server, as defined in [XCAP] Section 11.

The single document in the "global" tree corresponding to the "xcaps-caps" Application Usage SHALL be available to all principals as a part of the global URI tree.

6.7.2 XML Documents Directory

The XML Documents Directory application usage allows an XDM Client (corresponding to a given XUI) to fetch:

1. the list of all XCAP managed documents corresponding to that XUI across all XDMSes, or
2. the list of all documents for a given AUID corresponding to that XUI stored in an XDMS.

An XDMS SHALL support an application usage named "org.openmobilealliance.xcap-directory" and SHALL maintain one document in the "users" tree per XUI named "directory.xml".

Thus, a XCAP GET request targeted at the URI http://[XCAP Root URL]/ org.openmobilealliance.xcap-directory /users/sip:joe@example.com/directory.xml should return a list of all XML documents associated with all AUIDs for the user identified by sip:joe@example.com.

The structure of the "directory.xml" document is as follows: it is a well-formed and valid XML document encoded in UTF-8 that begins with the root element <xcap-directory>. It consists of a number of <folder> elements.

Each <folder> element SHALL have an attribute "auid", whose value corresponds to an AUID that the XCAP server supports and for which there are documents in the "users" tree corresponding to.a given XUI.

Every <folder> element consists of a number of <entry> elements. Each <entry> element containing a number of attributes, which are:

1. uri: this attribute SHALL be the Document URL for a document corresponding to the "auid" attribute value in the parent <folder> element and for the given XUI.
2. etag: this attribute SHALL contain the server computed etag value of the current instance of the XML document identified by the "uri" attribute value. (This allows the XCAP client to determine whether the locally cached copy of a document is up-to-date.
3. last-modified: this attribute is OPTIONAL. When present, it SHALL contain the date and time the document identified as above was last modified. (This allows the XCAP client to determine if whether a document has changed recently or not.)
4. size: this attribute is OPTIONAL. When present, it SHALL contain the size, in octets, of the document as identified above. (This can help an XCAP client determine if it wants to upload the entire document or a fragment, as appropriate based on any resource limitation such as bandwidth.)

Thus, for example, a XCAP GET request targeted at the URI http://[XCAP Root URL]/org.openmobilealliance.xcap-directory/users/sip:joe@example.com/directory.xml/~~/xcap-directory/folder[@auid="org.openmobilealliance.poc-groups"] should return all <entry> elements corresponding to sip:joe@example.com 's PoC groups.

Note: The character escaping SHALL be applied in HTTP URI representation according to [XCAP] Section 6.3.

6.7.2.1 Application Unique ID

This specification defines the "org.openmobilealliance.xcap-directory" AUID.

6.7.2.2 MIME Type

The MIME type for this document is “application/oma-directory+xml”

6.7.2.3 Default Namespace

The default namespace SHALL be:

“urn:oma.params:xml:ns:xcap-directory”

6.7.2.4 XML Schema

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

    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
               schemaLocation="http://www.w3.org/2001/xml.xsd"/>

<xs:element name="xcap-directory">
   <xs:complexType>
     <xs:sequence minOccurs="0" maxOccurs="unbounded">
       <xs:element name="folder">
          <xs:complexType>
             <xs:sequence minOccurs="0" maxOccurs="unbounded">
               <xs:element name="entry">
                 <xs:complexType>
                   <xs:attribute name="uri" type="xs:anyURI" use="required"/>
                   <xs:attribute name="etag" type="xs:string" use="required"/>
                   <xs:attribute name="last-modified" type="xs:dateTime"
use="optional"/>
                   <xs:attribute name="size" type="xs:nonNegativeInteger"
use="optional"/>
                   <xs:anyAttribute namespace="##other"/>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
              <xs:attribute name="auid" type="xs:string" use="required"/>
             </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

6.7.2.5 Additional Constraints

None.

6.7.2.6 Data Semantics

See section 6.7.2.

6.7.2.7 Naming Conventions

There is only one XCAP directory document per XUI in each XDMS. Therefore, the XDMS SHOULD assign the directory document the name “directory.xml”.

To retrieve such a directory document, the XCAP Client SHALL always use this same name.

6.7.2.8 Data Interdependencies

For every document created/deleted/modified in the "users" tree for a particular XUI and application usage, the XDMS SHALL add/delete/update the appropriate <entry> child element in the appropriate <folder> element of the "directory.xml" document corresponding to that XUI.

NOTE: This does not imply that the server must actually store this "directory" document. All that is required is that the XDMS be able to serve an up-to-date version of such a document when requested.

The XDMS SHOULD NOT generate an etag value for the "directory" document.

NOTE: This implies that conditional operations are not supported against the "directory" document. The XCAP Client should always refresh any cached copy.

6.7.2.9 Authorization Policies

The "directory.xml" document is created and modified only by the XDMS. Thus, authorized principals are only allowed to retrieve this document.

The authorization policies for retrieving a "directory.xml" document SHALL conform to those described in [XDM_Spec] section 6.4.3.

6.8 Global Documents

[XCAP] specifies a global tree which is used to place documents applicable to a particular application usage but which are not specific to any particular user. An example of this is the "xcap-caps" document (see section 6.7.1) describing the application usages supported by an XDMS.

If used, each application usage describes how each global document is constructed and any associated authorization policy.

Appendix A. Static Conformance Requirements (normative)

The SCRs defined in the following tables include SCR for:

- Aggregation Proxy
- XDM Server
- XDM Client

Each SCR table identifies a list of supported features as:

Item: Identifier for a feature.

Function: Short description of the feature.

Reference: Section(s) of this specification with more details on the feature.

Status: Whether support for the feature is mandatory or optional. MUST use "M" for mandatory support and "O" for optional support in this column.

Requirement: This column identifies other features required by this feature. If no other features are required, this column is left empty.

This section describes the dependency grammar notation to be used in the Requirement column of the SCR and CCR tables using ABNF [RFC2234].

TerminalExpression = ScrReference / NOT TerminalExpression / TerminalExpression LogicalOperator TerminalExpression / "(" TerminalExpression ")"

ScrReference = ScrItem / ScrGroup

ScrItem = SpecScrName "–" GroupType "–" DeviceType "–" NumericId / SpecScrName "–" DeviceType "–" NumericId

ScrGroup = SpecScrName ":" FeatureType / SpecScrName "– " GroupType "–" DeviceType "–" FeatureType

SpecScrName = 1*Character;

GroupType = 1*Character;

DeviceType = "C" / "S"; C – client, S – server

NumericId = Number Number Number

LogicalOperator = "AND" / "OR"; AND has higher precedence than OR and OR is inclusive

FeatureType = "MCF" / "OCF" / "MSF" / "OSF"; See Section A.1.6

Character = %x41-5A ; A-Z

Number = %x30-39 ; 0-9

A.1 XDM Client

A.1.1 XDM Client implemented in a UE

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-001	Support rules for constructing HTTP URIs	6.1.1.1	M	

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-002	Support for XDM Operations	6.1.1.2	M	
XDM-XDMC-C-003	Initial Subscription using the SUBSCRIBE message	6.1.2.1	O	XDM-XDMC-C-004
XDM-XDMC-C-004	Processing Received NOTIFY Request	6.1.2.2	O	XDM-XDMC-C-003
XDM-XDMC-C-005	Support HTTP Digest authentication	6.4.1	M	
XDM-XDMC-C-006	Support HTTP over TLS using the two supported cipher suites	6.4.1	M	
XDM-XDMC-C-007	Support other cipher suites defined in RFC2246	6.4.1	O	
XDM-XDMC-C-008	Support HTTP Compression	6.1.1.2	O	

A.1.2 XCAP Client implemented in an AS

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-001	Support rules for constructing HTTP URIs	6.1.1.1	M	
XDM-XDMC-C-002	Support for XDM Operations	6.1.1.2	M	
XDM-XDMC-C-003	Initial Subscription using the SUBSCRIBE message	6.1.2.1	O	XDM-XDMC-C-004

Item	Function	Reference	Status	Requirement
XDM-XDMC-C-004	Processing Received NOTIFY Request	6.1.2.2	O	XDM-XDMC-C-003
XDM-XDMC-C-008	Support HTTP Compression	6.1.1.2	O	

A.2 XDM Server

Item	Function	Reference	Status	Requirement
XDM-XDMS-S-001	Support for XCAP	6.2.1	M	
XDM-XDMS-S-002	Support Initial Subscription when SUBSCRIBE message received	6.2.2.1	O	
XDM-XDMS-S-003	"Not Implemented" Error Handling or SUBSCRIBE request Handling	6.2.2.1	M	
XDM-XDMS-S-004	Generating a NOTIFY request	6.2.2.2	O	
XDM-XDMS-S-005	Support XDMC identity access authorization	6.4.3	M	
XDM-XDMS-S-006	"Usage not understood" Error Handling	6.5	M	
XDM-XDMS-S-007	Support Application Usage "xcap-caps"	6.6.1	M	
XDM-XDMS-S-008	Support Application Usage "xcap-directory"	6.6.2	M	

A.3 Aggregation Proxy

Item	Function	Reference	Status	Requirement
XDM-AP-S-001	Support HTTP Digest authentication	5.3, 6.3.1, 6.4.1	M	

Item	Function	Reference	Status	Requirement
XDM-AP-S-002	Support HTTP over TLS using the two supported cipher suites	5.3, 6.4.2	M	
XDM-AP-S-003	Support other cipher suites defined in RFC2246	6.4.1	O	
XDM-AP-S-004	Support XDM Client Identity Assertion	5.3, 6.3.2	M	
XDM-AP-S-005	Support XCAP request forwarding	6.3.3	M	
XDM-AP-S-006	Support Compression	6.3.4	O	
XDM-AP-S-007	Support for GAA	6.3, 6.4	O	

Appendix B. Examples (informative)

B.1 Sample XCAP Operation

Figure B.1 describes how an XCAP operation is performed in 3GPP/3GPP2 IMS. The "resource-list" application usage (see [Shared_XDMS]) i.e. the manipulation of a URI List is used in this specific example, but the same types of messages apply for other application usages (although the HTTP body content would, of course, be different). It is also assumed that the address of Aggregation Proxy is "xcap.example.com" and the XCAP Root URL is xcap.example.com/services".

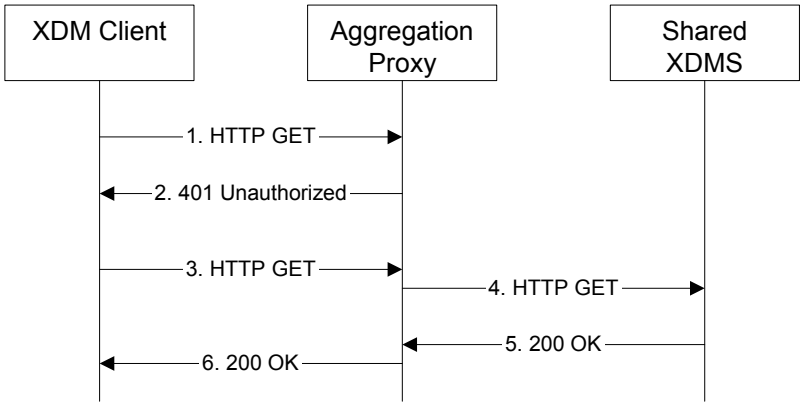

Figure B.1- Sample XCAP operation

The details of the flows are as follows:

1) The user "sip:joebloggs@example.com" wants to obtain an XML document. For this purpose the XDMC sends a HTTP GET request to the Aggregation Proxy.

```
GET http://xcap.example.com/services/resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA1.0
Date: Thu, 08 Jan 2004 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Length: 0
```

2) Upon receiving an unauthorized HTTP GET the Aggregation Proxy chooses to authenticate the XDMC.

```
HTTP/1.1 401 Unauthorized
Server: XDM-proxy/OMA1.0
Date: Thu, 08 Jan 2004 10:50:35 GMT
WWW-Authenticate: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c", qop=auth-
   int
Content-Length: 0
```

3) The XDMC sends a HTTP GET request including the Authorization header.

```
GET http://xcap.example.com/services/resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA1.0
Date: Thu, 08 Jan 2004 10:50:37 GMT
Authorization: Digest realm="xcap.example.com", nonce="47364c23432d2e131a5fb210812c",
   username="u29502566@example.com", qop=auth-int,
   uri="http://xcap.example.com/services/resource-lists/users/sip:joebloggs@example.com/index",
   response="2c8ee200cec7f6e966c932a9242554e4", cnonce="dcd99agsfgfsa8b7102dd2f0e8b1", nc=00000001
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Accept-Encoding: gzip
Content-Length: 0
```

4) Based on the AUID the Aggregation Proxy forwards the request to appropriate XDMS.

```
GET http://xcap.example.com/services/resource-lists/users/sip:joebloggs@example.com/index HTTP/1.1
Host: xcap.example.com
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
User-Agent: XDM-client/OMA1.0
Date: Thu, 08 Jan 2004 10:50:37 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Content-Length: 0
```

Note: If the "X-3GPP-Intended-Identity" is not included in the message (3), the Aggregation Proxy will include the "X-3GPP-Asserted-Identity" header.

5) After the XDMS has performed the necessary authorisation checks on the request originator, the XDMS sends an HTTP "200 OK" response including the requested document in the body.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:50:39 GMT
Etag: "eti87"
Content-Type: application/resource-lists+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list name="friends">
    <entry uri="sip:hermione.blossom@example.com"/>
    <entry uri="tel:5678;phone-context=+43012349999"/>
  </list>
</resource-lists>
```

6) The Aggregation Proxy encodes (optionally) the content and routes the response back to the XDM Client.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Via: HTTP/1.1 proxy.example.com (Apache/1.1)
Date: Thu, 08 Jan 2004 10:50:39 GMT
Authentication-Info: nextnonce="e966c32a924255e42c8ee20ce7f6"
Etag: "eti87"
Content-Encoding: gzip
Content-Type: application/resource-lists+xml
Content-Length: (...)

(binary data)
```

B.2 Sample XCAP message flow

Example B.2 describes the message flows used to manipulate an XML document in an XDMS after authentication.

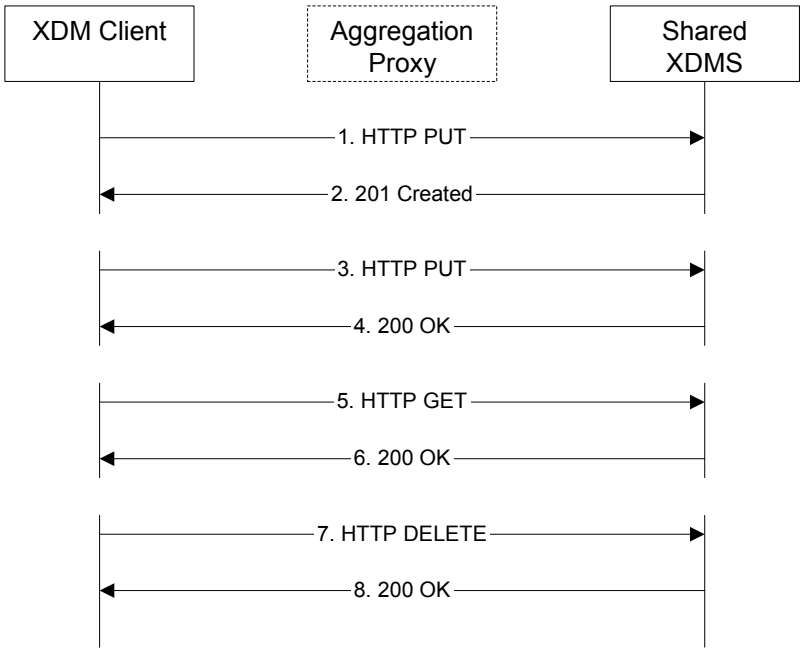

Figure B.2- XDM Client manipulating an XML document

NOTE: The request messages (1,3,5,7) are shown in one diagram for the convenience of the reader, but there is no implication that all of them have to be performed.

NOTE: The Aggregation Proxy is not shown in the flow diagram as its omission does not affect the content of the exchanged messages. The flow diagram also does not show the authentication headers and other HTTP headers not necessary to illustrate the XCAP functionality.

1) The XDMC sends an XCAP(HTTP) PUT request to create a new URI list document "friends.xml" for the user with a public SIP URI of "sip:joebloggs@example.com" in the (Shared) XDMS in the example.com domain.

```
PUT http://xcap.example.com/services/resource-lists/users/sip:joebloggs@example.com/friends.xml
   HTTP/1.1
…
Content-Type: application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list name="My_friends">
    <entry uri="sip:friend1@example.com">
      <display-name>Friend1</display-name>
    </entry>
  </list>
</resource-lists>
```

2) The XDMS acknowledges the creation of the friends.xml document with a XCAP(HTTP) 201 Created message, assuming that the XDMC had the necessary authorisation to perform the operation, and the operation was successful.

```
HTTP/1.1 201 CREATED
Etag: "cdcdcdcd"
…
Content-Length: 0
```

3) The XDMC sends a XCAP(HTTP) PUT request to the just-created "friends.xml" document in "sip:joebloggs@example.com"'s home directory to add a new <entry> sub-element to the <list> element identified as "My_friends".

```
PUT http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/friends.xml/~~/resource-
   lists/list[@name="My_friends"]/entry[@uri="sip:friend2@example.com"] HTTP/1.1
…
Content-Type: application/xcap-el+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
 <entry uri="sip:friend2@example.com">
   <display-name>Friend2</display-name>
 </entry>
```

Note: The use of the Content Type "application/xcap-el+xml".

4) The XDMS acknowledges the addition of new elements to the list with an XCAP(HTTP) "200 OK" reply.

```
HTTP/1.1 200 OK
Etag: "efefefef"

…
Content-Length: 0
```

5) The XDMC sends an XCAP(HTTP) GET request to retrieve "sip:joebloggs@example.com"'s "friends" list from the (Shared) XDMS.

```
GET http://xcap.example.com/services/resource-lists/users/sip:joebloggs@example.com/friends.xml
   HTTP/1.1
Content-Length: 0
```

6) The XDMS returns the list to the XDMC in the body of an XCAP(HTTP) “200 OK” message.

```
HTTP/1.1 200 OK
…
Etag: "abababˮ
Content-Type:application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
 <list name="My_friends>
  <entry uri="sip:friend1@example.com">
   <display-name>Friend1</display-name>
    </entry>
    <entry uri="sip:friend2@example.com">
    <display-name>Friend2</display-name>
  </entry>
</list>
</resource-lists>
```

7) The XDMC sends an XCAP(HTTP) DELETE request to delete an <entry> identified by the URI “sip:friend2@example.com” from sip:joebloggs@example.com” ‘s “My_friends” list in the Shared XDMS.

```
DELETE http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/friends.xml~~/resource-
   lists/list[@name="My_friends"]/entry[@uri=ˮsip:friend2@example.comˮ] HTTP/1.1
Content-length: 0
```

The XDMS, after checking the privileges of the principal, performs the deletion.

8) The XDMS acknowledges the deletion of the “friend2” element from the list with an XCAP(HTTP) 200 OK.

```
HTTP/1.1 200 OK
…
Content-Length: 0
```

B.3 Sample XCAP Directory Retrieval Operation of all user documents

Figure B.3 describes how an XCAP operation is performed to retrieve all of a user’s documents for all application usages. For simplicity, only two XDMSes are shown and the authentication steps are omitted.

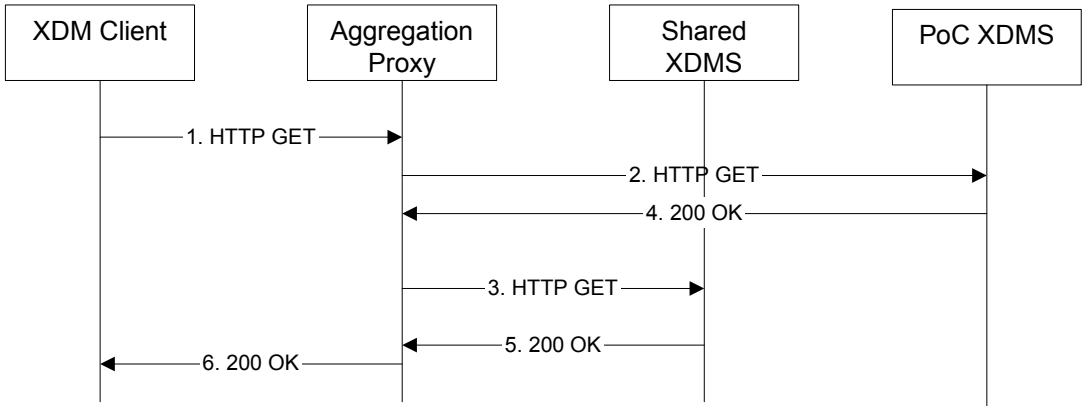

Figure B.3- Sample XCAP Directory retrieval operation

The details of the flows are as follows:

7) The user "sip:joebloggs@example.com" wants to obtain a list of all his documents stored in all XDMSes. For this purpose the XDMC sends a HTTP GET request to the Aggregation Proxy.

```
GET http://xcap.example.com/services/org.openmobilealliance.xcap-
   directory/users/sip:joebloggs@example.com/directory.xml HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA1.0
Date: Thu, 08 Jan 2004 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Content-Length: 0
```

8) The Aggregation proxy forwards the HTTP GET from step 1) to the PoC XDMS.

9) The Aggregation proxy forwards the HTTP GET from step 1) to the Shared XDMS.

10) The PoC XDMS returns the "directory.xml" document containing a list of all the PoC Group documents belonging to sip:joebloggs@example.com in a HTTP 200 OK response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:50:39 GMT
Etag: "eti87"
Content-Type: application/oma-directory+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:params:xml:ns:xcap-directory"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <folder auid=poc-groups>
    <entry uri="http://xcap.example.com/services/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/skiing" etag="abc123"/>
    <entry uri=http://xcap.example.com/services/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/shopping etag="def456"/>
  </folder>
</xcap-directory>
```

where each <entry> element lists a document containing one of sip:joebloggs@example.com's PoC Groups called "skiing" and "shopping" in this example.

11) The Shared XDMS returns the "directory.xml" document containing the URI lists belonging to sip:joebloggs@example.com in a HTTP 200 OK response

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:51:44 GMT
Etag: "eti99"
Content-Type: application/oma-directory+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:params:xml:ns:xcap-directory"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <folder auid=resource-lists>
    <entry uri=http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/friends etag="pqr999"/>
    <entry uri=http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/colleagues etag="xyz123"/>
  </folder>
</xcap-directory>
```

where each <entry> element lists one document corresponding to each of sip:joebloggs@example.com's URI lists, called "friends" and "colleagues" in this example.

12) The Aggregation Proxy returns the consolidated "directory.xml" document to the user in a HTTP 200 OK response.

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:55:39 GMT
Etag: "eti101"
Content-Type: application/oma-directory+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:params:xml:ns:xcap-directory"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <folder auid=resource-lists>
    <entry uri=http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/friends etag="pqr999"/>
    <entry uri=http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/colleagues etag="xyz123"/>
  </folder>
  <folder auid=poc-groups>
    <entry uri=http://xcap.example.com/services/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/skiing etag="abc123"/>
    <entry uri=" http://xcap.example.com/services/org.openmobilealliance.poc-
   groups/users/sip:joebloggs@example.com/shopping" etag="def456"/>
  </folder>
</xcap-directory>
```

B.4 Sample XCAP Directory Retrieval Operation of specific user documents

Figure B.4 describes how an XCAP operation is performed to retrieve all of a user's documents corresponding to a particular application usage. For simplicity, the authentication steps are omitted.

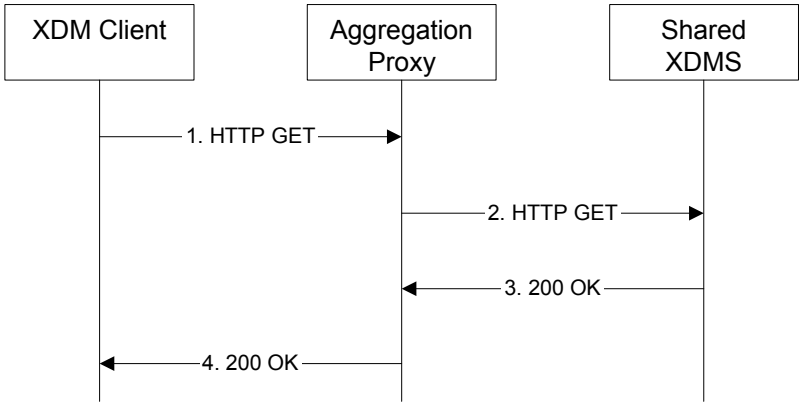

Figure B.4- Sample XCAP Directory retrieval operation from a particular XDMS

The details of the flows are as follows:

1) The user "sip:joebloggs@example.com" wants to obtain a list of all his documents (URI lists) stored in the Shared XDMS. For this purpose the XDMC sends a HTTP GET request to the Aggregation Proxy.

```
GET http://xcap.example.com/services/org.openmobilealliance.xcap-
   directory/users/sip:joebloggs@example.com/directory.xml/~~/xcap-
   directory/folder[@auid="resource-lists"] HTTP/1.1
Host: xcap.example.com
User-Agent: XDM-client/OMA1.0
Date: Thu, 08 Jan 2004 10:50:33 GMT
X-3GPP-Intended-Identity: "sip:joebloggs@example.com"
Content-Length: 0
```

2) The Aggregation proxy forwards the HTTP GET from step 1) to the Shared XDMS.

3) The Shared XDMS responds with a HTTP 200 OK including the directory document containing the URI lists belonging to sip:joebloggs@example.com

```
HTTP/1.1 200 OK
Server: XDM-serv/OMA1.0
Date: Thu, 08 Jan 2004 10:55:39 GMT
Etag: "eti101"
Content-Type: application/oma-directory+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<xcap-directory xmlns="urn:oma:params:xml:ns:xcap-directory"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <folder auid=resource-lists>
    <entry uri=" http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/friends" etag="pqr999"/>
    <entry uri=" http://xcap.example.com/services/resource-
   lists/users/sip:joebloggs@example.com/colleagues" etag="xyz123"/>
  </folder>
</xcap-directory>
```

4) The Aggregation proxy returns the same entity body as in step 3 to the XDMC is a HTTP 200 OK message.

Appendix C. XDMC Provisioning (Normative)

This appendix specifies the parameters that are needed by the XDM Client. Existing parameters in [Provisioning Content] and [OMA DM] are re-used; those without corresponding parameters are defined and to be registered in OMNA through OMA official registration process.

C.1 Provisioned XDMC Parameters

The parameters listed in the table below are needed for XDM client provisioning:

ID	Name	Description	Mandatory (M) /Optional (O)
1	Application identity	Uniquely identifies the application	M
2	Application name	User displayable name for the XML Document Management service	M
3	Provider -ID	Identity of the XDM service provider	O
4	Network Access Definitions	Reference to the connection used for the XCAP traffic.	M
5	XDM reference to SIP/IP Core	Reference to the SIP/IP core. Used to access XDM servers using the referenced SIP/IP core.	M
6	XCAP Root URI	The root of all XDM resources (which points to the Aggregation Proxy address). This is used when accessing via XCAP.	M
7	XCAP Authentication user name	HTTP digest “username”, for accessing an XDMS using the XCAP protocol	O
8	XCAP Authentication password	HTTP digest password	O
9	XCAP Authentication type	Authentication method for XDMS over XCAP	O

NOTE: The parameters ”XCAP Authentication username” and “XCAP Authentication password” are not needed if GAA is used in a 3GPP/3GPP2 realization.

In addition, there may be enabler-specific parameters related to XDMC that are described in separate specifications.

One type of provisioned parameter having a reusable structure is a URI Template. A URI Template is used to describe a single syntax for a URI (e.g. Conference URI of a PoC Group), so that the XDM Client can autonomously generate a URI that complies with local policy and uniqueness constraints. It is up to separate specifications to define provisioned parameters that make use of a URI Template.

A URI Template SHALL describe a URI as defined in [RFC3986]. The template contains a sequence, in any order, of:

a. unreserved characters according to [RFC3986], and
b. substitution tags enclosed in “< >”brackets.

The XDM Client SHALL support the following substitution tags:

<id> : The XDM Client SHALL replace this tag with a unique identifier, generated by the XDM Client.

<user> : The XDM Client SHALL replace this tag with the user part of the XUI.

<xui> : The XDM Client SHALL replace this tag with the XUI.

NOTE: the XUI is a Public SIP URI [RFC3261].

Illustrative examples of URI templates are shown in Table X.

Example URI Template	Example URI generated from template
sip:<id>@example.com	sip:abc123@example.com
sip:<id>_<user>@example.com	sip:abc123_joe@example.com
<xui>;poc-group=<id>	sip:joe@example.com;poc-group=abc123

Table X: Example usages of URI Templates

C.2 Initial Provisioning document

This chapter defines the provisioning document structure as described in [Provisioning Content].

The following table lists the parameters available in an instance of the XDM Application Characteristic

Parameter Name	Req / Opt	Instances	Default
Standard Application Characteristic fields as defined in [Provisioning Content]			
APPID	Required	1	"XDMS"
PROVIDER-ID	Optional	0 or 1	none
TO-APPREF	Required	1	n/a
NAME	Required	1	n/a
TO-NAPID	Required	1 or more	n/a
URI	Required	1	n/a
AAUTHNAME	Optional	0 or 1	n/a
AAUTHSECRET	Optional	0 or 1	n/a
AAUTHTYPE	Optional	0 or 1	n/a

The provisioning document in an AC file format

```
IDENTIFYING INFORMATION
########################
APPID: xx.
APPID type: OMNA.
Owner: OMA Presence and Availability Working Group.
Contact: OMA Presence and Availability Working Group <TECHNICAL-
COMMENTS@MAIL.OPENMOBILEALLIANCE.ORG>.
Registration version: 1.0.
Registration timestamp: 2004-12-xx.
Application description: XDM.
Application reference: XML Document Management(XDM) enabler. OMA XDM Enabler
Release 1.0 specifications, URL:http://www.openmobilealliance.org/documents.asp.

Legal text:
Use of this document is subject to all of the terms and conditions of the Use
Agreement located at http://www.openmobilealliance.org/UseAgreement.html.
```

WELL-KNOWN PARAMETERS
######################
Characteristic/name: APPLICATION/APPID.
Status: Required.
Occurs: 1/1.
Default value: None.
Used values: xx.
Interpretation: To uniquely identify the XDM enabler.

Characteristic/name: APPLICATION/PROVIDER-ID.
Status: Optional.
Occurs: 0/1.
Default value: None.
Used values: N/A.
Interpretation: Identity of the XDM service provider.

Characteristic/name: APPLICATION/TO-APPREF.
Status: Required.

```
Occurs: 1/1.
Default value: None.
Used values: N/A.
Interpretation: It specifies the linkage between XDM and the SIP-IP-core, e.g. IMS.
-------
Characteristic/name: APPLICATION/NAME.
Status: Required.
Occurs: 1/1.
Default value: None.
Used values: N/A.
Interpretation: User displayable name for the XML Document Management enabler.
-------
Characteristic/name: APPLICATION/TO-NAPID.
Status: Required if direct use of Network Access Point supported.
Occurs: 1/*.
Default value: None.
Used values: N/A.
Interpretation: specifies the network access point used for a given application.
-------
Characteristic/parameter: RESOURCE/URI.
Status: Required.
Occurs: 1/1.
Default value: None.
Used values: A HTTP URL.
Interpretation: Identifies the ROOT URI of the documents under the global XCAP
document tree managed by the XDM server.
-------
Characteristic/parameter: RESOURCE/AAUTHNAME.
Status: Optional.
Occurs: 0/1.
Default value: None.
Used values: String.
Interpretation: HTTP user name, for accessing XDM over XCAP.
-------
Characteristic/parameter: RESOURCE/AAUTHSECRET.
Status: Optional.
Occurs: 0/1.
Default value: None.
Used values: String.
Interpretation: HTTP digest password, for accessing XDM over XCAP.
-------
Characteristic/name: RESOURCE/AAUTHTYPE.
Status: Optional.
Occurs: 0/1.
Default value: None.
Used values: "HTTP-DIGEST". Value set can be extended with new values in future.
Interpretation: Authentication method for XDM over XCAP.
----------


EXAMPLE
#######
<characteristic type="APPLICATION">
  <parm name="APPID" value="XDM"/>
  <parm name=”PROVIDER-ID” value="Best"/>
  <parm name="NAME" value="XDM"/>
  <parm name="TO-APPREF" value="SIP-IP-CORE"/>
  <parm name="TO-NAPID" value="IMS-NAP"/>
```

```
  <characteristic type="RESOURCE">
    <parm name="URI" value="http://xcap.example.com/services"/>
    <parm name="AAUTHNAME" value="httpusername"/>
    <parm name="AAUTHSECRET" value="httpdigestpasswd"/>
    <parm name="AAUTHTYPE" value="HTTP-DIGEST"/>
  </characteristic>
</characteristic>
###END###
```

C.3 Continuous provisioning based on SyncML

The present section defines a mobile device Management Object (MO) for OMA XDM. The MO is used for continuous provisioning, which allows the service provider to update any parameter defined in MO tree for service configurations during service deployment. Also the AC file SHALL use the same list of parameters for initial provisioning.

The OMA XDM Management Object consists of relevant parameters required by [XDM RD]. It defined using the OMA DM Device Description Framework as described in [OMA-SyncML-DMTND-V1-1-2] and [OMA-SyncML-DMStdObj-V1-1-2].

The Management Object Identifier is: org.openmobilealliance/1.0/XDM

Protocol compatibility: This MO is compatible with OMA DM 1.1.2.

Management object name: OMA_XDM

C.3.1 OMA PAG Management Object tree

Figure C.1 shows the nodes and leaf objects for XDM continuous provisioning:

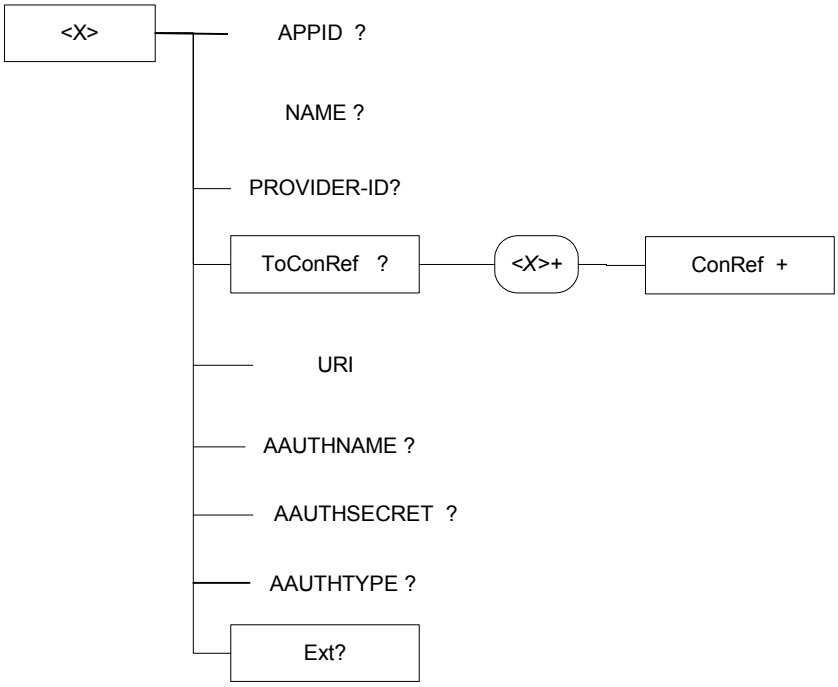

Figure C.1: The SynchML DM object for OMA XDM

C.3.2 Management Object parameters

This clause describes the parameters for the OMA XDM Management Object.

Node: /<X>

This interior node acts as a placeholder for one or more accounts for a fixed node.

Occurrence: OneOrMore
Format: Node
Access Types: Get
Values: N/A

The interior node is mandatory if the UE supports OMA XDM.

Editor note: The value of the <X> node and the base SyncML URI is to be determined by OMNA.

/<X>/APPID/

The APPID is the identity of the application service available at the described application service access point. The value is globally unique.

Occurrence: ZeroOrOne
Format: chr
Access Types: Get
Values: <Globally unique value>

Editor's Note: The value of the <X>/APPID is to be determined by OMNA .

/<X>/NAME/

The Name leaf is the application name, which is to be displayed in the user's equipment. It is specific for each service provider.

Occurrence: ZeroOrOne
Format: chr
Access Types: Get
Values: <User displayable name>

/<X>/PROVIDER-ID/

This parameter provides an identifier for the application service access point described by an APPLICATION characteristic.

Occurrence: ZeroOrOne
Format: chr
Access Types: Get
Values: N/A

/<X>/ToConRef

The ToConRef interior node is used to allow an application to refer to a collection of connectivity definitions. Several connectivity parameters may be listed for a give application under this interior node.
This parameter provides an identifier for the application service access point described by an APPLICATION characteristic.
Occurrence: ZeroOrOne
Format: node
Access Types: Get
Values: N/A

/<X>/ToConRef/<X>

This run-time node acts as a placeholder for one or more connectivity parameters.

Occurrence: OneOrMore
Format: Node
Access Type: Get
Value: N/A

/<X>/ToConRef/<X>/ConRef

The ConRef indicates the linkage to connectivity parameters. This parameter provides an identifier for the application service access point described by an APPLICATION characteristic, in this case the NAP ID and the SIP/IP core.

.

Occurrence: OneOrMore
Format: chr
Access Types: Get
Values: <A String>

/<X>/URI/

This parameter defines the root of all XDM resources (this is the Aggregation Proxy address). This is useful when accessing via XCAP.

Occurrence: One
Format: chr
Access Types: Get
Values: <a HTTP URI>

/<X>/AAUTHNAME/

This parameter defines the user name for XDMC authentication using HTTP digest.

Occurrence: ZeroOrOne
Format: chr
Access Types: Get
Values: N/A

/<X>/AAUTHSECRET/

This parameter defines the password for XDMC authentication using HTTP digest.

Occurrence: ZeroOrOne
Format: chr
Access Types: Get
Values: <a User specific value>

/<X>/AAUTHTYPE/

This parameter defines the authentication type for XDMC authentication.

Occurrence: ZeroOrOne
Format: chr
Access Types: Get
Values: <a token>
GAA: the authentication method will be GAA

Digest: the authentication method will be HTTP Digest.

/<X>/Ext/

The Ext is an interior node where the vendor-specific information about the XDM MO is placed (vendor means application vendor, device vendor etc.). Usually the vendor extension is identified by a vendor-specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include a non-standardized sub-tree.

Occurrence: ZeroOrOne
Format: node
Access Types: Get
Values: N/A

Appendix D. Change History (Informative)

D.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

D.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Version OMA-XDM_Specification-V1_0	27 Sept 2004	All	Template created
	28 Sept 2004	All	Initial content created for discussion in WG
	29 Sept 2004	All	Incorporated contents of contribution OMA-PAG-2004-0507R01-Initial-text-for-XDM-Specification
	8 Oct 2004	Appendix B	Incorporated accepted contribution OMA-PAG-2004-0524R02-XCAP-operations-signalling-flow
	15 Oct 2004	6.6	Incorporated OMA-PAG-2004-0545-Changes-to-Section-6.6-of-XDM-Spec
		6.4.1	Incorporated OMA-PAG-2004-0533R01-XDM-CR-DigestAuthentication
	22 Oct 2004	3.2, 3.3, 6.1.1.1, Appendix B	Incorporated OMA-PAG-2004-0573R01-XDM-Spec-update-to-include-XUI-details
		2.1, 6.1.2	Incorporated OMA-PAG-2004-0561R01-XDMC-SubscribingToChanges
		6.2.2	Incorporated OMA-PAG-2004-0562R01-XDMS-SubscribingToChanges
	28 Oct 2004	6.1.1.2.3	Incorporated OMA-PAG-2004-0600-XDM-Spec-Editorial-comments-section-6.1
	29 Oct 2004	3.2	Incorporated OMA-PAG-2004-0594R01-XDM-Specs-Definition-of-Global-Documents
		Appendix A	Incorporated OMA-PAG-2004-0589R01-GM-XDM-SCR
		6.1.1.2, 6.2.1	Incorporated OMA-PAG-2004-0602R01-XDM-Spec-version-control
		2.1, 5.3, 6.3, 6.4	Incorporated OMA-PAG-2004-0574R03-XDM-AggregationProxy-GM
	2 Nov 2005	6.2.1	Incorporated OMA-PAG-2004-0603R02-XDM-Spec-server-multiple-access
	8 Nov 2005	6.2.2.1	Incorporated OMA-PAG-2004-0667-XDM-Spec-error-code
		New Appendix C	Incorporated OMA-PAG-2004-0627R01-XDMS-Annex-client-provisioning
	11 Nov 2005	Appendix C	Incorporated OMA-PAG-2004-0682-LATE-XDM_Spec_Inconsistency and OMA-PAG-2004-0681-LATE-XDM_Spec_SCR_Inconsistency
		6.1.2.1	Incorporated OMA-PAG-2004-0677-XDM-Spec-providing-AUID-when-subscribing and OMA-PAG-2004-0678-XDM-Spec-subscribing-based-on-group-uri and OMA-PAG-2004-0687R01-XDM-no-single-subscription
		6.2.2.1, 6.4.2	Incorporated OMA-PAG-2004-0676R01-XDM-Spec-auth-subscribing-to-changes
		Appendix C	Incorporated OMA-PAG-2004-0684R01-PAG-XDM-provisioning-MO-file
	12 Nov 2005	Throughout	Editorial corrections submitted by Nicolas and Ajith
		Appendix B	Incorporated 0666R02-XDM-Comprehensive-Example
		Section 5	Incorporated 0651R01-Server-to-XDMS
		Appendix A.2	Incorporated 0698-XDM-Spec-Error-Code-scr

Document Identifier	Date	Sections	Description
	15 Nov 2005	Sections 2.1 6.1.1.2 and 6.3	Incorporated OMA-PAG-2004-0720R01-XDM-Compression
		Section 6.4	Incorporated OMA-PAG-2004-0718R01-LATE-XDM-Spec-authentication-AP-only
		Section 6.4	Incorporated OMA-PAG-2004-0717R01-LATE-XDM-Spec-confidentiality-heading
		Section 2.1 and Appendix C	Incorporated OMA-PAG-2004-0688R02-XDM-WAP-provisioning
		Appendix A	Incorporated OMA-PAG-2004-0726-XDM-Spec-SCR-changes
		Sec 2.2, new Section 6.6	Incorporated OMA-PAG-2004-0724R01-XDM-CommonPolicy-Extensions
	17 Nov 2005	Section 6.6.1.2	Incorporated OMA-PAG-2004-0751-XDM-Spec-anyURI
		6.5	Incorporated OMA-PAG-2004-0742-Removing-a-TBD-from-the-XDM-Spec
		6.6	Incorporated OMA-PAG-2004-0752R01-XDM-directory
		5.1, 6.2	Incorporated OMA-PAG-2004-0747R02-IMS-MMD-references-XDM-spec
		Appendix A	Incorporated OMA-PAG-2004-0764-More-XDM-Spec-SCR-changes
		Acronyms, References	Added/updated some acronyms and references
	18 Jan 2005	2.1, 2.2	Included Comments 6.015, 6.106, 6.022, 6.034, plus various editorials
		3	Included additional acronyms proposed in editorial comments
		4	Comment 6.004, 6.079
		5.3	Included comments 6.001, 008, 059
		6.1.1.1	Included comment 008,
		6.1.2.1	Included comments 011
		6.2.1.1	Comment 062
		6.2.2.1	Comment 022,
		6.2.2.2	Comment 023
		6.3.2	Comment 001,
		6.3.3	Comment 040
		6.3.4	Comment 041
		6.4.1	Comment 025,
		6.4.2	Comment 007
		6.4.3	Comment 001, 026
		6.5	Comment 027, 046
		6.6.1.2	Comment 070
		6.7	Comment 068
		A	Comments 028, 029, 031, 047, 049, 072
		B	Comments 002, 075, 076
		C	Comment 053
		6.1.2.1, 6.2.2.1	Incorporated contribution PAG-2004-834R04
		6.4.1	Incorporated contribution PAG-2005-0029R01 with a minor editorial amendment to make the sentence grammatically correct
		6.7.2.7	Incorporated PAG-2005-0005
		5.1/5.2	Incorporated PAG-2005-0009, with corrected chapter reference in section 5.2

Document Identifier	Date	Sections	Description
		C.3	Incorporated PAG-2004-821R03 but this needs to be carefully checked as the source text for 821R03 is not based on the Nov 18 (consistency review) version but an earlier draft.
		C.2	Incorporated PAG-822R01 and PAG-2005-0008
	27 Jan. 05	6.1.2.2, 6.2.2.1 6.2.2.2	Incorporated contribution 2005-0011R01 which addresses comment 6.060
		Title page, 2.1, 2.2	Updated XDM document references to conform to the process document
		1, 6.4.3 and Appendix C	Incorporated contribution 2005-0035R01
		6.4.1, B.1	Incorporated contribution 2004-839R01
		C	Various editorial corrections under Tao's guidance, to make the text conform to 2004-821R03
		6.4.1	Addressed comment 6.066 based on new text – see CONRR
		6.6.1.1	Incorporated contribution 2005-0059R01
		6.6.1	Incorporated contribution 2004-0783R01
		6.5	Incorporated contribution 0068 addresseing comment 6.065
		5.2, 6.4.3	Incorporated contribution 0069 addressing comment 6.005
		A.1	Incorporated contribution 0071
		2.2, Appendix C	Various changes as described in 0070
		6.1	Incorporated 0077 addressing 5.025
		6.6.1	Incorporated 0067R02
		Appendix	Incorporated 0078
		Table A.2	Comment 6.030
		6.4.3	Contribution 0098R01
		6.4.3, 6.8	Contribution 0095
		6.1.1.1, 2.1, 6.1.2.1	Contribution 0094R01
Candidate Version OMA-TS-XDM_Core-V1_0	4 Feb 2005	n/a	Status changed to Candidate by TP: OMA-TP-2005-0060-XDM_1_0--for-candidate-approval
OMA-TS-XDM_Core-V1_0	15 Apr 2005	2.1, 6.7.1, 6.7.2.3 2.1, C.1 6.6 6.1, 6.7 and B.1 6.7 and B	Incorporated CR 2005-0191R01 Incorporated CR 2005-0279R02 Incorporated CR 2005-0286R02 Incorporated CR 2005-0291R01 Incorporated CR 2005-0302