



RESTful Network API for Anonymous Customer Reference Management

Candidate Version 1.0 – 01 Dec 2015

Open Mobile Alliance
OMA-TS-REST_NetAPI_ACR-V1_0-20151201-C

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>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2015 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE	6
2. REFERENCES	7
2.1 NORMATIVE REFERENCES	7
2.2 INFORMATIVE REFERENCES	8
3. TERMINOLOGY AND CONVENTIONS	9
3.1 CONVENTIONS	9
3.2 DEFINITIONS	9
3.3 ABBREVIATIONS	9
4. INTRODUCTION	10
4.1 VERSION 1.0	10
5. ANONYMOUS CUSTOMER REFERENCE MANAGEMENT API DEFINITION	11
5.1 RESOURCES SUMMARY	11
5.2 DATA TYPES	14
5.2.1 XML Namespaces.....	14
5.2.2 Structures	14
5.2.2.1 Type: AcrList.....	14
5.2.2.2 Type: Acr	15
5.2.2.3 Type: Status.....	17
5.2.3 Enumerations	17
5.2.3.1 Enumeration: AcrStatus.....	17
5.2.4 Values of the Link “rel” attribute.....	17
5.3 SEQUENCE DIAGRAMS	17
5.3.1 Operations on Anonymous Customer References	17
6. DETAILED SPECIFICATION OF THE RESOURCES	20
6.1 RESOURCE: END USER’S APPLICATION	20
6.1.1 Request URL variables	21
6.1.2 Response Codes and Error Handling	21
6.1.3 GET.....	21
6.1.3.1 Example 1: Retrieve the issued ACR using “auth” keyword (Informative).....	21
6.1.3.1.1 Request.....	21
6.1.3.1.2 Response.....	22
6.1.3.2 Example 2: Retrieve the issued ACR using MSISDN (Informative)	22
6.1.3.2.1 Request.....	22
6.1.3.2.2 Response.....	22
6.1.3.3 Example 3: Retrieve a non-existent ACR using MSISDN (Informative).....	22
6.1.3.3.1 Request.....	22
6.1.3.3.2 Response.....	23
6.1.4 PUT.....	23
6.1.5 POST.....	23
6.1.5.1 Example 1: Create an ACR using “auth” keyword with expiry value included in OAuth access token (Informative).....	23
6.1.5.1.1 Request.....	23
6.1.5.1.2 Response.....	24
6.1.5.2 Example 2: Create an ACR using MSISDN (Informative)	24
6.1.5.2.1 Request.....	24
6.1.5.2.2 Response.....	24
6.1.5.3 Example 3: Create an ACR over mobile network authenticated connection, response with a location of created resource (Informative).....	25
6.1.5.3.1 Request.....	25
6.1.5.3.2 Response.....	25
6.1.5.4 Example 4: Create ACR over mobile network authenticated connection, response with creation failure due to an existing expired ACR (Informative).....	25
6.1.5.4.1 Request.....	25
6.1.5.4.2 Response.....	26
6.1.5.5 Example 5: Create an ACR over mobile network authenticated connection, response with a copy of created resource (Informative).....	26

- 6.1.5.5.1 Request..... 26
- 6.1.5.5.2 Response..... 26
- 6.1.6 DELETE 26
- 6.2 RESOURCE: ANONYMOUS CUSTOMER REFERENCE..... 27**
- 6.2.1 Request URL variables 28
- 6.2.2 Response Codes and Error Handling 28
- 6.2.3 GET..... 28
- 6.2.3.1 *Example 1: Retrieve ACR data (Informative)* 28
- 6.2.3.1.1 Request..... 28
- 6.2.3.1.2 Response..... 28
- 6.2.4 PUT..... 28
- 6.2.5 POST..... 29
- 6.2.6 DELETE 29
- 6.2.6.1 *Example 1: Remove an ACR (Informative)* 29
- 6.2.6.1.1 Request..... 29
- 6.2.6.1.2 Response..... 29
- 6.3 RESOURCE: ACR STATUS..... 29**
- 6.3.1 Request URL variables 29
- 6.3.2 Response Codes and Error Handling 29
- 6.3.3 GET..... 29
- 6.3.3.1 *Example 1: Read the Status of an ACR using “auth” keyword (Informative)*..... 30
- 6.3.3.1.1 Request..... 30
- 6.3.3.1.2 Response..... 30
- 6.3.3.2 *Example 2: Read the Status of a non-existent ACR using “auth” keyword (Informative)* 30
- 6.3.3.2.1 Request..... 30
- 6.3.3.2.2 Response..... 30
- 6.3.4 PUT..... 30
- 6.3.4.1 *Example 1: Refresh an expired ACR by using “auth” keyword (Informative)*..... 31
- 6.3.4.1.1 Request..... 31
- 6.3.4.1.2 Response..... 31
- 6.3.4.2 *Example 2: Refreshing a revoked ACR is rejected (Informative)*..... 31
- 6.3.4.2.1 Request..... 31
- 6.3.4.2.2 Response..... 32
- 6.3.5 POST..... 32
- 6.3.6 DELETE 32
- 7. FAULT DEFINITIONS 33**
- 7.1 SERVICE EXCEPTIONS..... 33**
- 7.1.1 SVC1005: ACR creation failed. Unknown userId 33
- 7.1.2 SVC1006: ACR not found 33
- 7.2 POLICY EXCEPTIONS 33**
- 7.2.1 POL1024: ACR creation failed. An active ACR already exists..... 33
- 7.2.2 POL1025: ACR creation failed. An expired ACR already exists 34
- 7.2.3 POL1026: Creation of Static ACR is not supported 34
- 7.2.4 POL1027: Revoked ACR can no longer be used or refreshed..... 34
- 7.2.5 POL1028: Expired ACR may not be used unless refreshed 34
- APPENDIX A. CHANGE HISTORY (INFORMATIVE)..... 35**
- A.1 APPROVED VERSION HISTORY 35**
- A.2 DRAFT/CANDIDATE VERSION 1.0 HISTORY 35**
- APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)..... 37**
- B.1 SCR FOR REST.ACRMGT SERVER 37**
- B.1.1 SCR for REST. ACRMgt.UsersApp Server 37
- B.1.2 SCR for REST. ACRMgt.ACR Server 37
- B.1.3 SCR for REST. ACRMgt.ACR.Status Server 37
- APPENDIX C. APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR POST OPERATIONS (NORMATIVE) 38**
- APPENDIX D. JSON EXAMPLES (INFORMATIVE) 39**

D.1 RETRIEVE THE ISSUED ACR USING “AUTH” KEYWORD (SECTION 6.1.3.1).....39

D.2 RETRIEVE THE ISSUED ACR USING MSISDN (SECTION 6.1.3.2)39

D.3 RETRIEVE A NON-EXISTENT ACR USING MSISDN (SECTION 6.1.3.3).....40

D.4 CREATE AN ACR USING “AUTH” KEYWORD WITH EXPIRY VALUE INCLUDED IN OAUTH ACCESS TOKEN (SECTION 6.1.5.1).....41

D.5 CREATE AN ACR USING MSISDN (SECTION 6.1.5.2)41

D.6 CREATE AN ACR OVER MOBILE NETWORK AUTHENTICATED CONNECTION, RESPONSE WITH A LOCATION OF CREATED RESOURCE (SECTION 6.1.5.3).....42

D.7 CREATE ACR OVER MOBILE NETWORK AUTHENTICATED CONNECTION, RESPONSE WITH CREATION FAILURE DUE TO AN EXISTING EXPIRED ACR (SECTION 6.1.5.4).....42

D.8 CREATE AN ACR OVER MOBILE NETWORK AUTHENTICATED CONNECTION, RESPONSE WITH A COPY OF CREATED RESOURCE (SECTION 6.1.5.5)43

D.9 RETRIEVE ACR DATA (SECTION 6.2.3.1)43

D.10 REMOVE AN ACR (SECTION 6.2.6.1)44

D.11 READ THE STATUS OF AN ACR USING “AUTH” KEYWORD (SECTION 6.3.3.1)44

D.12 READ THE STATUS OF A NON-EXISTENT ACR USING “AUTH” KEYWORD (SECTION 6.3.3.2).....45

D.13 REFRESH AN EXPIRED ACR BY USING “AUTH” KEYWORD (SECTION 6.3.4.1).....45

D.14 REFRESHING A REVOKED ACR IS REJECTED (SECTION 6.3.4.2)46

APPENDIX E. OPERATIONS MAPPING TO A PRE-EXISTING BASELINE SPECIFICATION (INFORMATIVE)47

APPENDIX F. LIGHT-WEIGHT RESOURCES (INFORMATIVE)48

APPENDIX G. AUTHORIZATION ASPECTS (NORMATIVE)49

G.1 USE WITH OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIS.....49

G.1.1 Scope values49

G.1.1.1 Definitions.....49

G.1.1.2 Downscoping49

G.1.1.3 Mapping with resources and methods.....49

G.1.2 Use of ‘acr:auth’51

APPENDIX H. DEFINITION OF THE ACR: URI SCHEME (NORMATIVE)52

H.1 “ACR” URI SYNTAX52

H.2 EXAMPLES (INFORMATIVE)53

H.3 RATIONALE (INFORMATIVE)53

H.3.1 Privacy policies.....53

H.3.2 Cookie support54

H.3.3 Sharing identity.....54

H.3.4 Relation to SIP54

H.4 SECURITY CONSIDERATIONS54

H.5 IANA REGISTRATION TEMPLATE.....55

Figures

Figure 1: Resource structure defined by this specification12

Figure 2: Management of ACR18

Tables

Table 1: Scope values for RESTful Anonymous Customer Reference Management API.....49

Table 2: Required scope values for: Management of Anonymous Customer Reference50

1. Scope

This specification defines a RESTful API for Anonymous Customer Reference Management using HTTP protocol bindings.

2. References

2.1 Normative References

- [**Autho4API_10**] “Authorization Framework for Network APIs”, Open Mobile Alliance™, OMA-ER-Autho4API-V1_0, URL: <http://www.openmobilealliance.org/>
- [**ITU-T E.212**] Recommendation ITU-T E.212, The international identification plan for public networks and subscriptions, International operation – Maritime mobile service and public land mobile service, SERIES E: OVERALL NETWORK OPERATION, TELEPHONE SERVICE, SERVICE OPERATION AND HUMAN FACTORS
- [**RD-REST_NetAPI_ACR**] “OMA RESTful Network API for Anonymous Customer Reference Management Requirements”, Open Mobile Alliance™, OMA-RD-REST_NetAPI_ACR-V1_0, URL: <http://www.openmobilealliance.org/>
- [**REST_NetAPI_Common**] “Common definitions for RESTful Network APIs”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL: <http://www.openmobilealliance.org/>
- [**REST_SUP_ACR**] “XML schema for the RESTful Network API for Anonymous Customer Reference Management”, Open Mobile Alliance™, OMA-SUP-XSD_rest_netapi_acr-V1_0, 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>
- [**RFC2616**] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et. al, January 1999, URL: <http://www.ietf.org/rfc/rfc2616.txt>
- [**RFC3261**] “SIP: Session Initiation Protocol”, J. Rosenberg, et. Al, June 2002, URL: <http://www.ietf.org/rfc/rfc3261.txt>
- [**RFC3323**] “A Privacy Mechanism for the Session Initiation Protocol (SIP)”, Peterson, J., November 2002, URL: <http://tools.ietf.org/html/rfc3323>
- [**RFC3966**] “The tel URI for Telephone Numbers”, H.Schulzrinne, December 2004, URL: <http://www.ietf.org/rfc/rfc3966.txt>
- [**RFC3986**] “Uniform Resource Identifier (URI): Generic Syntax”, R. Fielding et. al, January 2005, URL: <http://www.ietf.org/rfc/rfc3986.txt>
- [**RFC5234**] “Augmented BNF for Syntax Specifications: ABNF”, STD 68, Crocker, D. and P. Overell, January 2008, URL: <http://tools.ietf.org/html/rfc5234.txt>
- [**RFC7159**] “The JavaScript Object Notation (JSON) Data Interchange Format”, T. Bray, Ed., March 2014, <https://tools.ietf.org/html/rfc7159.txt>
- [**SCRRULES**] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL: <http://www.openmobilealliance.org/>
- [**XMLSchema1**] W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures Second Edition, W3C Recommendation 5 April 2012, URL: <http://www.w3.org/TR/xmlschema11-1/>
- [**XMLSchema2**] W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation 5 April 2012, URL: <http://www.w3.org/TR/xmlschema11-2/>

2.2 Informative References

- [IETF_ACR_draft] “The acr URI for anonymous users”, S.Jakobsson, K.Smith, January 2010, URL: <http://tools.ietf.org/html/draft-uri-acr-extension-04>
- [OMADICT] “Dictionary for OMA Specifications”, Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_9, URL: <http://www.openmobilealliance.org/>
- [REST_WP] “Guidelines for RESTful Network APIs”, Open Mobile Alliance™, OMA-WP-Guidelines_for_RESTful_Network_APIs, URL: <http://www.openmobilealliance.org/>
- [RFC2234] “Augmented BNF for Syntax Specifications: ABNF”, D. Crocker and P. Overell, November 1997, URL: <http://www.ietf.org/rfc/rfc2234.txt>
- [RFC4648] “The Base16, Base32, and Base64 Data Encodings”, S. Josefsson, October 2006, URL: <http://www.ietf.org/rfc/rfc4648.txt>

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST 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

For the purpose of this TS, all definitions from the OMA Dictionary apply [OMADICT].

3.3 Abbreviations

ACR	Anonymous Customer Reference
API	Application Programming Interface
APP	Application
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MCC	Mobile Country Code
MIME	Multipurpose Internet Mail Extensions
MSISDN	Mobile Subscriber ISDN Number
OMA	Open Mobile Alliance
REST	REpresentational State Transfer
SCR	Static Conformance Requirements
SIP	Session Initiation Protocol
TS	Technical Specification
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WP	White Paper
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

The Technical Specification of the RESTful Network API for Anonymous Customer Reference Management contains HTTP protocol bindings based on the requirements for Anonymous Customer Reference Management defined in [RD-REST_NetAPI_ACR], using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML, JSON).

Additionally this specification defines the acr: URI scheme. The "acr" URI describes an anonymous reference that can be mapped to a resource or user.

4.1 Version 1.0

Version 1.0 of this specification supports the following operations:

- Create an Anonymous Customer Reference (ACR)
- Query the status of an ACR
- Refresh an ACR

In addition, this specification provides:

- Support for scope values used with authorization framework defined in [Autho4API_10]
- Support for Anonymous Customer Reference (ACR) as an end user identifier
- Support for “acr:auth” as a reserved keyword in a resource URL variable that identifies an end user

5. Anonymous Customer Reference Management API definition

This section is organized to support a comprehensive understanding of the Anonymous Customer Reference (ACR) Management API design. It specifies the definition of all resources, definition of all data structures, and definitions of all operations permitted on the specified resources.

Common data types, naming conventions, fault definitions and namespaces are defined in [REST_NetAPI_Common].

The remainder of this document is structured as follows:

Section 5 starts with a diagram representing the resources hierarchy followed by a table listing all the resources (and their URL) used by this API, along with the data structure and the supported HTTP verbs (section 5.1). What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

Section 6 contains detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP methods, and the supported HTTP verbs. For each supported HTTP verb, a description of the functionality is provided, along with an example of a request and an example of a response. For each unsupported HTTP verb, the returned HTTP error status is specified, as well as what should be returned in the Allow header.

All examples in section 6 use XML as the format for the message body. JSON examples are provided in Appendix D.

Section 7 contains fault definition details such as Service Exceptions and Policy Exceptions.

Appendix B provides the Static Conformance Requirements (SCR).

Appendix E provides the operations mapping to a pre-existing baseline specification, where applicable.

Appendix F provides a list of all light-weight resources, where applicable.

Appendix G defines authorization aspects to control access to the resources defined in this specification.

Appendix H defines the acr: URI scheme

Note: Throughout this document client and application can be used interchangeably.

5.1 Resources Summary

This section summarizes all the resources used by the RESTful Network API for Anonymous Customer Reference Management.

The "apiVersion" URL variable SHALL have the value "v1" to indicate that the API corresponds to this version of the specification. See [REST_NetAPI_Common] which specifies the semantics of this variable.

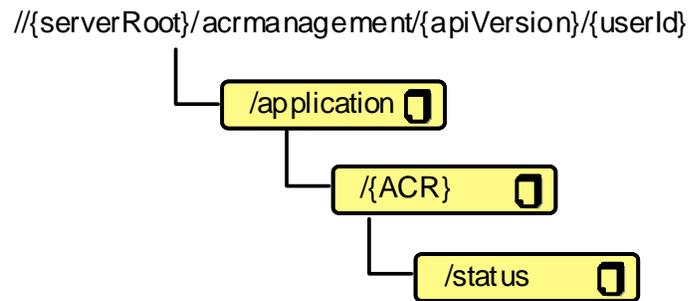


Figure 1: Resource structure defined by this specification

The following tables give a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods.

Purpose: Operations on Anonymous Customer Reference

Resource	URL Base URL: http://{serverRoot}/acr management/{apiVersion}/ {userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
End user's application	/application	Acrlist (used for GET response) Acr (used for POST)	Retrieves the issued ACR for the given end user identified by the {userId} in the context of a given application.	No	Creates an ACR for the user identified by the {userId}.	no
Anonymous customer reference	/application/{ACR}	Acr (used for GET response)	Retrieves the ACR data.	No	no	Remove an ACR.
ACR status	/application/{ACR}/status	Status	Retrieves the ACR status.	Refresh an "expired" ACR. (set the ACR status to "Valid")	no	no

5.2 Data Types

5.2.1 XML Namespaces

The XML namespace for the Anonymous Customer Reference Management data types is:

urn:oma:xml:rest:netapi:acrmangement:1

The 'xsd' namespace prefix is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace prefix is used in the present document to refer to the data types defined in [REST_NetAPI_Common]. The use of namespace prefixes such as 'xsd' is not semantically significant.

The XML schema for the data structures defined in the section below is given in [REST_SUP_ACR].

5.2.2 Structures

The subsections of this section define the data structures used in the Anonymous Customer Reference Management API.

Some of the structures can be instantiated as so-called root elements.

5.2.2.1 Type: AcrList

This type represents a list of Anonymous Customer References.

Element	Type	Optional	Description
acr	Acr [0..unbounded]	Yes	A list of ACRs.
resourceURL	xsd:anyURI	No	Self referring URL.

A root element named acrList of type AcrList is allowed in response bodies.

5.2.2.2 Type: Acr

This type represents an individual Anonymous Customer Reference and associated metadata.

Element	Type	Optional	Description
value	xsd:anyURI	Yes	<p>The string comprising the ACR, as per Appendix H.</p> <p>It MUST be included in responses.</p> <p>“DYNA” and “STAT” are the values of the type field in the ACR value as per Appendix H.</p> <p>A static ACR once created would never expire. It stays valid until it is revoked by either the user or the server.</p> <p>On the other hand a dynamic ACR is valid only for a certain period of time (as authorized by the user, requested by the application or determined by the server policy), after which it goes into an “expired” status. An expired ACR would need to be refreshed before it can be used again.</p> <p>An “expired” ACR after being dormant for a period of time MAY be removed by the server. Once a dormant-expired dynamic ACR is removed by the server, the application would need to request the creation of a new ACR (hence the name Dynamic as the ACR value may change if not used/refreshed in a timely manner).</p> <p>The value of expiry parameter as explained below determines whether the ACR is a dynamic or static ACR.</p>
acrStatus	AcrStatus	Yes	Status of the ACR. It MUST be included in responses, if the ACR exists.

<p>expiry</p>	<p>xsd:dateTimeStamp</p>	<p>Yes</p>	<p>The time at which the ACR will expire.</p> <p>This parameter MUST be included in responses if ACR is dynamic. However, this parameter may be excluded from responses for static (non-temporary) ACRs which means that the ACR shall not expire. If expiry parameter for static ACR is included in the responses, it shall have the value of "0001-01-01T00:00:00Z".</p> <p>For Dynamic ACRs expiry date/time may be requested or determined in the following ways:</p> <ul style="list-style-type: none"> • user authorizing a expiry date/time through OAuth access token which may be present in the request header (see [Autho4API_10]). • application indicating a expiry date/time in the API request via this ("expiry") parameter. • server's policy. <p>In API requests if OAuth access token specifies an expiry date/time for the ACR then it has precedence over the application's requested expiry date/time (i.e. "expiry" parameter in the request is ignored).</p> <p>Requested expiry date/time may be overridden by the server. As a result, the actual ACR expiry date/time is returned in the responses to the ACR creation requests.</p> <p>Expiry date/time value of "0001-01-01T00:00:00Z" in the requests (either through OAuth access token or the "expiry" parameter), would mean the request is for a Static ACR (i.e. the ACR should never expire).</p> <p>An expired ACR needs to be refreshed before it can be used again.</p> <p>How long an expired ACR would be kept around on the server's side before it is removed is determined by server's policy and out side of the scope of this document.</p>
<p>resourceURL</p>	<p>xsd:anyURI</p>	<p>Yes</p>	<p>Self referring URL.</p> <p>The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.</p>

A root element named acr of type Acr is allowed in request and/or response bodies.

5.2.2.3 Type: Status

This type represents the status of an Anonymous Customer Reference.

Element	Type	Optional	Description
acrStatus	AcrStatus	Yes	Status of the ACR. It MUST be included in responses, if the ACR exists.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named status of type Status is allowed in request and/or response bodies.

5.2.3 Enumerations

The subsections of this section define the enumerations used in the Anonymous Customer Reference Management API.

5.2.3.1 Enumeration: AcrStatus

This enumeration defines possible values to describe the status of the validity of an ACR.

Enumeration	Description
Valid	Indicates that the ACR is recognised by the server and may be used as the userId part within other requests managed by that server.
Revoked	The ACR has been revoked by the server, and may not be used as the userId part of other requests. A new ACR must be obtained for that MSISDN/User.
Expired	The ACR has expired, and a request can be made to refresh it. An expired ACR will have to be refreshed (see section 6.3.4) before it can be used as part of any other API request.

5.2.4 Values of the Link “rel” attribute

This specification does not define any elements of type Link.

5.3 Sequence Diagrams

The following subsections describe the resources, methods and steps involved in typical scenarios.

5.3.1 Operations on Anonymous Customer References

This figure below shows a scenario for retrieving, issuing and managing ACRs and ACR data.

The resources:

- To retrieve the issued ACR for the given end user identified by the {userId} in the context of a given application, read resource under
http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application
- To issue (create) an ACR for the user identified by the {userId}, create resource under
http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application
- To retrieve ACR data , read resource under

http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application/{ACR}

- To remove an ACR, delete resource under

http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application/{ACR}

- To retrieve the ACR status, read resource under

http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application/{ACR}/status

- To refresh an “expired” ACR (i.e. set the ACR status to “valid”), update resource under

http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application/{ACR}/status

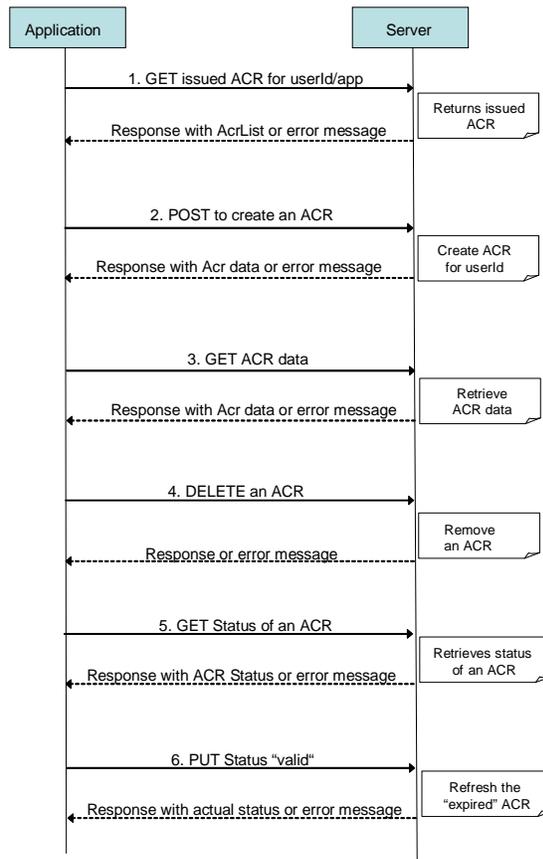


Figure 2: Management of ACR

1. An application requests the ACR for the given user using GET method and receives the already issued application-specific ACR.
2. An application requests creation of an application-specific ACR for the given user identified by the userId using POST method and receives the resulting ACR.
3. An application requests the ACR data (value, status, expiry) using GET method and receives the ACR data.
4. An application deletes (removes) an ACR with data by using DELETE method and receives response with the result of operation.
5. An application requests the status of an ACR using GET method and receives the actual status of the ACR.
6. An application refreshes an “expired” ACR (i.e. sets the ACR status to “valid”) by using PUT method and receives response with updated ACR status.

6. Detailed specification of the resources

The ACR Management API amongst other operations enables an application to create an ACR resource which in fact is an application-specific user identity of the user on whose behalf the application is acting. To create an ACR, the user's identity has to be known to the server through {userId} part of the resource URL. The following values are possible for the {userId}:

- 1) an identifier such as MSISDN
- 2) the "acr:auth" keyword.

When the {userId} is set to "acr:auth" keyword, it implies user's identity is known either through the OAuth access token (see [Autho4API_10]) available in the HTTP Authorization header of the API request or if the user's device is connected to the mobile network then MSISDN is known by the network and it is not necessary to be passed explicitly by the application in the API request.

The following applies to all resources defined in this specification regardless of the representation format (i.e. XML and JSON):

- Reserved characters in URL variables (parts of a URL denoted below by a name in curly brackets) **MUST** be percent-encoded according to [RFC3986]. Note that this always applies, no matter whether the URL is used as a Request URL or inside the representation of a resource (such as in "resourceURL" and "link" elements).
- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an MSISDN, it **MUST** be defined as a global number according to [RFC3966] (e.g. tel: +19585550100). The use of characters other than digits and the leading "+" sign **SHOULD** be avoided in order to ensure uniqueness of the resource URL. This applies regardless of whether the user identifier appears in a URL variable or in a parameter in the body of an HTTP message.
- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of a SIP URI, it **MUST** be defined according to [RFC3261].
- If a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an Anonymous Customer Reference (ACR), it **MUST** be defined according to Appendix H.
 - The ACR 'auth' is a supported reserved keyword, and **MUST NOT** be assigned as an ACR to any particular end user. See G.1.2 for details regarding the use of this reserved keyword.
- For requests and responses that have a body, the following applies: in the requests received, the server **SHALL** support JSON and XML encoding of the parameters in the body. The Server **SHALL** return either JSON or XML encoded parameters in the response body, according to the result of the content type negotiation as specified in [REST_NetAPI_Common]. In notifications to the Client, the server **SHALL** use either XML or JSON encoding, depending on which format the client has specified in the related subscription. The generation and handling of the JSON representations **SHALL** follow the rules for JSON encoding in HTTP Requests/Responses as specified in [REST_NetAPI_Common].

6.1 Resource: End User's Application

The resource used is:

http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application

This resource is used for retrieving an already issued application-specific ACR and for creating an application-specific ACR.

6.1.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	Version of the API client wants to use. The value of this variable is defined in section 5.1
userId	User identifier. Examples: tel:+19585550100, acr:auth

See section 6 for a statement on the escaping of reserved characters in URL variables.

6.1.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Anonymous Customer Reference Management, see section 7.

6.1.3 GET

This operation is used for retrieving an already issued application-specific ACR. List returned may contain more than one ACR. For instance the list may contain a revoked ACR and an active ACR. It is assumed that the server knows how to determine which ACRs belong to a given application and appropriately list them in the response, through methods which are beyond the scope of this document (e.g. using the access token provided in the HTTP request header – see [Autho4API_10]).

As shown in the example below, GET operation enables a 3rd-party application the retrieval of an application-specific ACR over the RESTful API. Please note that, to support certain use cases, it might also be possible for the service provider to push an already created and valid ACR to the 3rd-party application, e.g. via HTTP header enrichment techniques (i.e. by injecting additional HTTP header fields carrying an ACR into the user's traffic). HTTP header enrichment with ACR is beyond the scope of this document.

6.1.3.1 Example 1: Retrieve the issued ACR using “auth” keyword (Informative)

6.1.3.1.1 Request

```
GET /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Host: example.com
Accept: application/xml
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903
```

6.1.3.1.2 Response

```

HTTP/1.1 200 OK
Date: Thu, 04 Oct 2012 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cr:acrList xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <acr>
    <value>acr:abc123;ncc=23415;type=Stat</value>
    <acrStatus>Valid</acrStatus>

    <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth
/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat</resourceURL>
  </acr>
  <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application</resourceURL>
</cr:acrList>

```

6.1.3.2 Example 2: Retrieve the issued ACR using MSISDN (Informative)**6.1.3.2.1 Request**

```

GET /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application HTTP/1.1
Host: example.com
Accept: application/xml

```

6.1.3.2.2 Response

```

HTTP/1.1 200 OK
Date: Thu, 04 Oct 2012 02:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cr:acrList xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <acr>
    <value>acr:abc123;ncc=23415;type=Dyna</value>
    <acrStatus>Valid</acrStatus>
    <expiry>2013-10-26T21:32:52Z</expiry>

    <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application/acr%3Aabc123%3Bncc%3D2
3415%3Btype%3DDyna</resourceURL>
  </acr>
  <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application</resourceURL>
</cr:acrList>

```

6.1.3.3 Example 3: Retrieve a non-existent ACR using MSISDN (Informative)**6.1.3.3.1 Request**

```

GET /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application HTTP/1.1
Host: example.com
Accept: application/xml

```

6.1.3.3.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Oct 2012 02:51:59 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC1006</messageId>
    <text>ACR not found</text>
  </serviceException>
</common:requestError>
```

6.1.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET, POST’ field in the response as per section 14.7 of [RFC2616].

6.1.5 POST

This operation is used for creating an ACR. The “expiry” value which may be provided by the OAuth access token in the HTTP header (see [Autho4API_10]) or the body of the POST, would determine the type of the ACR (Dynamic or Static) requested.

Please note that, the inclusion of “expiry” value as part of the OAuth access token in order to ensure user’s authorization (as shown in Example 1 below) would require the usage of “subscoping” technique as defined in [Autho4API_10]). Please also note that, expiry value may also be present in the request body (as shown in example 1 below) as per aforementioned “subscoping” technique however, as stated in section 5.2.2.2 the value of expiry value in the OAuth access token takes precedence (if one is available).

6.1.5.1 Example 1: Create an ACR using “auth” keyword with expiry value included in OAuth access token (Informative)

This example illustrates the case of creating an ACR using “auth” keyword, where expiry value “0001-01-01T00:00:00Z” is included in OAuth access token.

6.1.5.1.1 Request

```
POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903
```

```
<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <expiry>0001-01-01T00:00:00Z</expiry>
</cr:acr>
```

6.1.5.1.2 Response

```

HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <value>acr:abc123;ncc=23415;type=Stat</value>
  <acrStatus>Valid</acrStatus>
  <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat</resourceURL>
</cr:acr>

```

6.1.5.2 Example 2: Create an ACR using MSISDN (Informative)

6.1.5.2.1 Request

```

POST /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <expiry>2013-10-26T21:32:52Z</expiry>
</cr:acr>

```

6.1.5.2.2 Response

```

HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <value>acr:abc123;ncc=23415;type=Dyna</value>
  <acrStatus>Valid</acrStatus>
  <expiry>2013-10-26T21:32:52Z</expiry>

  <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/tel%2B%3A4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna</resourceURL>
</cr:acr>

```

6.1.5.3 Example 3: Create an ACR over mobile network authenticated connection, response with a location of created resource (Informative)

6.1.5.3.1 Request

```
POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <expiry>2013-10-26T21:32:52Z</expiry>
</cr:acr>
```

6.1.5.3.2 Response

```
HTTP/1.1 201 Created
Location:
http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
<resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna</resourceURL>
</common:resourceReference>
```

6.1.5.4 Example 4: Create ACR over mobile network authenticated connection, response with creation failure due to an existing expired ACR (Informative)

6.1.5.4.1 Request

```
POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <expiry>2013-10-26T21:32:52Z</expiry>
</cr:acr>
```

6.1.5.4.2 Response

HTTP/1.1 403 Forbidden
 Content-Type: application/xml
 Content-Length: nnnn
 Date: Thu, 04 Oct 2012 02:51:59 GMT

```
<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <policyException>
    <messageId>POL1025</messageId>
    <text>An expired ACR, %1, already exists which needs to be refreshed prior to usage</text>
    <variables>abc123;ncc=23415;type=Dyna</variables>
  </policyException>
</common:requestError>
```

6.1.5.5 Example 5: Create an ACR over mobile network authenticated connection, response with a copy of created resource (Informative)

6.1.5.5.1 Request

POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
 Accept: application/xml
 Content-Length: nnnn
 Content-Type: application/xml
 Host: example.com

```
<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <expiry>2013-10-26T21:32:52Z</expiry>
</cr:acr>
```

6.1.5.5.2 Response

HTTP/1.1 201 Created
 Location:
<http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna>
 Content-Type: application/xml
 Content-Length: nnnn
 Date: Thu, 26 Oct 2012 21:32:52 GMT

```
<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <value>acr:abc123;ncc=23415;type=Dyna</value>
  <acrStatus>Valid</acrStatus>
  <expiry>2013-10-26T21:32:52Z</expiry>

  <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna</resourceURL>
</cr:acr>
```

6.1.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC2616].

6.2 Resource: Anonymous Customer Reference

The resource used is:

`http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application/{ACR}`

This resource is used for retrieving ACR data and for removing an ACR.

6.2.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	Version of the API client wants to use. The value of this variable is defined in section 5.1
userId	User identifier. Examples: tel:+19585550100, acr:auth
ACR	The string comprising the ACR, as per Appendix H

See section 6 for a statement on the escaping of reserved characters in URL variables.

6.2.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Anonymous Customer Reference Management, see section 7.

6.2.3 GET

This operation is used for retrieving ACR data.

6.2.3.1 Example 1: Retrieve ACR data

(Informative)

6.2.3.1.1 Request

```
GET /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna
HTTP/1.1
Accept: application/xml
Host: example.com
```

6.2.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

<?xml version="1.0" encoding="UTF-8"?>
<cr:acr xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <value>acr:abc123;ncc=23415:type=Dyna</value>
  <acrStatus>Valid</acrStatus>
  <expiry>2013-10-26T21:32:52Z</expiry>
  <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application/acr%3Aabc123%3Bncc%3D2
3415%3Btype%3DDyna</resourceURL>
</cr:acr>
```

6.2.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: [GET, DELETE]' field in the response as per section 14.7 of [RFC2616].

6.2.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: [GET, DELETE]’ field in the response as per section 14.7 of [RFC2616].

6.2.6 DELETE

This operation is used for removing an ACR.

6.2.6.1 Example 1: Remove an ACR (Informative)

6.2.6.1.1 Request

```
DELETE /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3Dstat HTTP/1.1
Host: example.com
Accept: application/xml
```

6.2.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 26 Oct 2012 21:32:52 GMT
```

6.3 Resource: ACR Status

The resource used is:

`http://{serverRoot}/acrmanagement/{apiVersion}/{userId}/application/{ACR}/status`

This resource is used for retrieving and refreshing the status of an ACR.

6.3.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	Version of the API client wants to use. The value of this variable is defined in section 5.1
userId	User identifier. Examples: tel:+19585550100, acr:auth
ACR	The string comprising the ACR, as per Appendix H

See section 6 for a statement on the escaping of reserved characters in URL variables.

6.3.2 Response Codes and Error Handling

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Anonymous Customer Reference Management, see section 7.

6.3.3 GET

This operation is used for retrieving the status of an ACR.

6.3.3.1 Example 1: Read the Status of an ACR using “auth” keyword (Informative)

6.3.3.1.1 Request

```
GET /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%Btype%3DStat/status HTTP/1.1
Accept: application/xml
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903
```

6.3.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

<?xml version="1.0" encoding="UTF-8"?>
<cr:status xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <acrStatus>Valid</acrStatus>
  <resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%Btype%3DDyna/status</resourceURL>
</cr:status>
```

6.3.3.2 Example 2: Read the Status of a non-existent ACR using “auth” keyword (Informative)

6.3.3.2.1 Request

```
GET /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%Btype%3DStat/status HTTP/1.1
Accept: application/xml
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903
```

6.3.3.2.2 Response

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Oct 2012 02:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <serviceException>
    <messageId>SVC1006</messageId>
    <text>ACR not found</text>
  </serviceException>
</common:requestError>
```

6.3.4 PUT

This operation is used for refreshing an expired ACR.

6.3.4.1 Example 1: Refresh an expired ACR by using “auth” keyword (Informative)

6.3.4.1.1 Request

```
PUT /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna/status HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903

<?xml version="1.0" encoding="UTF-8"?>
<cr:status xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <acrStatus>Valid</acrStatus>

<resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%
3DDyna/status</resourceURL>
</cr:status>
```

6.3.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

<?xml version="1.0" encoding="UTF-8"?>
<cr:status xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">

  <acrStatus>Valid</acrStatus>

<resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%
3DDyna/status</resourceURL>
</cr:status>
```

6.3.4.2 Example 2: Refreshing a revoked ACR is rejected (Informative)

6.3.4.2.1 Request

```
PUT /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna/status HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903

<?xml version="1.0" encoding="UTF-8"?>
<cr:status xmlns:cr="urn:oma:xml:rest:netapi:acrmanagement:1">
  <acrStatus>Valid</acrStatus>

<resourceURL>http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%
3DDyna/status</resourceURL>
</cr:status>
```

6.3.4.2.2 Response

```
HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Oct 2012 02:51:59 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <policyException>
    <messageId>POL1027</messageId>
    <text>ACR, %1, is revoked. A new ACR is required to be created</text>
    <variables>abc123;ncc=23415;type=Dyna</variables>
  </policyException>
</common:requestError>
```

6.3.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: [GET, PUT]' field in the response as per section 14.7 of [RFC2616].

6.3.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: [GET, PUT]' field in the response as per section 14.7 of [RFC2616].

7. Fault definitions

7.1 Service Exceptions

For common Service Exceptions refer to [REST_NetAPI_Common].

The following additional Service Exception codes are defined for the RESTful Anonymous Customer Reference Management API.

7.1.1 SVC1005: ACR creation failed. Unknown userId

Name	Description
MessageID	SVC1005
Text	ACR creation operation failed. Unknown userId
Variables	None
HTTP status code(s)	403 Forbidden

7.1.2 SVC1006: ACR not found

Name	Description
MessageID	SVC1006
Text	ACR not found
Variables	None
HTTP status code(s)	404 Not Found

7.2 Policy Exceptions

For common Policy Exceptions refer to [REST_NetAPI_Common].

The following additional Policy Exception codes are defined for the RESTful Anonymous Customer Reference Management API.

7.2.1 POL1024: ACR creation failed. An active ACR already exists

Name	Description
MessageID	POL1024
Text	An active ACR, %1, already exists
Variables	%1 – ACR value
HTTP status code(s)	403 Forbidden

7.2.2 POL1025: ACR creation failed. An expired ACR already exists

Name	Description
MessageID	POL1025
Text	An expired ACR, %1, already exists which needs to be refreshed prior to usage
Variables	%1 – ACR value
HTTP status code(s)	403 Forbidden

7.2.3 POL1026: Creation of Static ACR is not supported

Name	Description
MessageID	POL1026
Text	Creation of Static ACR is not supported
Variables	None
HTTP status code(s)	403 Forbidden

7.2.4 POL1027: Revoked ACR can no longer be used or refreshed

Name	Description
MessageID	POL1027
Text	ACR, %1, is revoked. A new ACR is required to be created.
Variables	%1 – ACR value
HTTP status code(s)	403 Forbidden

7.2.5 POL1028: Expired ACR may not be used unless refreshed

Name	Description
MessageID	POL1028
Text	ACR, %1, is expired. It is required to be refreshed before it is used.
Variables	%1 – ACR value
HTTP status code(s)	403 Forbidden

Appendix A. Change History

(Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No previous version within OMA

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions: REST_NetAPI_ACR-V1_0	13 Mar 2012	all	Initial baseline. Incorporates input to committee: OMA-ARC-2012-0060-INP_BaselineREST_NetAPI_ACR_TS
	17 Jul 2012	5, 7	Incorporated: OMA-ARC-REST-ACR-2012-0004-CR_TS_section5_section7
	24 Sep 2012	5.1, 5.2	Incorporated: OMA-ARC-REST-ACR-2012-0008R01- CR_resource_summary_section OMA-ARC-REST-ACR-2012-0009R02-CR_Data_types_section
	08 Oct 2012	5.3	Incorporated: OMA-ARC-REST-ACR-2012-0010R01-CR_Sequence_flows
	22 Oct 2012	many	Incorporated: OMA-ARC-REST-ACR-2012-0011R01-CR_Detailed_specification with additional comments included received in R&A.
	31 Oct 2012	6, App B, App G	Incorporated: OMA-ARC-REST-ACR-2012-0012R01-CR_SCRs_and_editorials
	13 Nov 2012	many	Incorporated: OMA-ARC-REST-ACR-2012-0017R02- CR_acr_auth_replacing_me_keyword
	31 Jan 2013	Many	Incorporated : OMA-ARC-REST-ACR-2013-0001- CR_fixing_TS_based_on_CONRR OMA-ARC-REST-ACR-2013-0004- CR_CONRR_fix_B051_B052_JSON_examples_in_TS
	09 Feb 2013	Many	Incorporated: OMA-ARC-REST-ACR-2013-0006- CR_fixing_TS_based_on_CONRR File format corrected from docx to doc
	19 Feb 2013	Many	Incorporated: OMA-ARC-REST-ACR-2013-0012- CR_Resolving_XML_validation_issues OMA-ARC-REST-ACR-2013-0011- CR_fixing_TS_based_on_remaining_CONRR_comments
	21 Feb 2013	Appendix D	Include JSON examples
	26 Feb 2013	D.4, D.5, D.6, D.9, D.10, D.11, D.13,	JSON example errors corrected.
Candidate version REST_NetAPI_ACR-V1_0	05 Mar 2013	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2013-0072- INP_REST_NetAPI_ACR_V1.0_ERP_and_ETR_for_Candidate_app roval
Draft Version: REST_NetAPI_ACR-V1_0	20 Mar 2013	6.1.5.1, 6.1.5.4.2, 6.3.4.2.2, D.1, D.4, D.7, D.14	Incorporated CR: OMA-ARC-RCS-NetAPI-2013-0030- CR_ACR_TS_fixing_requestError_responses Editorial changes
Candidate Version: REST_NetAPI_ACR-V1_0	25 Jun 2013	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2013-0209- INP_REST_NetAPI_ACR_V1_0_ERP_for_Notification

Document Identifier	Date	Sections	Description
Draft Versions: REST_NetAPI_ACR-V1_0	15 Apr 2014	2.1, 3.3, 4, 5, H	Incorporated CR: OMA-ARC-REST-NetAPI-2014-0036R01- CR_Add_Definition_of_the_acr_URI_scheme_to_ACR_TS Editorial changes
	08 May 2014	2, H	Incorporated CR: OMA-ARC-REST-NetAPI-2014-0038R02- CR_Additional_changes_to_ACR_TS
	01 Jul 2014	H.5	Incorporated CR: OMA-ARC-REST-NetAPI-2014-0052- CR_ACR_IANA_Registration_Template
	24 Sep 2014	2.2, 5.2.2.2, 5.3.1, 6, 6.2.1, 6.3.1, H, H.1, H.2, H.4, H.5	Incorporated CRs: OMA-ARC-REST-NetAPI-2014-0053R01- CR_ACR_IANA_Registration_addressing_review_comments OMA-ARC-REST-NetAPI-2014-0071R01- CR_Remaining_ACR_fixes_TS_ACR
	03 Oct 2014	H.5	Incorporated CR: OMA-ARC-REST-NetAPI-2014-0080- CR_ACR_IANA_Registration_clerical_change
	04 Nov 2014	H.1	Incorporated CR: OMA-ARC-REST-NetAPI-2014-0081R01- CR_ACR_IANA_Registration_bugfix
Candidate Version: REST_NetAPI_ACR-V1_0	03 Dec 2014	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2014-0270R01- INP_REST_NetAPI_ACR_V1_0_ERP_for_Notification
Draft Version: REST_NetAPI_ACR-V1_0	28 Feb 2015	2.1, 5.2.2.2, 6.1.3.2.2, 6.1.5.1, 6.1.5.1.1, 6.1.5.2, 6.1.5.3.1, 6.1.5.4.1, 6.1.5.5, 6.2.3.1.2, D, D.2, D.4-D.9, H	Incorporated CRs: OMA-ARC-REST-NetAPI-2014-0095R01- CR_ACR_change_JSON_reference OMA-ARC-REST-NetAPI-2014-0097- CR_ACR_change_dateTime_to_dateTimeStamp Editorial changes
Candidate Version: REST_NetAPI_ACR-V1_0	01 Dec 2015	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2015-0196- INP_REST_NetAPI_ACR_V1_0_ERP_for_Candidate_re_approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for REST.ACRMgt Server

Item	Function	Reference	Requirement
REST-ACRMGNT-SUPPORT-S-001-M	Support for the RESTful Anonymous Customer Reference Management API	5, 6	
REST-ACRMGNT-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-ACRMGNT-SUPPORT-S-003-M	Support for the JSON request & response format	6	

B.1.1 SCR for REST.ACRMgt.UsersApp Server

Item	Function	Reference	Requirement
REST-ACRMGNT-USERSAPP-S-001-M	Support for creation of an ACR and retrieval of an issued ACR	6.1	
REST-ACRMGNT-USERSAPP-S-002-M	Retrieve the issued ACR - GET	6.1.3	
REST-ACRMGNT-USERSAPP-S-003-M	Create an ACR – POST (XML or JSON)	6.1.5	

B.1.2 SCR for REST.ACRMgt.ACR Server

Item	Function	Reference	Requirement
REST-ACRMGNT-ACR-S-001-M	Support for removal of an ACR and retrieval of ACR data	6.2	
REST-ACRMGNT-ACR-S-002-M	Retrieve ACR data - GET	6.2.3	
REST-ACRMGNT-ACR-S-003-M	Remove an ACR – DELETE	6.2.5	

B.1.3 SCR for REST.ACRMgt.ACR.Status Server

Item	Function	Reference	Requirement
REST-ACRMGNT-ACRSTATUS-S-001-M	Support for refreshing expired ACRs and reading the status of an ACR	6.3	
REST-ACRMGNT-ACRSTATUS-S-002-M	Read the status of an ACR – GET	6.3.3	
REST-ACRMGNT-ACRSTATUS-S-003-M	Refresh an expired ACR – PUT	6.3.4	

Appendix C. Application/x-www-form-urlencoded Request Format for POST Operations (Normative)

This specification does not define any API request based on application/x-www-form-urlencoded MIME type.

Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a light-weight, text-based, language-independent data interchange format. It provides a simple means to represent basic name-value pairs, arrays and objects. JSON is relatively trivial to parse and evaluate using standard JavaScript libraries, and hence is suited for REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC7159].

The following examples show the request and response for various operations using the JSON data format. The examples follow the XML to JSON serialization rules in [REST_NetAPI_Common]. A JSON response can be obtained by using the content type negotiation mechanism specified in [REST_NetAPI_Common].

For full details on the operations themselves please refer to the section number indicated.

D.1 Retrieve the issued ACR using “auth” keyword (section 6.1.3.1)

Request:

```
GET /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Host: example.com
Accept: application/json
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Oct 2012 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"acrList": {
  "acr": {
    "acrStatus": "Valid",
    "resourceURL": "http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth
/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat",
    "value": "acr:abc123;ncc=23415;type=Stat"
  },
  "resourceURL": "http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application"
}}
```

D.2 Retrieve the issued ACR using MSISDN (section 6.1.3.2)

Request:

```
GET /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Oct 2012 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"acrList": {
  "acr": {
    "acrStatus": "Valid",
    "expiry": "2013-10-26T21:32:52Z",
    "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna",
    "value": "acr:abc123;ncc=23415;type=Dyna"
  },
  "resourceURL": "http://example.com/exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application"
}}
```

D.3 Retrieve a non-existent ACR using MSISDN (section 6.1.3.3)

Request:

```
GET /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application HTTP/1.1
Host: example.com
Accept: application/xml
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Oct 2012 02:51:59 GMT

{"requestError": {"serviceException": {
  "messageId": "SVC1006",
  "text": "ACR not found"
}}}
```

D.4 Create an ACR using “auth” keyword with expiry value included in OAuth access token (section 6.1.5.1)

Request:

```
POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Accept: application/json
Content-Length: nnnn
Content-Type: application/json
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903

{"acr": {"expiry": "0001-01-01T00:00:00Z"}}
```

Response:

```
HTTP/1.1 201 Created
Location: http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

{"acr": {
  "acrStatus": "Valid",
  "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat",
  "value": "acr:abc123;ncc=23415:type=Stat"
}}
```

D.5 Create an ACR using MSISDN (section 6.1.5.2)

Request:

```
POST /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application HTTP/1.1
Accept: application/json
Content-Length: nnnn
Content-Type: application/json
Host: example.com

{"acr": {"expiry": "2013-10-26T21:32:52Z"}}
```

Response:

```
HTTP/1.1 201 Created
Location:
http://example.com/exampleAPI/acrmanagement/v1/tel%2B%3A4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyn
a
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

{"acr": {
  "acrStatus": "Valid",
```

```
"expiry": "2013-10-26T21:32:52Z",
"resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/tel%2B%3A4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna
a",
"value": "acr:abc123;ncc=23415;type=Dyna"
}}
```

D.6 Create an ACR over mobile network authenticated connection, response with a location of created resource (section 6.1.5.3)

Request:

```
POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Accept: application/json
Content-Length: nnnn
Content-Type: application/json
Host: example.com

{"acr": {"expiry": "2013-10-26T21:32:52Z"}}
```

Response:

```
HTTP/1.1 201 Created
Location:
http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

{"resourceReference": {"resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna"}}
```

D.7 Create ACR over mobile network authenticated connection, response with creation failure due to an existing expired ACR (section 6.1.5.4)

Request:

```
POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com

{"acr": {"expiry": "2013-10-26T21:32:52Z"}}
```

Response:

```
HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Oct 2012 02:51:59 GMT

{"requestError": {"policyException": {
  "messageId": "POL1025",
  "text": "An expired ACR, %1, already exists which needs to be refreshed prior to usage",
  "variables": "abc123;ncc=23415:type=Dyna"
}}}
```

D.8 Create an ACR over mobile network authenticated connection, response with a copy of created resource (section 6.1.5.5)

Request:

```
POST /exampleAPI/acrmanagement/v1/acr%3Aauth/application HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com

{"acr": {"expiry": "2013-10-26T21:32:52Z"}}
```

Response:

```
HTTP/1.1 201 Created
Location:
http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

{"acr": {
  "acrStatus": "Valid",
  "expiry": "2013-10-26T21:32:52Z",
  "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna",
  "value": "acr:abc123;ncc=23415:type=Dyna"
}}
```

D.9 Retrieve ACR data (section 6.2.3.1)

Request:

```
GET /exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna
HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

{"acr": {
  "acrStatus": "Valid",
  "expiry": "2013-10-26T21:32:52Z",
  "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/tel%3A%2B4479901234567/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna",
  "value": "acr:abc123;ncc=23415:type=Dyna"
}}
```

D.10 Remove an ACR (section 6.2.6.1)

Request:

```
DELETE /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat HTTP/1.1
Host: example.com
Accept: application/json
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 26 Oct 2012 21:32:52 GMT
```

D.11 Read the Status of an ACR using “auth” keyword (section 6.3.3.1)

Request:

```
GET /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DStat/status HTTP/1.1
Accept: application/json
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

{"status": {
  "acrStatus": "Valid",
  "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna/status"
}}
```

D.12 Read the Status of a non-existent ACR using “auth” keyword (section 6.3.3.2)

Request:

```
GET /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%Btype%3DStat/status HTTP/1.1
Accept: application/xml
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Oct 2012 02:51:59 GMT

{"requestError": {"serviceException": {
  "messageId": "SVC1006",
  "text": "ACR not found"
}}}
```

D.13 Refresh an expired ACR by using “auth” keyword (section 6.3.4.1)

Request:

```
PUT /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%Btype%3DDyna/status HTTP/1.1
Accept: application/json
Content-Length: nnnn
Content-Type: application/json
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903

{"status": {
  "acrStatus": "Valid",
  "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%Btype%3DDyna/status"
}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Thu, 26 Oct 2012 21:32:52 GMT

{"status": {
  "acrStatus": "Valid",
  "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%Btype%3DDyna/status"
}}
```

D.14 Refreshing a revoked ACR is rejected (section 6.3.4.2)

Request:

```
PUT /exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna/status HTTP/1.1
Accept: application/xml
Content-Length: nnnn
Content-Type: application/xml
Host: example.com
Authorization: BEARER 08776724-6d0d-4aa6-a404-2bc19b5cf903

{"status": {
  "acrStatus": "Valid",
  "resourceURL":
"http://example.com/exampleAPI/acrmanagement/v1/acr%3Aauth/application/acr%3Aabc123%3Bncc%3D23415%3Btype%3DDyna/status"
}}
```

Response:

```
HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: nnnn
Date: Thu, 04 Oct 2012 02:51:59 GMT

{"requestError": {"policyException": {
  "messageId": "POL1027",
  "text": "ACR, %1, is revoked. A new ACR is required to be created",
  "variables": "abc123;ncc=23415;type=Dyna"
}}}
```

Appendix E. Operations mapping to a pre-existing baseline specification (Informative)

As this specification does not have a baseline specification, this appendix is empty.

Appendix F. Light-weight resources (Informative)

As this version of the specification does not define any light-weight resources, this appendix is empty.

Appendix G. Authorization aspects (Normative)

This appendix specifies how to use the RESTful Anonymous Customer Reference Management API in combination with some authorization frameworks.

G.1 Use with OMA Authorization Framework for Network APIs

The RESTful Anonymous Customer Reference Management API MAY support the authorization framework defined in [Autho4API_10].

A RESTful Anonymous Customer Reference Management API supporting [Autho4API_10]:

- SHALL conform to section D.1 of [REST_NetAPI_Common];
- SHALL conform to this section G.1.

G.1.1 Scope values

G.1.1.1 Definitions

In compliance with [Autho4API_10], an authorization server serving clients requests for getting authorized access to the resources exposed by the RESTful Anonymous Customer Reference Management API:

- SHALL support the scope values defined in the table below;
- MAY support scope values not defined in this specification.

Scope value	Description	For one-time access token
oma_rest_acrm.all_{apiVersion}	Provide access to all defined operations on the resources in this version of the API. The {apiVersion} part of this identifier SHALL have the same value as the “apiVersion” URL variable which is defined in section 5.1. This scope value is the union of the other scope values listed in next rows of this table.	No

Table 1: Scope values for RESTful Anonymous Customer Reference Management API

G.1.1.2 Downscoping

This version of the specification does not define any scope values to be used for downscoping purposes by the server and/or the resource owner.

G.1.1.3 Mapping with resources and methods

Tables in this section specify how the scope values defined in section G.1.1.1 for the RESTful Anonymous Customer Reference Management API map to the REST resources and methods of this API. In these tables, the root “oma_rest_acrm.” of scope values is omitted for readability reasons.

Resource	URL Base URL: http://{serverRoot}/acrmanagement/{apiVersion}/{userId}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
End User's Application	/application	6.1	all_{apiVersion}	n/a	all_{apiVersion}	n/a
Anonymous Customer Reference	/application/{ACR}	6.2	all_{apiVersion}	n/a	n/a	all_{apiVersion}
ACR Status	/application/{ACR}/status	6.3	all_{apiVersion}	all_{apiVersion}	n/a	n/a

Table 2: Required scope values for: Management of Anonymous Customer Reference

G.1.2 Use of 'acr:auth'

This section specifies the use of 'acr:auth' in place of an end user identifier in a resource URL path.

An 'acr' URI of the form 'acr:auth', where 'auth' is a reserved keyword MAY be used to avoid exposing a real end user identifier in the resource URL path.

A client MAY use 'acr:auth' in a resource URL in place of the {userId} resource URL variable in the resource URL path, when the RESTful Anonymous Customer Reference Management API is used in combination with [Autho4API_10].

In the case the RESTful Anonymous Customer Reference Management API supports [Autho4API_10], the server:

- SHALL accept 'acr:auth' as a valid value for the resource URL variable {userId}.
- SHALL conform to [REST_NetAPI_Common] section 5.8.1.1 regarding the processing of 'acr:auth'

Appendix H. Definition of the acr: URI scheme (Normative)

This appendix is built on top of [RFC3966] and it specifies the URI (Uniform Resource Identifier) scheme "acr". This URI scheme is intended to be used in places where URIs following the "tel" or "sip" scheme are normally used, but without disclosing the true identity of a reference or a user. The "acr" URI describes an anonymous reference that can be mapped to a resource or a user or a group of resources or users. There are multiple situations where the true identity of a user or a resource cannot be disclosed. The "acr" URI is an identifier ("name") only; it does not describe the steps necessary to reach the user or the device. However it can contain a parameter indication what body or organization could resolve it. It is intended for privacy protection, where a user trusts a translating party that can route or forward the request or message to the actual user or resource.

Informational note: This appendix is derived from the original proposal of the ACR concept [IETF_ACR_draft] in the IETF. As it is closely related to the OMA RESTful Network APIs, this work was not pursued to completion in the IETF, but was transferred to and finalized in the Open Mobile Alliance.

H.1 "acr" URI syntax

An "acr" URI conforms to the general URI syntax as defined in [RFC3986]. An "acr" URI is identified by the scheme "acr". The "authority", "query" and "fragment" components defined by [RFC3986] are empty. The scheme part "acr" is separated from the rest of the URI by a colon ":" character as defined by [RFC3986]. The ":" character is only significant as a separating character at the leftmost position. The "path" component according to [RFC3986] consists only of a single segment: an anonymous-customer-identifier and optional parameters. This means that an "acr" URI is not a hierarchical URI, i.e. URI parsers SHALL treat the "path" as an opaque string. The parameters are separated from the identifier and from each other by the reserved character ";", and a parameter name is separated from its value by the reserved character "=", as indicated to be common practice for segments by [RFC3986]. These two characters MUST be percent encoded if they appear in "acr" URI parameter values. The parts "query" and "fragment" according to [RFC3986] are not used in the "acr" URI scheme. Still, the two characters "?" and "#" which delimit these parts are reserved according to [RFC3986] and MUST therefore be percent encoded if they appear in "acr" URI parameter values.

The "acr" URI is defined using ABNF (Augmented Backus-Naur Form) as described in [RFC5234] and uses elements from the core definitions (appendix B of [RFC5234]).

Note: This specification uses the definition of the ABNF syntax in [RFC5234], whereas [RFC3986] references an earlier version [RFC2234]. The elements relevant for this specification are however identical in both versions.

The acr URI has the following syntax:

```
acr-uri           = "acr:" anonymous-customer-identifier *par
anonymous-customer-identifier = 1*paramchar
par              = parameter / network-code / acr-type / domainname
network-code    = ";"ncc=" 1*paramchar
acr-type        = ";type=" 1*( "DYNA" / "STAT" )
domainname      = ";domain=" *( domainlabel "." ) toplabel [ "." ]
domainlabel     = ALPHA / ALPHA *( alphanum / "-" ) alphanum
toplabel       = ALPHA / ALPHA *( alphanum / "-" ) alphanum
parameter       = ";" pname [ "=" pvalue ]
pname           = 1*( alphanum / "-" / "_" / "." )
pvalue         = 1*paramchar
paramchar       = param-unreserved / unreserved / pct-encoded
param-unreserved = "[" / "]" / "/" / ":" / "&" / "+" / "$" / "!" / "*" /
  "'" / "(" / ")" / "@" / ","
alphanum        = ALPHA / DIGIT
```

The following elements used in the rules above are imported from other specifications as listed below.

ALPHA	as defined by [RFC5234] and referenced by [RFC3986] ("A"- "Z", "a"- "z")
DIGIT	as defined by [RFC5234] and referenced by [RFC3986] ("0"- "9")
HEXDIG	as defined by [RFC5234] and referenced by [RFC3986] (hexadecimal digits)
pct-encoded	as defined by [RFC3986] ("% " HEXDIG HEXDIG)

unreserved as defined by [RFC3986] (ALPHA / DIGIT / "-" / "." / "_" / "~")

The anonymous-customer-identifier can be created from some suitable user or customer data such as phone number, and validation date. In order to provide anonymization, this data MUST not be included unchanged within the ACR. Rather it MUST be encrypted, hashed, represented by a look-up reference or otherwise obfuscated. The issuing provider is responsible for dereferencing the ACR to the user or resource. It may be necessary to convert the value of the anonymous-customer-identifier that results from the above-mentioned creation process into a format that is compatible with the allowed character set defined in the ABNF above, e.g. by using the “base64url” encoding [RFC4648].

If it is necessary to know who issued the anonymous-customer-identifier, the network code (ncc) or domain name MUST be included, for cross-operator identification and to ensure it is known which entity can deference the ACR. The "network code" (ncc) for public mobile land networks is a concatenation of "mobile country code" (MCC) and "mobile network code" (MNC) as defined in [ITU-T E.212]. An example ncc of “23415” consists of the concatenation of MCC=“234” and MNC=“15”. The current version of this specification makes no assumption about the structure of the “ncc” parameter for networks other than public mobile land networks.

The “acr-type” indicates if the ACR is a static type or a dynamic type. An ACR of static type uniquely maps a persistent customer identifier (such as an MSISDN number) to an anonymous-customer-identifier value without expiry time, whereas an ACR of dynamic type allows reuse of the anonymous-customer-identifier value after a time period defined by service provider policies.

Additional parameters can be defined as extensions.

The anonymous-customer-identifier value “auth” MUST NOT be assigned as an ACR to a particular user. It is reserved for use cases that determine the identity of the user by side information, such as an authorization token or particular network-based identification mechanisms, as specified in [REST_NetAPI_Common].

H.2 Examples

(Informative)

acr:0123456890123456789ABC

This URI points to a user. It is for network internal use only since the network code is not provided.

acr:0123456890123456789ABC;ncc=12345

This URI points to a user belonging to the network with MCC=“123” and MNC=“45”.

acr:0123456890123456789ABC;ncc=23415;type=DYNA

This temporal URI points to a user or group of users and can be resolved by the particular mobile network with MCC=“234” and MNC=“15”.

acr:0123456890123456789ABC;ncc=123654

This URI points to a group of users belonging to the network with MCC=“123” and MNC=“654”. Note that the fact that more than one user is represented is not intrinsic to the ACR but only known to the issuing network.

acr:0123456890123456789ABC;domain=example.com

This URI points to a user belonging in domain "example.com"

H.3 Rationale

(Informative)

H.3.1 Privacy policies

Existing privacy policies and legislation restrict the sharing of certain user identifiers, such as the MSISDN, since it may be used to breach a user's privacy (unauthorized location look-up, cold calling, SMS Spam etc.). An ACR prevents such identifiers from being circulated.

H.3.2 Cookie support

Cookie support is inconsistent across mobile devices. An ACR can help identify a returning mobile user to a Website, and hence facilitate the provisioning of a personalized service based on previous preferences and activity.

H.3.3 Sharing identity

Mobile, broadband and other access networks do not typically share a user identifier. The ACR is not bound to a particular access network and can hence be used to provision user identifiers between networks.

H.3.4 Relation to SIP

The ACR can help the implementation of SIP privacy considerations, as detailed in [RFC3323], 'A Privacy Mechanism for the Session Initiation Protocol'. Specifically the ACR can be used as the value for the 'anonymous from' header field (see [RFC3323] section 4.1), and is consistent with the recommendation to remove Subject, Call-info, Organization, User Agent, Reply-To, In-Reply-To in [RFC3323] section 4.3.

H.4 Security Considerations

Since the ACR is used to protect the identity of a user or a device the forwarding party must not disclose information that would or can be used to reveal the identity of the user. However the network code or domain name will reveal some information of the user's affiliation or service provider.

The security considerations parallel those for the "tel" URI [RFC3966].

Web clients and similar tools MUST NOT use the "acr" URI to place telephone calls or send messages without the explicit consent of the user of that client. Placing calls or sending messages automatically without appropriate user confirmation may incur a number of risks, such as those described below:

- Calls or messages may incur costs.
- The URI may be used to place malicious or annoying calls.
- A call will take the user's phone line off-hook, thus preventing its use.
- A call may reveal the user's possibly unlisted phone number to the remote host in the caller identification data and may allow the attacker to correlate the user's phone number with other information, such as an e-mail or IP address.

This is particularly important for "acr" URIs embedded in HTML links, as a malicious party may hide the true nature of the URI in the link text, as in

```
<a href="acr:123456">Find free information here</a>
```

```
<a href="acr:123456">Call Foobar organization for help</a>
```

"acr" URIs may still reveal private information if the mapping from a user identity to an anonymous-customer-identifier is designed poorly. It is advised to ensure that the anonymous-customer-identifier is sufficiently random, e.g. by applying a hashing step.

It needs to be noted that hashing is prone to accelerated brute force attacks e.g. using rainbow tables; in particular if the space of clear text is small, as this may be the case if phone numbers are hashed directly. Countermeasures such as salting need to be considered. In addition, the use of a hash function that is strong enough (i.e. produces a sufficient number of bits) is necessary to prevent such attacks from being successful in practice.

H.5 IANA registration template

URI scheme name.

acr

Status.

provisional

URI scheme syntax.

See Appendix H.1 in [1].

URI scheme semantics.

The "acr" URI describes an anonymous reference that can be mapped to a resource or a user or a group of resources or users, and that may include information which hints to an authority that can perform such mapping.

Encoding considerations.

The syntax definition above contains a complete enumeration of allowed characters, and contains provisions for percent-encoding of unsafe characters.

Applications/protocols that use this URI scheme name.

This URI scheme has been defined in [1] for use with a number of OMA RESTful Network APIs (of which [2] is an example).

Interoperability considerations.

This URI scheme has been defined for use with the OMA RESTful Network APIs. Multibyte character sets are not supported.

Security considerations.

See Appendix H.4 in [1].

Contact.

Uwe [dot] Rauschenbach [at] nsn [dot] com

Author/Change controller.

OMNA - Open Mobile Naming Authority of the Open Mobile Alliance
OMA-OMNA [at] mail [dot] openmobilealliance [dot] org

References.

[1] OMA RESTful Network API for Anonymous Customer Reference Management, OMA-TS-REST_NetAPI_ACR-V1_0, The Open Mobile Alliance. Published under <http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/rest-netapi-acr-v1-0>.

[2] OMA RESTful Network API for Chat, OMA-TS-REST_NetAPI_Chat-V1_0, The Open Mobile Alliance. Published under <http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/restfulnetworkapi-chat-v1-0>