



RESTful Network API for Converged Address Book (CAB)

Approved Version 1.0 – 02 Jul 2013

Open Mobile Alliance
OMA-TS-REST_NetAPI_CAB-V1_0-20130702-A

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.

© 2013 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	7
2.	REFERENCES	8
2.1	NORMATIVE REFERENCES	8
2.2	INFORMATIVE REFERENCES	9
3.	TERMINOLOGY AND CONVENTIONS	10
3.1	CONVENTIONS	10
3.2	DEFINITIONS	10
3.3	ABBREVIATIONS	10
4.	INTRODUCTION	12
4.1	VERSION 1.0	12
5.	CAB API DEFINITION	13
5.1	RESOURCES SUMMARY	14
5.2	CAB DATA TYPES	22
5.2.1	XML Namespaces	22
5.2.2	Structures	22
5.2.2.1	<i>Type: ContactCollection</i>	22
5.2.2.2	<i>Type: Contact</i>	22
5.2.2.3	<i>Type: SharedIdentity</i>	22
5.2.2.4	<i>Type: AttributeList</i>	22
5.2.2.5	<i>Type: Attribute</i>	22
5.2.2.6	<i>Type: ListCollection</i>	24
5.2.2.7	<i>Type: List</i>	24
5.2.2.8	<i>Type: MemberCollection</i>	24
5.2.2.9	<i>Type: Member</i>	24
5.2.2.10	<i>Type: ListReferenceCollection</i>	25
5.2.2.11	<i>Type: AbChangesSubscriptionCollection</i>	25
5.2.2.12	<i>Type: AbChangesSubscription</i>	26
5.2.2.13	<i>Type: AbChangeNotification</i>	26
5.2.2.14	<i>Type: ProfileCollection</i>	27
5.2.2.15	<i>Type: Profile</i>	27
5.2.2.16	<i>Type: ApplicableToProfile</i>	28
5.2.2.17	<i>Type: ApplicableToProfileNode</i>	29
5.2.2.18	<i>Type: ApplicableToContacts</i>	29
5.2.2.19	<i>Type: ExternalRequestCollection</i>	29
5.2.2.20	<i>Type: ExternalRequest</i>	29
5.2.3	Enumerations	30
5.2.3.1	<i>Enumeration: ListType</i>	30
5.2.3.2	<i>Enumeration: ResourceStatus</i>	31
5.2.4	Values of the Link “rel” attribute	31
5.3	SEQUENCE DIAGRAMS	31
5.3.1	Managing contacts in a flat list	31
5.3.2	Accessing the lists and members	31
5.3.3	Subscribing to address book changes and receiving notifications	31
5.3.4	Managing authorization rules	31
6.	DETAILED SPECIFICATION OF THE RESOURCES	32
6.1	RESOURCE: COLLECTION OF CONTACTS	32
6.2	RESOURCE: INDIVIDUAL CONTACT	32
6.3	RESOURCE: ATTRIBUTES FOR A CONTACT	32
6.4	RESOURCE: INDIVIDUAL ATTRIBUTE FOR A CONTACT	33
6.5	RESOURCE: LISTS	33
6.6	RESOURCE: INDIVIDUAL LIST	33
6.7	RESOURCE: MEMBERS IN A LIST	33
6.8	RESOURCE: INDIVIDUAL MEMBER IN A LIST	33
6.9	RESOURCE: ATTRIBUTES FOR A MEMBER IN A LIST	33
6.10	RESOURCE: INDIVIDUAL ATTRIBUTE FOR A MEMBER IN A LIST	33

6.11	RESOURCE: LIST REFERENCES	34
6.12	RESOURCE: INDIVIDUAL LIST REFERENCE	34
6.13	RESOURCE: ADDRESS BOOK CHANGES SUBSCRIPTIONS	34
6.14	RESOURCE: INDIVIDUAL LIST CHANGES SUBSCRIPTION	34
6.15	RESOURCE: CLIENT RESOURCE FOR LIST CHANGES NOTIFICATIONS	34
6.16	RESOURCE: COLLECTION OF PROFILES	35
6.16.1	Request URL variables	35
6.16.2	Response Codes and Exceptions.....	35
6.16.3	GET.....	35
6.16.3.1	<i>Example 1: Retrieve all profiles (Informative)</i>	35
6.16.4	PUT.....	38
6.16.5	POST.....	38
6.16.6	DELETE	38
6.17	RESOURCE: INDIVIDUAL PROFILE	38
6.17.1	Request URL variables	38
6.17.2	Response Codes and Exceptions.....	39
6.17.3	GET.....	39
6.17.3.1	<i>Request</i>	39
6.17.3.2	<i>Response</i>	39
6.17.4	PUT.....	42
6.17.4.1	<i>Example: Update an existing profile as an XML object (Informative)</i>	42
6.17.4.2	<i>Example: Update attributes of an existing profile using "Profile" (Informative)</i>	43
6.17.5	POST.....	46
6.17.6	DELETE	46
6.17.6.1	<i>Example: Delete a profile (Informative)</i>	46
6.18	RESOURCE: ATTRIBUTES OF A PROFILE	47
6.18.1	Request URL variables	47
6.18.2	Response Codes and Exceptions.....	47
6.18.3	GET.....	47
6.18.3.1	<i>Example: Retrieve all attributes of a profile (Informative)</i>	47
6.18.4	PUT.....	49
6.18.5	POST.....	49
6.18.6	DELETE	49
6.19	RESOURCE: INDIVIDUAL ATTRIBUTE FOR A PROFILE	49
6.19.1	Request URL variables	49
6.19.2	Response Codes and Exceptions.....	50
6.19.3	GET.....	50
6.19.3.1	<i>Example : Retrieve a profile's attribute (Informative)</i>	50
6.19.4	PUT.....	51
6.19.4.1	<i>Example : Update an existing profile with new attribute (Informative)</i>	51
6.19.5	POST.....	52
6.19.6	DELETE	52
6.19.6.1	<i>Request</i>	52
6.19.6.2	<i>Response</i>	52
6.20	RESOURCE: AUTHORIZATION RULES	52
6.21	RESOURCE: INDIVIDUAL AUTHORIZATION RULE	52
6.22	RESOURCE: INDIVIDUAL AUTHORIZATION RULE DATA	53
6.23	RESOURCE: COLLECTION OF EXTERNAL REQUESTS	53
6.23.1	Request URL variables	53
6.23.2	Response Codes and Exceptions.....	53
6.23.3	GET.....	53
6.23.3.1	<i>Example 1: Retrieve all External Requests (Informative)</i>	53
6.23.4	PUT.....	54
6.23.5	POST.....	54
6.23.5.1	<i>Example 1: Create a new External Request (Informative)</i>	54
6.23.6	DELETE	55
6.24	RESOURCE: INDIVIDUAL EXTERNAL REQUEST	55
6.24.1	Request URL variables	55
6.24.2	Response Codes and Exceptions.....	56

- 6.24.3 GET.....56
 - 6.24.3.1 Example 1: Retrieve an individual External Request (Informative).....56
- 6.24.4 PUT.....56
 - 6.24.4.1 Example 1: Update an individual External Request (Informative).....56
- 6.24.5 POST.....57
- 6.24.6 DELETE57
 - 6.24.6.1 Example 1: Delete an individual External Request (Informative)57
- 7. FAULT DEFINITIONS59**
 - 7.1 SERVICE EXCEPTIONS.....59**
- APPENDIX A. CHANGE HISTORY (INFORMATIVE).....60**
 - A.1 APPROVED VERSION HISTORY60**
- APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)61**
 - B.1 SCR FOR REST. CAB SERVER61**
 - B.1.1 SCR for REST.CAB.ContactCollection Server61
 - B.1.2 SCR for REST.CAB.IndividualContact Server61
 - B.1.3 SCR for REST.CAB.AttributesForAContact Server61
 - B.1.4 SCR for REST.CAB.IndividualAttributeForAContact Server.....62
 - B.1.5 SCR for REST.CAB.Lists Server62
 - B.1.6 SCR for REST.CAB.IndividualList Server62
 - B.1.7 SCR for REST.CAB.MemberInAList Server62
 - B.1.8 SCR for REST.CAB.IndividualMemberInAList Server63
 - B.1.9 SCR for REST.CAB.AttributesForAMemberInAList Server63
 - B.1.10 SCR for REST.CAB.IndividualAttributeForAMemberInAList Server63
 - B.1.11 SCR for REST.CAB.ListReferences Server63
 - B.1.12 SCR for REST.CAB.IndivListReference Server64
 - B.1.13 SCR for REST.CAB.List.Subscr Server64
 - B.1.14 SCR for REST.CAB.Individual.Subscr Server64
 - B.1.15 SCR for REST.CAB.Notif Server.....65
 - B.1.16 SCR for REST.CAB.Authorization.RuleList Server65
 - B.1.17 SCR for REST.CAB.Indiv.AuthorizationRule Server65
 - B.1.18 SCR for REST.CAB.Indiv.AuthorizationRuleData Server.....66
 - B.1.19 SCR for REST.CAB.ProfileCollection Server.....66
 - B.1.20 SCR for REST.CAB.IndividualProfile Server.....66
 - B.1.21 SCR for REST.CAB.AttributesForAProfile Server66
 - B.1.22 SCR for REST.CAB.IndividualAttributeForAProfile Server67
 - B.1.23 SCR for REST.CAB.ExternalRequestCollection Server67
 - B.1.24 SCR for REST.CAB.IndividualExternalRequest Server67
- APPENDIX C. APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR SELECTED REST OPERATIONS (NORMATIVE).....69**
 - C.1 CREATE A NEW EXTERNAL REQUEST69**
 - C.1.1 Example 1: Create a new External Request (Informative).....70
 - C.1.1.1 Request.....70
 - C.1.1.2 Response70
- APPENDIX D. JSON EXAMPLES (INFORMATIVE)72**
- APPENDIX E. OPERATIONS MAPPING TO A PRE-EXISTING BASELINE SPECIFICATION (INFORMATIVE).....73**
- APPENDIX F. LIGHT-WEIGHT RESOURCES FOR CAB APIS (INFORMATIVE)74**
- APPENDIX G. AUTHORIZATION ASPECTS (NORMATIVE)75**
 - G.1 USE WITH OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIS.....75**
 - G.1.1 Scope values75
 - G.1.1.1 Definitions.....75
 - G.1.1.2 Downscoping75
 - G.1.1.3 Mapping with resources and methods.....75
 - G.1.2 Use of ‘acr:auth’80

APPENDIX H. EXPOSURE BY 3RD PARTY SYSTEMS81

H.1 IMPORT CAB USER’S PERSONAL PROFILE INFORMATION FROM 3RD PARTY SYSTEMS 81

H.2 EXPORT CAB USER’S PCC INFORMATION TO 3RD PARTY SYSTEMS..... 81

H.3 SUBSCRIPTION TO CAB USER’S PERSONAL PROFILE INFORMATION IN 3RD PARTY SYSTEMS..... 81

Figures

Figure 1 Resource structure defined by this specification..... 14

Tables

Table 1 Address Book management 16

Table 2 PCC management 18

Table 3 Subscriptions to changes..... 18

Table 4 Contact Subscriptions management 19

Table 5 External Requests management..... 20

Table 6 Access Permissions management 21

Table 7 External Request 70

Table 8 Light-weight Resources for CAB APIs..... 74

Table 9 Scope values for RESTful Converged Address Book API..... 75

Table 10 Required scope values for: managing contacts..... 76

Table 11 Required scope values for: managing profile 77

Table 12 Required scope values for: managing CAB subscription-list and entries in the list 78

Table 13 Required scope values for: managing subscriptions to notifications about contacts, profile and CAB subscription-list changes 78

Table 14 Required scope values for: managing external requests..... 78

Table 15 Required scope values for: managing authorization rules for shared contacts and profile 79

1. Scope

This specification defines a RESTful API for CAB using HTTP protocol bindings.

2. References

2.1 Normative References

- [Addressbook TS 1.0] RESTful Network API for Address Book “, Version 1.0, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Addressbook-V1_0, URL:<http://www.openmobilealliance.org/>
- [Autho4API_10] “Authorization Framework for Network APIs”, Open Mobile Alliance™, OMA-ER-Autho4API-V1_0, URL:<http://www.openmobilealliance.org/>
- [CAB AD] “Converged Address Book Architecture”, Version 1.0, Open Mobile Alliance™, OMA-AD-CAB-V1_0, URL:<http://www.openmobilealliance.org/>
- [CAB_APIs RD] “Converged Address Book Application Program Interfaces Requirements”, Version 1.0, Open Mobile Alliance™, OMA-RD-CAB_APIs-V1_0, URL:<http://www.openmobilealliance.org/>
- [CAB RD] “Converged Address Book Requirements”, Version 1.0, Open Mobile Alliance™, OMA-RD-CAB-V1_0, URL:<http://www.openmobilealliance.org/>
- [CAB TS] “Converged Address Book Specification”, Version 1.0, Open Mobile Alliance™, OMA-TS-CAB-V1_0, URL:<http://www.openmobilealliance.org/>
- [CAB XDMS] “Converged Address Book XDM Specification”, Version 1.0, Open Mobile Alliance™, OMA-TS-CAB-XDMS-V1_0, URL:<http://www.openmobilealliance.org/>
- [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-00>
- [REST_NetAPI_Common] “Common definitions and specifications for RESTful Network API”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL:<http://www.openmobilealliance.org/>
- [REST_NetAPI_NotificationChannel] “RESTful Network API for Notification Channel”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-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>
- [RFC3966] “The tel URI for Telephone Numbers”, H.Schulzrinne, December 2004, URL:<http://www.ietf.org/rfc/rfc3966.txt>
- [RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et. al, January 1999, URL:<http://www.ietf.org/rfc/rfc2616.txt>
- [RFC3986] “Uniform Resource Identifier (URI): Generic Syntax”, R. Fielding et. al, January 2005, URL:<http://www.ietf.org/rfc/rfc3986.txt>
- [RFC4234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. October 2005, URL:<http://www.ietf.org/rfc/rfc4234.txt>
- [RFC4627] “The application/json Media Type for JavaScript Object Notation (JSON)”, D. Crockford, July 2006, URL:<http://www.ietf.org/rfc/rfc4627.txt>
- [RFC4661] “An Extensible Markup Language (XML)-Based Format for Event Notification Filtering” H. Khartabil, September 2006, URL:<http://www.ietf.org/rfc/rfc4661.txt>
- [OMA XDM Core] “XML Document Management Specification”, Version 2.1, Open Mobile Alliance™, OMA-TS-XDM_Core-V2_1, URL:<http://www.openmobilealliance.org/>
- [SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:<http://www.openmobilealliance.org/>
- [W3C_URLENC] HTML 4.01 Specification, Section 17.13.4 Form content types, The World Wide Web Consortium, URL:<http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1>

- [XMLSchema1] W3C Recommendation, XML Schema Part 1: Structures Second Edition,
URL:<http://www.w3.org/TR/xmlschema-1/>
- [XMLSchema2] W3C Recommendation, XML Schema Part 2: Datatypes Second Edition,
URL:<http://www.w3.org/TR/xmlschema-2/>

2.2 Informative References

- [OMADICT] “Dictionary for OMA Specifications”, Version x.y, 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/>

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 [OMA-DICT].

Address Book	Refer to [CAB_APIs RD]
CAB User	Refer to [CAB_APIs RD]
Contact Entry	Refer to [CAB_APIs RD]
Contact Share	Refer to [CAB_APIs RD]
Contact Status	Refer to [CAB_APIs RD]
Contact Subscription	Refer to [CAB_APIs RD]
Contact View	Refer to [CAB_APIs RD]
External Requests	A request by which the CAB Enabler is instructed to interact with an External Entity on user’s behalf (e.g. import user’s profile information from 3 rd Party System)
Group-Details	Structure within the PCC that represents the information pertinent to a group inside a PCC document (i.e. <group-details> as defined in [CAB XDMS] sect. 5.2.1 “PCC Application Usage”).
Heavy-weight Resource	A resource which is identified by a resource URL which is then used by HTTP methods to operate on the entire data structure representing the resource.
JavaScriptObjectNotation	Refer to [CAB_APIs RD]
Legacy Formats	Refer to [CAB_APIs RD]
Light-weight Resource	A subordinate resource of a Heavy-weight Resource which is identified by its own resource URL which is then used by HTTP methods to operate on a part of the data structure representing the Heavy-weight Resource. The Light-weight Resource URL can be seen as an extension of the Heavy-weight Resource URL. There could be several levels of Light-weight Resources below the ancestor Heavy-weight Resource, depending on the data structure.
Org-Details	Structure within the PCC that represents the information pertinent to an organization inside a PCC document (i.e. <org-details> as defined in [CAB XDMS] sect. 5.2.1 “PCC Application Usage”).
Personal Contact Card	Refer to [CAB_APIs RD]
Person-Details	Structure within the PCC that represents the information pertinent to an individual inside a PCC document (i.e. <person-details> as defined in [CAB XDMS] sect. 5.2.1 “PCC Application Usage”).
Published Contact Card	Refer to [CAB_APIs RD]

3.3 Abbreviations

AB	Address Book
ACR	Anonymous Customer Reference
CAB	Converged Address Book
API	Application Programming Interface

HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
OMA	Open Mobile Alliance
MIME	Multipurpose Internet Mail Extensions
PCC	Personal Contact Card
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 for the OMA RESTful CAB API contains HTTP protocol bindings for Converged Address Book (CAB) Enabler, 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, and application/x-www-form-urlencoded).

The JSON format used in the CAB APIs 1.0 specification should follow requirements in [REST_common].

4.1 Version 1.0

Version 1.0 of this specification supports operations exposing the CAB PCC and AB data.

The following operations have been reused from the [Addressbook TS 1.0] as follows:

- reused by CAB APIs entirely (“as is”):
 - management of contacts in the CAB AB and their attributes
 - manage subscriptions to CAB AB changes
 - send notifications about AB changes
- enhanced with additional extensions (e.g. resources, data types or values) for CAB APIs v1.0
 - manage ContactSubscription lists
 - manage subscriptions to contacts’ CAB PCC changes
 - send notifications about contact’s PCC changes
 - authorization (management of CAB PCC and AB AccessPermissions)

The following operations have been introduced by CAB APIs specifications:

- management of user profile (CAB PCC)
- support for external requests (e.g. import and export of contacts and profiles from external parties, subscriptions to external profiles changes and notifications)

The following new functionality has been introduced:

- 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 an ACR

5. CAB API definition

This section is organized to support a comprehensive understanding of the CAB 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 [OMA_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 the detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP methods, the possible HTTP response codes, 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. Application/x-www-form-urlencoded examples are provided in Appendix C, while 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 describes the CAB APIs resources that a 3rd Party system can expose to facilitate the interaction with the CAB Server in order to fulfil Import, Export and Subscription External Requests.

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

Conventions in this specification

The RESTful CAB APIs specification inherits and extends the Addressbook API as specified in [Addressbook TS 1.0]. Therefore the operations and data types reused from [Addressbook TS 1.0] are marked as such throughout the CAB APIs TS.

The data types tables in section 5.2 contain two additional columns compared to the original TS template, in order to clearly distinguish between reuse and extensions. These additional columns contain details of the reuse, extended data and/or a specific usage indication within the CAB APIs, as follows:

- "Defined in" column: contains the specification that defines the data type and it can be any one of: [Addressbook TS 1.0], or CAB APIs TS 1.0.*
- "Reuse or extended use description": provides details regarding the assigned usage within the CAB APIs specification (e.g. "not used in CAB APIs", "used in CAB APIs for subscription-list, etc).*

5.1 Resources Summary

This section summarizes all the resources used by the CAB API.

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.

The figure below visualizes the resource structure defined by this specification. Note that those nodes in the resource tree which have associated HTTP methods defined in this specification are depicted by solid boxes.

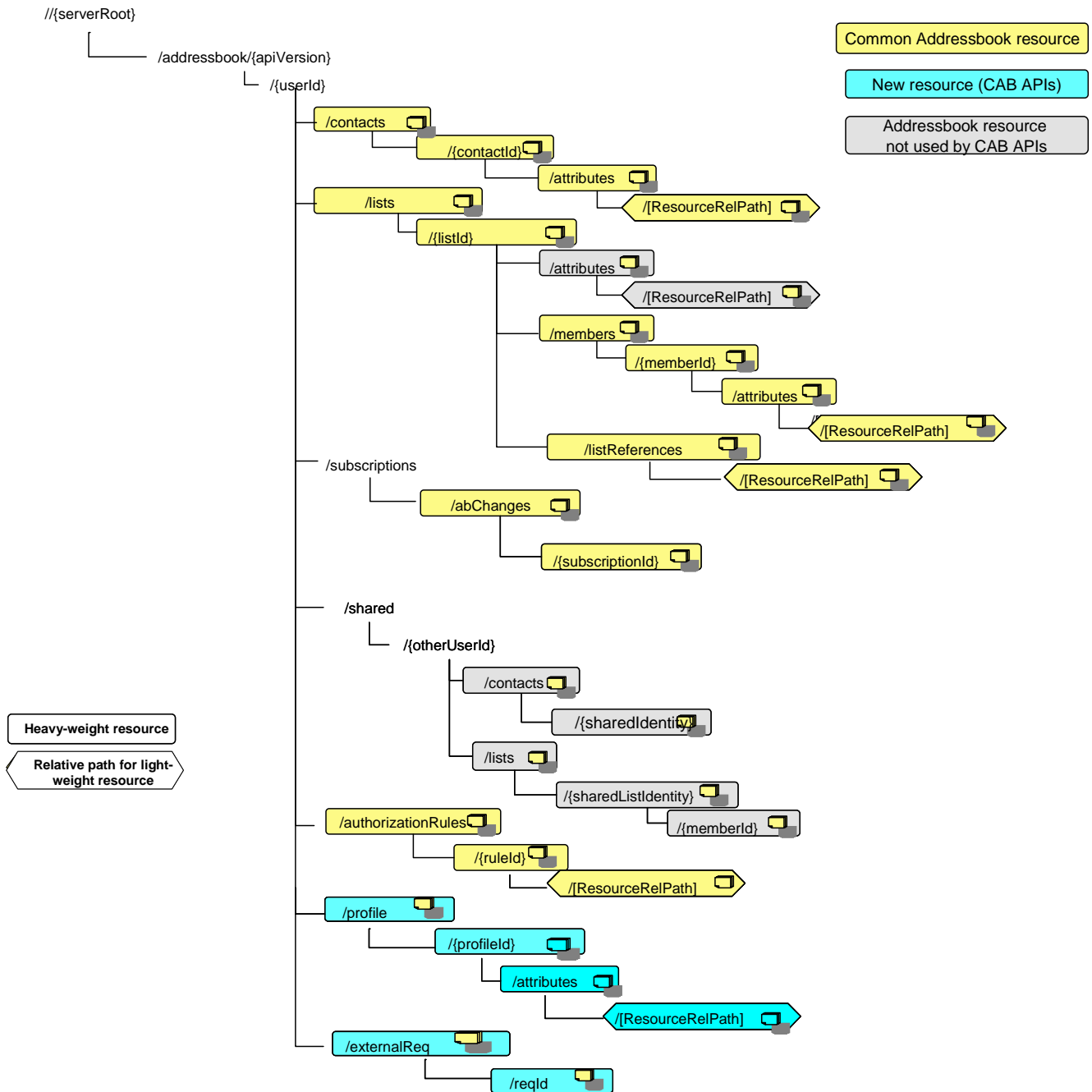


Figure 1 Resource structure defined by this specification

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: Address Book operations: to allow client to manage all contacts

Resource	URL Base URL: http://{serverRoot}/ addressbook/{apiV ersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of contacts	/{userId}/contacts	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Individual contact	/{userId}/contacts/{contactId}	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Attributes for a contact	/{userId}/contacts/{contactId}/attributes	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Individual attribute for a contact	/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].

Table 1 Address Book management

Purpose: Management of PCC

Resource	URL Base URL: http://{serverRoot}/ addressbook/{apiV ersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE

Resource	URL Base URL: http://{serverRoot}/ addressbook/{apiV ersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
User Profile	/{userId}/profile	ProfileCollection	This operation retrieves information about users' profile data (e.g. CAB PCC)	No	No	No
Individual Profile	/{userId}/profile/ /{profileId}	Profile	This operation retrieves information about profile data of an user,	This operation creates a new profile, or a new structure of a profile (e.g. Person-Details, Org-Details or Group-Details), or, updates information about an existing profile structure of a user.	No	This operation removes a profile, or a specific part of the profile
Attributes for a Profile	/{userId}/profile/{profileId}/attributes	AttributeList	This operation returns all attributes for a Profile.	no	no	no
Individual attribute in a Profile	/{userId}/profile/{profileId}/attributes/{ResourceRelPath}	Attribute	This operation retrieves Profile attribute data as specified in the URL.	This operation creates or updates attribute data pointed by the URL.	no	This operation removes the attribute data pointed by the URL.

Table 2 PCC management

Purpose: To allow client to manage any one of the following subscriptions to changes: addressbook, profile and ContactSubscription list

Resource	Base URL: http://{serverRoot}/ addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Address Book changes subscriptions	{userId}/subscriptions/abChanges	AbChangesSubscription Collection (Used for GET) AbChangesSubscription (Used for POST)	As defined in [Addressbook TS 1.0].	no	This operation creates a new subscription for CAB resource changes.	no
Individual CAB resource changes subscription (i.e. AB, Profile, CABSubscriptionList)	{userId}/subscriptions/abChanges/{subscriptionId}	AbChangesSubscription	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].

Table 3 Subscriptions to changes

Purpose: To allow client to manage its CAB subscription-list for ContactSubscriptions

Resource	Base URL: http://{serverRoot}/ addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Lists	{userId}/lists	ListCollection	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Individual list	{userId}/lists/{listId}	List	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Members in a list	{userId}/lists/{listId}/members	MemberCollection	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].

Resource	Base URL: http://{serverRoot}/ addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
				TS 1.0].	1.0].	1.0]
Individual member in a list	{userId}/lists/{listId}/members/{memberId}	Member	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Attributes for a member in a list	{userId}/lists/{listId}/members/{memberId}/attributes	AttributeList	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Individual attribute for a member in a list	{userId}/lists/{listId}/members/{memberId}/attributes/{ResourceRelPath}	Attribute	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
List references	{userId}/lists/{listId}/listReferences	ListReferenceCollection	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Individual list reference	{userId}/lists/{listId}/listReferences/{ResourceRelPath}	common:Link	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].

Table 4 Contact Subscriptions management

Purpose: External requests

Resource	URL Base URL: http://{serverRoot}/ addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of External Requests	{userId}/externalRequest	ExternalRequestCollection is used for GET requests, and ExternalRequest is used for POST requests.	This operation retrieves all active External Requests (e.g. import operations).	No	Create a new External Request.	No

Resource	URL Base URL: http://{serverRoot}/ addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Individual External Request to/from a 3rd party system)	{userId}/externalRequest/{reqId}	ExternalRequest	This operation retrieves information about an individual External Request.	This operation updates an existing External Request.	No	This operation deletes an existing External Request.

Table 5 External Requests management

Purpose: Management of Access Permissions to allow client access to authorization rules for shared contacts and to profile

Resource	Base URL: http://{serverRoot}/ addressbook/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Authorization Rules	{userId}/authorizationRules	RuleList (Used for GET) Rule (Used for POST) common:ResourceReference (optional alternative for POST response)	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
Individual authorization rule	{userId}/authorizationRules/{ruleId}	Rule (Used for PUT/GET)	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].

Individual authorization rule data	/{userId}/authorizationRules/{ruleId}/[ResourceRelPath]	The data structure corresponds to the element pointed out by the request-URL. (Used for PUT/GET)	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].	As defined in [Addressbook TS 1.0].
------------------------------------	---	---	-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------

Table 6 Access Permissions management

5.2 CAB Data Types

5.2.1 XML Namespaces

The XML namespace for the CAB data types is:

urn:oma:xml:rest:netapi:cab: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.

5.2.2 Structures

The subsections of this section define the data structures used in this RESTful CAB APIs.

Some of the structures can be instantiated as so-called root elements, i.e. they define the type of a representation of a so-called Heavy-weight Resource.

The column [ResourceRelPath] in the tables below, if used, includes relative resource paths for Light-weight Resource URLs that are used to access individual elements in the data structure (so-called Light-weight Resources). A string from this column needs to be appended to the corresponding Heavy-weight Resource URL in order to create Light-weight Resource URL for that particular element in the data structure. "Not applicable" means that individual access to that element is not supported. The root element and data type of the resource associated with the [ResourceRelPath] are defined by the Element and Type columns in the row that defines the [ResourceRelPath].

For structures that contain elements which describe a user identifier, the statements in section 6 regarding 'tel', 'sip' and 'acr' URI schemes apply.

5.2.2.1 Type: ContactCollection

Collection of all individual contacts for a user, in the form of a flat unique list

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.1, and reused by CAB APIs.

5.2.2.2 Type: Contact

Individual contact information in the flat unique list of contacts

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.2, and reused by CAB APIs.

5.2.2.3 Type: SharedIdentity

List of publicly known identities

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.3, and reused by CAB APIs.

5.2.2.4 Type: AttributeList

Attributes of a list member (subscription-list entry), or a contact or a profile.

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.6, and it is used by CAB APIs only as part of the 'Profile', 'Member' and 'Contact' data type structures (and not for 'List').

5.2.2.5 Type: Attribute

Individual attribute of a list member (subscription-list entry), or a contact or a profile.

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.7, and it is and it is used by CAB APIs only as part of the 'Member', 'Profile' and 'Contact' data type structures (and not for 'List').

The following extension is introduced for support of profile, based on the extensibility of the Attribute data type through <any element>:

Element	Type	Optional	Description	Defined in	Reuse or extended use description
name	xsd:string	No	<p>The following names of the attribute are predefined by this specification, in addition to the ones available in [Addressbook TS 1.0] sect. 5.2.2.7:</p> <p>“Profile” is a reserved name and SHALL only be used to access the CAB PCC [CAB XDMS].</p> <p>“XMLObj” is a reserved name and SHALL only be used to access the entire XML document, or fragment of an XML document, e.g. CAB PCC as an opaque object.</p>	[Addressbook TS 1.0]	Element reused for CAB APIs, and extended with new pre-defined values: “XMLObj” and “Profile”.
value	xsd:string	Choice	Optional element; if present it provides the value of the attribute.	[Addressbook TS 1.0]	<p>Element reused for CAB API.</p> <p>Not used when name=“Profile”..</p>
objectValue	xsd:base64Binary	Choice	<p>Optional element; if present it contains a base64Binary object.</p> <p>In case name is “vCard2.1” this element SHALL contain a base64Binary encoded vCard2.1 object.</p> <p>In case name is “vCard3.0” this element SHALL contain a base64Binary encoded vCard3.0 object.</p> <p>In case the name is “XMLObj”, this element SHALL contain a base64Binary encoded XML object.</p>	[Addressbook TS 1.0]	Element reused for CAB APIs, and extended with provisions for the new pre-defined value: “XMLObj”.

XSD modelling uses an optional “choice” to select either a value or an objectValue or <any element>, or none of them.

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

5.2.2.6 Type: ListCollection

List of lists

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.4, and reused by CAB APIs.

5.2.2.7 Type: List

List of identities

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.5, and reused by CAB APIs, which further specify a pre-defined enumeration value of “CABSubscriptionList” for the listId, allocated to the CAB subscription-list. The List data type is reused by CAB APIs with the following clarifications:

Element	Type	Optional	Description	Defined in	Reuse or extended use description
listId	xsd:anyURI	Yes	As defined in [Addressbook TS 1.0]. Contains a relative URI representing the list identity (e.g. “friends”).	[Addressbook TS 1.0]	Element unchanged. Reused for CAB subscription-list, which SHALL have the pre-defined listId of “CABSubscriptionList”, added as a new enumeration value in this specification.
memberCollection	MemberCollection	Yes	As defined in [Addressbook TS 1.0]. Members in the list.	[Addressbook TS 1.0]	Element unchanged. Reused for CAB APIs, for the collection of subscription-list entries.
attributeList	AttributeList	Yes	As defined in [Addressbook TS 1.0]. Contains a list of attributes related to a list.	[Addressbook TS 1.0]	N/A Element not used for CAB APIs (subscription-list has no attributes).

A root element named list of type List is allowed in response bodies.

5.2.2.8 Type: MemberCollection

Collection of members that are part of a list (entries in CAB Subscription-list)

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.8, and it is used unchanged by CAB APIs for the subscription-list collection of entries.

5.2.2.9 Type: Member

Individual member information in a list

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.9, and it is used unchanged by CAB APIs for the subscription-list <entry> elements, with the following clarifications:

Element	Type	Optional	Description	Defined in	Reuse or extended use description
memberId	xsd:anyURI	Yes	As defined in [Addressbook TS 1.0]. Contains an identifier of a single member (e.g. 'sip' URI, 'tel' URI, 'acr' URI).	[Addressbook TS 1.0]	Element unchanged. Reused for CAB subscription-list, equivalent to the 'id' attribute of the <entry>element of subscription-list..
attributeList	AttributeList	Yes	As defined in [Addressbook TS 1.0]. Contains a list of attributes (e.g. filter-set) related to a member.	[Addressbook TS 1.0]	Element unchanged. Reused for CAB APIs for the attributes of an <entry> in the <subscription-list>, such as the filter-set.

5.2.2.10 Type: ListReferenceCollection

List of references to other lists

This data type is defined in the [Addressbook TS 1.0].sect. 5.2.2.10, and it is used unchanged by CAB APIs for the subscription-list <entry> elements that contain an XUI of a list.

5.2.2.11 Type: AbChangesSubscriptionCollection

Collection of subscriptions to notifications of changes in contacts, profile or CABSubscriptionList.

This data type is defined in the [Addressbook TS 1.0], sect 5.2.2.11 “Type: AbChangesSubscriptionCollection” and it is used in CAB APIs to create subscriptions to one of:

- a user’s “profile” data changes,
- a CAB User’s subscription-list changes, and
- a user’s Address Book (AB) changes.

Element	Type	Optional	Description	Defined in	Reuse or extended use description
abChangesSubscription	AbChangesSubscription [0..unbounded]	Yes	Contains a list of subscriptions related to contacts changes and/or list changes.	[Addressbook TS 1.0]	Element unchanged. Reused for CAB AB.

resourceURL	xsd:anyURI	No	Self referring URL	[Addressbook TS 1.0]	Element unchanged. Reused for CAB AB.
-------------	------------	----	--------------------	----------------------	---------------------------------------

A root element named abChangesSubscriptionCollection of type AbChangesSubscriptionCollection is allowed in response bodies.

5.2.2.12 Type: AbChangesSubscription

This data type is defined in the [Addressbook TS 1.0], sect 5.2.2.12 “Type: AbChangesSubscription” and it is used in CAB APIs to realize subscriptions to:

- a user’s “profile” data changes,
- a CAB User’s subscription-list changes, and
- a user’s Address Book (AB) changes.

The following “profile” extension is added to the reused “AbChangesSubscription” data type definition from [Addressbook TS 1.0], and additional clarifications on the reuse and usage in CAB APIs of the existing structure for “anyContacts” and “listId” is made as follows:

Element	Type	Optional	Description	Defined in	Reuse or extended use description
anyContacts	empty	Choice	Indicates a subscription for changes about any contacts in the ContactCollection.	[Addressbook TS 1.0]	Element unchanged. Reused for CAB AB.
profile	empty	Choice	Indicates a subscription for changes about the user’s profile (e.g. CAB PCC).	CAB APIs TS 1.0	Added in CAB APIs TS 1.0
listId	xsd:anyURI	Choice	Indicates a subscription for changes about the specified list.	[Addressbook TS 1.0]	Element unchanged. Reused for the CAB subscription-list.

5.2.2.13 Type: AbChangeNotification

This data type is defined in the [Addressbook TS 1.0], sect 5.2.2.13 “Type: AbChangeNotification” and it is used in CAB APIs to generate notifications in the following cases:

- a contact is added to the contact collection
- contact is removed from the contact collection- profile is removed from the profile collection
- attribute related to the profile or a contact is updated (attribute added, attribute value updated or attribute removed)
- changes occurred in the CAB User’s subscription-list (contact added, contact removed).
- to indicate that the subscription has been terminated by the server.
(ResourceStatus=TerminatedTimeout/TerminatedNoResource/TerminatedOther)

The following extensions are added to the reused AbChangesNotifications data type definition from [Addressbook TS 1.0]:

Element	Type	Optional	Description	Defined in	Reuse or extended use description
link	common: Link [1..unbounded]	No	<p>SHALL contain reference to the added or updated member or contact (rel="Member" or rel="Contact" respectively).</p> <p>A notification MAY include several links for each member or contact.</p> <p>SHALL contain reference to the list in case a member or contact has been removed (rel="List" or rel="ContactCollection" respectively).</p> <p>SHALL contain reference to the profile if it has been updated (rel="Profile")</p> <p>SHALL contain reference to the profile collection in case a profile has been removed (rel="ProfileCollection")</p> <p>SHALL contain reference to the subscription (rel="AbChangesSubscription") if resourceStatus is 'Active'.</p>	[Addressbook TS 1.0]	Element updated. Added "profile" support (e.g. CAB PCC).

A root element named abChangeNotification of type AbChangeNotification is used by server in the POST request to the client informing it about change in the profile/list that client is monitoring (using one of the subscriptions).

5.2.2.14 Type: ProfileCollection

Collection of user's Profile data such as the CAB PCC (in different format, from different sources)

Element	Type	Optional	Description	Defined in	Reuse or extended use description
profile	Profile [0..unbounded]	Yes	Contains a list of user's profiles.	CAB APIs TS 1.0.	Added in CAB APIs TS 1.0.
resourceURL	xsd:anyURI	No	Self referring URL	CAB APIs TS 1.0.	Element replicated from [Addressbook TS 1.0].

A root element named profileCollection of type ProfileCollection is allowed in response bodies.

5.2.2.15 Type: Profile

User's individual Profile

Element	Type	Optional	Description	Defined in	Reuse or extended use description
profileId	xsd:anyURI	Yes	Contains a relative URI of the profile (e.g. "AlicePCC"). The element is mandatory when used inside 'ProfileCollection' type. It is a key property of the element and SHALL NOT be altered when included in the Light-weight Resource URL. If included, and if profileId is also part of the request URL, the two MUST have the same value.	CAB APIs TS 1.0.	Added in CAB APIs TS 1.0.
attributeList	AttributeList	Yes	Contains a list of attributes related to a profile,	[Addressbook TS 1.0]	Element unchanged. Reused for CAB APIs.
xcapNodeSelector	xsd:anyURI	Yes	Contains the XCAP node selector to an XML node (e.g. the Profile (PCC)).	CAB APIs TS 1.0	Added in CAB APIs TS 1.0 It is used when Attribute name is any one of "Profile" or "XMLObj".

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

5.2.2.16 Type: ApplicableToProfile

Profile the rule applies to

Element	Type	Optional	Description
profileId	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified list of 'profileId' that are part of the 'ProfileCollection'.
allProfiles	(empty)	Choice	The rule applies to all profiles.

XSD modelling uses an optional "choice" to select either profileId or allProfiles.

5.2.2.17 Type: ApplicableToProfileNode

Profile XML node(s) the rule applies to

Element	Type	Optional	Description
profileId	xsd:anyURI [0..unbounded]	No	The rule applies to the specified 'profileId'.
xcapNodeSelector	xsd:anyURI [0..unbounded]	Choice	The rule applies to the specified list of 'xcapNodeSelector' that are part of the 'Profile'.
contactView	xsd:string	Choice	The rule applies to the listed Contact View(s). Multiple values are delimited by separator.

XSD modelling uses an optional "choice" to select either xcapNodeSelector or contactView, or both.

5.2.2.18 Type: ApplicableToContacts

Contacts the rule applies to, as defined in the [Addressbook TS 1.0], sect 5.2.2.16 "Type:ApplicableToContacts".

5.2.2.19 Type: ExternalRequestCollection

Element	Type	Optional	Description
externalRequest	ExternalRequest [0..unbounded]	Yes	Contains a list of external requests.
resourceURL	xsd:anyURI	No	Self referring URL.

A root element named externalRequestCollection of type ExternalRequestCollection is allowed in response bodies.

5.2.2.20 Type: ExternalRequest

Element	Type	Optional	Description
importNonCAB	empty	Choice	Indicates the request is an import of a non-CAB Address Book contacts
importProfile	empty	Choice	Indicates the request is an import of Profile data from 3 rd Party.
exportProfile	empty	Choice	Indicates the request is an export of PCC data to a 3 rd Party profile.
subscribeProfile	empty	Choice	Indicates the request is subscription to CAB User's personal profile information in 3 rd party system
nonCABSource	xsd:string	No	element indicating the source (e.g. domain name) of the external system
scheduledInterval	xsd:int	No	Number of hours between the successive exchanges with external system. It shall be set to '0' in case of subscription to CAB User's personal profile information in 3 rd party system, or in case of a single request not intended for periodic repetition
username	xsd:string	Yes	User name that is used to access the external system. The use of username and password is one of the alternatives to

			access the external system.
password	xsd:string	Yes	Password that is used to access the external system. . The use of username and password is one of the alternatives to access the external system.
token	xsd:string	Yes	A token that is used to access the external system. . The use of a token is one of the alternatives to access the external system.
filterSet	filter:filter-set	Yes	Filter to be applied to the imported/exported/subscribed Profile data. This element may be present in import-Profile, export-Profile and subscribe-Profile operations, and it is defined in and [OMA XDM Core] and [RFC4661].
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server. This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
applicationTag	xsd:string	Yes	A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.
resourceURL	xsd:anyURI	Yes	Self referring URL.

A single element from the choice of elements describing the type of External Request (importNonCAB, importProfile, exportProfile, subscribeProfile) SHALL be conveyed in elements of type ExternalRequest

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

5.2.3 Enumerations

The subsections of this section define the enumerations used in this RESTful CAB API.

5.2.3.1 Enumeration: ListType

Enumeration of types of lists

This enumeration is defined in [Addressbook TS 1.0] sect. 5.2.3.1 and it is reused and extended by CAB APIs with the following new enumeration value:

Enumeration	Description	Defined in	Reuse or extended use description
CABSubscriptionList	Indicates that this list is used for CAB subscription-list, needed for ContactSubscription.	CAB APIs TS 1.0	Extension defined by CAB APIs.

5.2.3.2 Enumeration: ResourceStatus

Enumeration of possible status of a subscription

This enumeration is defined in [Addressbook TS 1.0] sect. 5.2.3.2 and it is reused by CAB APIs.

5.2.4 Values of the Link “rel” attribute

The “rel” attribute of the Link element is a free string set by the server implementation, to indicate a relationship between the current resource and an external resource. The following are possible strings specific for CAB API on top of the ones defined in [Addressbook TS 1.0] (list is non-exhaustive, and can be extended):

- Profile
- ProfileCollection

These values indicate the kind of resource that the link points to.

5.3 Sequence Diagrams

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

5.3.1 Managing contacts in a flat list

Refer to the flow described in [Addressbook TS 1.0] sect. 5.3.1.

5.3.2 Accessing the lists and members

Refer to the flow described in [Addressbook TS 1.0] sect. 5.3.2

5.3.3 Subscribing to address book changes and receiving notifications

Refer to the flow described in [Addressbook TS 1.0] sect. 5.3.3.

5.3.4 Managing authorization rules

Refer to the flow described in [Addressbook TS 1.0] sect. 5.3.5.

6. Detailed specification of the resources

The following applies to all resources defined in this specification regardless of the representation format (i.e. XML, JSON, application/x-www-form-urlencoded):

- 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 an Anonymous Customer Reference (ACR), it **MUST** be defined according to [IETF_ACR_draft], i.e. it **MUST** include the protocol prefix 'acr:' followed by the ACR.
 - The ACR ‘auth’ is a supported reserved keyword, and **MUST NOT** be assigned as an ACR to any particular end user. See [Addressbook TS 1.0] Appendix 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, and **MAY** support application/x-www-form-urlencoded 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: Collection of contacts

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts

and it is specified within [Addressbook TS 1.0] sect.6.1.

6.2 Resource: Individual contact

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}

and it is specified within [Addressbook TS 1.0] sect.6.2.

6.3 Resource: Attributes for a contact

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes

and it is specified within [Addressbook TS 1.0] sect.6.3.

6.4 Resource: Individual attribute for a contact

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/contacts/{contactId}/attributes/[ResourceRelPath]

and it is specified within [Addressbook TS 1.0] sect.6.4.

6.5 Resource: Lists

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists

This resource is used to retrieve all lists belonging to a user and it is specified within [Addressbook TS 1.0] sect.6.5.

6.6 Resource: Individual list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}

and it is specified within [Addressbook TS 1.0] sect.6.6. This is used in CAB APIs for the CAB subscription-list.

6.7 Resource: Members in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members

and it is specified within [Addressbook TS 1.0] sect.6.9. This is used in CAB APIs for the retrieval of entries in a CAB User's subscription-list.

6.8 Resource: Individual member in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}

and it is specified within [Addressbook TS 1.0] sect.6.10. This is used in CAB APIs for the management of entries in the CAB User's subscription-list.

6.9 Resource: Attributes for a member in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}/attributes

and it is specified within [Addressbook TS 1.0] sect.6.11. This is used in CAB APIs to retrieve all attributes of entries in the CAB User's subscription-list (e.g. all filter-set).

6.10 Resource: Individual attribute for a member in a list

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/members/{memberId}/attributes/[ResourceRelPath]

and it is specified within [Addressbook TS 1.0] sect.6.12. This is used in CAB APIs to manage the attributes of entries in the CAB User's subscription-list (e.g. the filter-set).

6.11 Resource: List references

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/listReferences

and it is specified within [Addressbook TS 1.0] sect.6.13.

6.12 Resource: Individual list reference

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/lists/{listId}/listReferences/[ResourceRelPath]

and it is specified within [Addressbook TS 1.0] sect.6.14.

6.13 Resource: Address Book changes subscriptions

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/subscriptions/abChanges

and it is specified within [Addressbook TS 1.0] sect.6.15, with additional CAB APIs extensions specified in sect. 5.2.2.1. This resource is used to retrieve and to create new subscriptions to any one of the following:

- a user's "profile" data changes,
- a CAB User's subscription-list changes, and
- a user's Address Book (AB) changes.

6.14 Resource: Individual list changes subscription

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/subscriptions/abChanges/{subscriptionId}

and it is specified within [Addressbook TS 1.0] sect.6.16, with additional CAB APIs extensions specified in sect. 5.2.2.1. This resource is used to retrieve or remove subscriptions to any one of the following:

- a user's "profile" data changes,
- a CAB User's subscription-list changes, and
- a user's Address Book (AB) changes.

6.15 Resource: Client resource for list changes notifications

The resource is specified within [Addressbook TS 1.0] sect.6.17 and it is a callback URL provided by the client for notification about outbound message delivery status.

A notification SHALL be generated by the system exposing the CAB APIs for all events specified in [Addressbook TS 1.0] table in sect. 6.17, with the following additions and clarifications:

Type of notification	Value of ResourceStatus	Generated in the following occasions:	Defined in	Reuse or extended use description
Update profile	Active	A profile or attributes of a profile were added, profile attribute(s) value updated or profile attribute removed.	CAB APIs 1.0	Added in CAB APIs TS 1.0
Remove profile	Active	A profile was removed.	CAB APIs 1.0	Added in CAB APIs TS 1.0
Update contact collection or list attributes	Active	A contact collection's attributes are updated (attribute added, changed or removed).	[Addressbook TS 1.0]	The list attributes are not supported for CAB APIs.

6.16 Resource: Collection of Profiles

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/profile

This resource is used to retrieve a flat list of all profiles, and/or sub-structures of a profile owned by a given user.

6.16.1 Request URL variables

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

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	Version of the API clients want to use. The value of this variable is defined in section 5.1.
userId	User identifier. Examples: tel:+19585550100, acr:pseudonym123

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

6.16.2 Response Codes and Exceptions

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Restful CAB APIs, see section 7.

6.16.3 GET

This resource is used to retrieve a list of profiles and/or sub-structures of a profile owned by a given user.

6.16.3.1 Example 1: Retrieve all profiles (Informative)

6.16.3.1.1 Request

GET /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile?resFormat=XML&indivFilter=Profile HTTP/1.1

```
Accept: application/xml
Host: example.com
```

6.16.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 03 Feb 2012 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<cab:profileCollection xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-v1_0.xsd"
  xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">

  <profile>
    <profileId>mariaPCC</profileId>
    <attributeList>
      <attribute>
        <name>Profile</name>

        <pcc pcc-type="individual" xmlns="urn:oma:xml:cab:pcc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
          <person-details index="gt4fd890bu8">

            <name index="fg4fd890der">

              <name-entry index="dslkhdsjk" name-type="LegalName" view-type="work">
                <title display-order="1">Mr.</title>
                <given display-order="1">Joe</given>
                <family display-order="1">Smith</family>
                <gen-id display-order="1">Jr.</gen-id>
                <degree display-order="1">PhD</degree>
                <display-name>Joe Smith</display-name>

              </name-entry>

              <name-entry index="riuetutl" name-type="KnownAs" view-type="personal">
                <given display-order="1">Joe</given>
                <family display-order="1">Smith</family>
                <display-name>Joe</display-name>

              </name-entry>

            </name>

            <comm-addr xml:lang="en">
              <uri-entry index="glrjgil" addr-uri-type="Home SIP-URI" view-type="personal">

                <addr-uri>sip:joe.smith@example.com;</addr-uri>

                <label>My IP Phone</label>
              </uri-entry>
              <uri-entry index="asdhfdsag" addr-uri-type="Email">
```

```

        <addr-uri>mailto:jboggs@example.com</addr-uri>

        <label>My personal Email</label>
    </uri-entry>
    <tel index="nbvfjf" tel-type="Mobile" view-type="work">

        <tel-nb>

            <tel-str>1-222-222-2222</tel-str>

        </tel-nb>

        <label>My mobile Phone at work</label>
    </tel>
    <tel index="ijlfushf" tel-type="Home" view-type="personal">
        <tel-nb>
            <tel-str>1-111-111-1111</tel-str>
        </tel-nb>
        <label>My phone at home</label>
    </tel>
    <tel index="ijlfdferf" tel-type="Other" view-type="personal,work">
        <tel-nb>
            <tel-str>+1-333-333-3333</tel-str>
        </tel-nb>
        <label>Other</label>
    </tel>

    </comm-addr>

    <birth xml:lang="en" view-type="personal">
    <birth-date>

        <date>1957-07-09T06:01:00</date>

    </birth-date>
    <place index="hgdfersx">Saint Joesph Hospital Little Rock, Arkansas</place>
    </birth>
</person-details>
</pcc>
</attribute>

<resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC</resourceURL>
</attributeList>

<xcapNodeSelector>/org.openmobilealliance.cab-pcc/users/maria%40example.com/pcc</xcapNodeSelector>
</profile>

<profile>
    <profileId>mariaSocialPCC</profileId>
    <attributeList>
        <attribute>

```

```

    <name>Profile</name>

    </attribute>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaSocialPCC</resourceURL>
    </attributeList>
    <xcapNodeSelector>/org.openmobilealliance.cab-pcc/users/maria%40example.com/Socialpcc</xcapNodeSelector>
  </profile>

  <profile>
    <profileId>mariaPCCvCard</profileId>
    <attributeList>
      <attribute>
        <name>vCard2.1</name>
        <objectValue>aaaa</objectValue>
      </attribute>

      <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCCvCard</resourceURL>
    </attributeList>
    <xcapNodeSelector />
  </profile>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile</resourceURL>
</cab:profileCollection>

```

6.16.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.16.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.16.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: GET’ field in the response as per section 14.7 of [RFC 2616].

6.17 Resource: Individual profile

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/profile/{profileId}

This resource is used to retrieve, create, update or delete individual profile from the collection of profiles of a given user.

6.17.1 Request URL variables

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

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:pseudonym123
profileId	Profile identifier. Examples: AlicePCC, AlicePCCvcard, AliceExtProfile.

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

6.17.2 Response Codes and Exceptions

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to the RESTful CAB API, see section 7.

6.17.3 GET

This resource is used to retrieve individual profile data from a list of profiles available for a given user. The data is retrieved as an opaque XML object.

Supported parameters in the query string of the Request URL are:

Name	Type/Values	Optional	Description
indivFilter	xsd:string	Yes	<p>Defines the level of information related to a specific profile, returned in a GET response body.</p> <p>If indivFilter is absent, GET response body SHALL include all profile sub-elements, plus all profile attributes (default behaviour).</p> <p>If indivFilterValue = Profile attribute name (e.g. name1), GET response body SHALL include all profile sub-elements, plus only selected profile attributes (i.e. name1). Any number of indivFilter=indivFilterValue pairs separated by & are allowed.</p> <p>No other indivFilterValue is supported.</p>

6.17.3.1 Request

```
GET /org.openmobilealliance.cab-pcc/users/sip%3Amaria%40example.com/pcc/mariaPCC?indivFilter=XMLObj HTTP/1.1
Accept: application/xml
Host: example.com
```

6.17.3.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 03 Feb 2012 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cab:profile xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-v1_0.xsd"
xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">

  <profileId>mariaPCC</profileId>
  <attributeList>
```



```

Content-Length : nnnn
<?xml version="1.0" encoding="UTF-8"?>
<cab:profile xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-v1_0.xsd"
xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <profileId>mariaPCC</profileId>
  <attributeList>
    <attribute>
      <name>XMLObj</name>

<objectValue>PD94bWwgdMvYc2lrbj0iMS4wliBlbmNvZGluZz0iVVRGLTgiPz4NCjxwY2MgcGNjLXR5cGU9ImluZGI2aWR1YWwiiHhtbG5zPSJ1cm46b21
hOnhtbDpjYWl6cGNjiiB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzlwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiA+DQogJCAgIDxwZX
Jzb24tZGV0YWlscyBpbmRleD0iZ3Q0ZmQ0OTBidTgiPg0KICAgJCAgJCAgJCA8YWRkcmVzc24NCgk8YWRkcmVzcy1lbnRyeSBpbmRleD0iZ
GFzZmRoYXNsiBhZGRyLXR5cGU9IldvcmsiPg0KCSAgJCAgPg0KCSAgJCAgJCA8dGltZXpvmU+DQoJCTx0ei11cmw
+bWFwcy5leGFtcGxILmNvbS8xMjM0PC90ei11cmw+DQoJICAgJCAgJCAgIDwvdGltZXpvmU+DQoJICAgJCA8L2xvY2F0aW9uPg0KCTwvY
WRkcmVzcy1lbnRyeT4NCiAgJCAgJCAgIDwvYWRkcmVzc24NCjwvcGVyc29uLWRldGFpbHM+DQo8L3BjYz4NCg== </objectValue>
    </attribute>
    <resourceURL>http://example.com/org.openmobilealliance.cab-
pcc/users/sip%3Amaria%40example.com/profile/mariaPCC/attributes/pcc%5D %5B@pcc-type=%22individual%22/~~/pcc/person-details%5B@index=%2
gt4fd890bu8%22%5D/address/address-entry%5B@index=%22dasfdhas%22%5D %5B@addr-type=%22Work%5D/location/timezone/tz-
url</resourceURL>
  </attributeList>
  <xcapNodeSelector>/org.openmobilealliance.cab-pcc/users/sip%3Amaria%40example.com/mariaPCC </xcapNodeSelector>
</cab:profile>

```

6.17.4.2 Example: Update attributes of an existing profile using “Profile”(Informative)

The PUT example shows how a node in the profile is updated, using the “Profile” option and a value containing the XML data compliant with CAB PCC [CAB XDMS].

6.17.4.2.1 Request

```

PUT /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cab:profile xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1
http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-v1_0.xsd urn:oma:xml:rest:netapi:addressbook:1
http://www.openmobilealliance.org/tech/profiles/rest_netapi_addressbook-v1_0.xsd"
xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
  <profileId>mariaPCC</profileId>
  <attributeList>
    <attribute>
      <name>Profile</name>

<pcc pcc-type="individual" xmlns="urn:oma:xml:cab:pcc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <person-details index="gt4fd890bu8">
    <name index="fg4fd890der">
      <name-entry index="dslkhdskj" name-type="LegalName" view-type="work">
        <title display-order="1">Mr.</title>

```

```

    <given display-order="1">Joe</given>
    <family display-order="1">Smith</family>
    <gen-id display-order="1">Jr.</gen-id>
    <degree display-order="1">PhD</degree>
    <display-name>Joe Smith</display-name>
  </name-entry>
  <name-entry index="riuetul" name-type="KnownAs" view-type="personal">
    <given display-order="1">Joe</given>
    <family display-order="1">Smith</family>
    <display-name>Joe</display-name>
  </name-entry>
</name>

<comm-addr xml:lang="en">
  <uri-entry index="glrjgil" addr-uri-type="Home SIP-URI" view-type="personal">
    <addr-uri>sip:joe.smith@example.com;</addr-uri>
    <label>My IP Phone</label>
  </uri-entry>
  <uri-entry index="asdhfdsag" addr-uri-type="Email">
    <addr-uri>mailto:jboggs@example.com</addr-uri>
    <label>My personal Email</label>
  </uri-entry>
  <tel index="nbvfjf" tel-type="Mobile" view-type="work">
    <tel-nb>
      <tel-str>1-222-222-2222</tel-str>
    </tel-nb>
    <label>My mobile Phone at work</label>
  </tel>
  <tel index="ijlfushf" tel-type="Home" view-type="personal">
    <tel-nb>
      <tel-str>1-111-111-1111</tel-str>
    </tel-nb>
    <label>My phone at home</label>
  </tel>
  <tel index="ijlfdferf" tel-type="Other" view-type="personal,work">
    <tel-nb>
      <tel-str>+1-333-333-3333</tel-str>
    </tel-nb>
    <label>Other</label>
  </tel>
</comm-addr>
<birth xml:lang="en" view-type="personal">
  <birth-date>
    <date>1957-07-09T06:01:00</date>
  </birth-date>
  <place index="hgdfersx">Saint Joesph Hospital Little Rock, Arkansas</place>
</birth>
</person-details>
</pcc>

  </attribute>

  <resourceURL> /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC</resourceURL>
</attributeList>
  <xcapNodeSelector>/org.openmobilealliance.cab-pcc/users/sip%3Amaria%40example.com/mariaPCC/pcc
</xcapNodeSelector>

```

```
</cab:profile>
```

6.17.4.2.2 Response

HTTP/1.1 200 OK

Date: Fri, 10 Dec 2010 12:51:59 GMT

Location: http://exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC

Content-Type: application/xml

Content-Length : nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<cab:profile xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-
v1_0.xsd"
    xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
```

```
<profileId>mariaPCC</profileId>
```

```
<attributeList>
```

```
<attribute>
```

```
<name>Profile</name>
```

```
<pcc pcc-type="individual" xmlns="urn:oma:xml:cab:pcc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<person-details index="gt4fd890bu8">
```

```
<name index="fg4fd890der">
```

```
<name-entry index="dslkhdsj" name-type="LegalName" view-type="work">
```

```
<title display-order="1">Mr.</title>
```

```
<given display-order="1">Joe</given>
```

```
<family display-order="1">Smith</family>
```

```
<gen-id display-order="1">Jr.</gen-id>
```

```
<degree display-order="1">PhD</degree>
```

```
<display-name>Joe Smith</display-name>
```

```
</name-entry>
```

```
<name-entry index="riuetutl" name-type="KnownAs" view-type="personal">
```

```
<given display-order="1">Joe</given>
```

```
<family display-order="1">Smith</family>
```

```
<display-name>Joe</display-name>
```

```
</name-entry>
```

```
</name>
```

```
<comm-addr xml:lang="en">
```

```
<uri-entry index="glrjgil" addr-uri-type="Home SIP-URI" view-type="personal">
```

```
<addr-uri>sip:joe.smith@example.com;</addr-uri>
```

```
<label>My IP Phone</label>
```

```
</uri-entry>
```

```
<uri-entry index="asdhfdsag" addr-uri-type="Email">
```

```
<addr-uri>mailto:jboggs@example.com</addr-uri>
```

```
<label>My personal Email</label>
```

```
</uri-entry>
```

```
<tel index="nbvfjf" tel-type="Mobile" view-type="work">
```

```
<tel-nb>
```

```
<tel-str>1-222-222-2222</tel-str>
```

```
</tel-nb>
```

```
<label>My mobile Phone at work</label>
```

```

        </tel>
        <tel index="ijfushf" tel-type="Home" view-type="personal">
            <tel-nb>
                <tel-str>1-111-111-1111</tel-str>
            </tel-nb>
            <label>My phone at home</label>
        </tel>
        <tel index="ijfdferf" tel-type="Other" view-type="personal,work">
            <tel-nb>
                <tel-str>+1-333-333-3333</tel-str>
            </tel-nb>
            <label>Other</label>
        </tel>
    </comm-addr>
    <birth xml:lang="en" view-type="personal">
        <birth-date>
            <date>1957-07-09T06:01:00</date>
        </birth-date>
        <place index="hgdfersx">Saint Joesph Hospital Little Rock, Arkansas</place>
    </birth>
</person-details>
</pcc>

</attribute>
<resourceURL>/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC</resourceURL>
</attributeList>
<xcapNodeSelector> /org.openmobilealliance.cab-pcc/users/sip%3Amaria%40example.com/mariaPCC/pcc </xcapNodeSelector>

</cab:profile>

```

6.17.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.17.6 DELETE

This operation deletes a profile.

6.17.6.1 Example: Delete a profile

(Informative)

6.17.6.1.1 Request

```

DELETE /org.openmobilealliance.cab-pcc/users/sip%3Amaria%40example.com/profile/mariaPCC HTTP/1.1
Accept: application/xml
Host: example.com

```

6.17.6.1.2 Response

```

HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT

```


6.18 Resource: Attributes of a profile

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/profile/{profileId}/attributes

This resource is used to retrieve all attributes for a profile.

6.18.1 Request URL variables

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

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: sip:maria@example.com
profileId	Profile identifier. Example: mariaPCC

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

6.18.2 Response Codes and Exceptions

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to the RESTful CAB API, see section 7.

6.18.3 GET

This operation is used for retrieval of all attributes of a profile for a given user identity.

6.18.3.1 Example: Retrieve all attributes of a profile (Informative)

Retrieve all attributes of a profile belonging to a user, and return result in XML format, using resFormat parameter in URL. The attributes of a profile are Person-Details, Org-Details and Group-Details, This example contains a profile that only has one Person-Details and 0 Org-Details and 0 Group-Details.

6.18.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC/attributes?resFormat=XML HTTP/1.1
Accept: application/xml
Host: example.com
```

6.18.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cab:profile xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-
v1_0.xsd"
    xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1">
```

```

<attributeList>
  <attribute>
    <name>Profile</name>

<pcc pcc-type="individual" xmlns="urn:oma:xml:cab:pcc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <person-details index="gt4fd890bu8">
    <name index="fg4fd890der">
      <name-entry index="dslkhdsj" name-type="LegalName" view-type="work">
        <title display-order="1">Mr.</title>
        <given display-order="1">Joe</given>
        <family display-order="1">Smith</family>
        <gen-id display-order="1">Jr.</gen-id>
        <degree display-order="1">PhD</degree>
        <display-name>Joe Smith</display-name>
      </name-entry>
      <name-entry index="riuetul" name-type="KnownAs" view-type="personal">
        <given display-order="1">Joe</given>
        <family display-order="1">Smith</family>
        <display-name>Joe</display-name>
      </name-entry>
    </name>
    <comm-addr xml:lang="en">
      <uri-entry index="glrjgii" addr-uri-type="Home SIP-URI" view-type="personal">
        <addr-uri>sip:joe.smith@example.com;</addr-uri>
        <label>My IP Phone</label>
      </uri-entry>
      <uri-entry index="asdhfdsag" addr-uri-type="Email">
        <addr-uri>mailto:jboggs@example.com</addr-uri>
        <label>My personal Email</label>
      </uri-entry>
      <tel index="nbvfjj" tel-type="Mobile" view-type="work">
        <tel-nb>
          <tel-str>1-222-222-2222</tel-str>
        </tel-nb>
        <label>My mobile Phone at work</label>
      </tel>
      <tel index="ijlfushf" tel-type="Home" view-type="personal">
        <tel-nb>
          <tel-str>1-111-111-1111</tel-str>
        </tel-nb>
        <label>My phone at home</label>
      </tel>
      <tel index="ijlfdferf" tel-type="Other" view-type="personal,work">
        <tel-nb>
          <tel-str>+1-333-333-3333</tel-str>
        </tel-nb>
        <label>Other</label>
      </tel>
    </comm-addr>
    <birth xml:lang="en" view-type="personal">
      <birth-date>
        <date>1957-07-09T06:01:00</date>
      </birth-date>
      <place index="hgdfersx">Saint Joesph Hospital Little Rock, Arkansas</place>
    </birth>

```



```

    </person-details>
</pcc>

    </attribute>
<resourceURL>http://exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC </resourceURL>

</attributeList>
    <xcapNodeSelector />
</cab:profile>

```

6.18.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

6.18.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

6.18.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

6.19 Resource: Individual attribute for a profile

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/profile/{profileId}/attributes/[ResourceRelPath]

This Light-weight Resource is used to manage attributes for a profile, which include creation, retrieval, update and delete operations.

6.19.1 Request URL variables

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

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: sip: maria@example.com
profileId	Contact identifier. Example: mariaPCC
[ResourceRelPath]	Light-weight relative resource path. The allowed string for this Light-weight Resource is {name}, as defined in the [ResourceRelPath] column in table 5.2.2.6. {name} indicates the name of the attribute. Example: xcapNodeSelector of the attribute tz-url of the CAB PCC.

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

6.19.2 Response Codes and Exceptions

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to the RESTful CAB API, see section 7.

6.19.3 GET

This operation is used for retrieval of a specific attribute value of a profile of a user, in this example the tz-url value.

6.19.3.1 Example : Retrieve a profile's attribute (Informative)

6.19.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC/attributes/~~/pcc/person-
details %5B@index=%22gt4fd890bu8%22%5D/address/address-entry%5B@index=%22dasfdhasl%22%5D%5B@addr-
type=%22Work%22%5D/location/timezone/tz-url HTTP/1.1
Accept: application/xml
Host: example.com
```

6.19.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:addressbook:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_addressbook-
v1_0.xsd">

  <name>Profile</name>

  <pcc pcc-type="individual" xmlns="urn:oma:xml:cab:pcc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
    <person-details index="gt4fd890bu8">
      <address>
        <address-entry index="dasfdhasl" addr-type="Work">
          <location>
            <time-zone>
              <tz-url>maps.com/1234</tz-url>
            </time-zone>
          </location>
        </address-entry>
      </address>
    </person-details>
  </pcc>

</ab:attribute>
```

6.19.4 PUT

This operation is used for creation or update of an attribute of a profile. If successful, this SHALL also create or update the attribute for a user's profile.

6.19.4.1 Example : Update an existing profile with new attribute (Informative)

The PUT example shows how a node in the profile is updated with the new value "maps.com" for the tz-url, using the "Profile" option and a value containing the XML data compliant with CAB PCC [CAB XDMS].

6.19.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/ariaPCC/attributes/~pcc%5D%5B@pcc-
type=%22individual%22%5D%5B@index=%22gt4fd890bu8%22%5D/address/address-
entry%5B@index=%22dasfdhas!%22%5D%5B@addr-type=%22Work%22%5D/location/timezone/tz-url
HTTP/1.1
```

```
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length : nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ab:attribute xmlns:ab="urn:oma:xml:rest:netapi:addressbook:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:addressbook:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_addressbook-
v1_0.xsd">
```

```
<name>Profile</name>
```

```
<pcc pcc-type="individual" xmlns="urn:oma:xml:cab:pcc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <person-details index="gt4fd890bu8">
    <address>
      <address-entry index="dasfdhas" addr-type="Work">
        <location>
          <time-zone>
            <tz-url>maps.com/1234</tz-url>
          </time-zone>
        </location>
      </address-entry>
    </address>
  </person-details>
</pcc>
```

```
</ab:attribute>
```

6.19.4.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Dec 2010 12:51:59 GMT
Location:http://exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/ariaPCC/attributes/~pcc%5D%5B@pcc-
type=%22individual%22%5D%5B@index=%22gt4fd890bu8%22%5D/address/address-
entry%5B@index=%22dasfdhas!%22%5D%5B@addr-type=%22Work%22%5D/location/timezone/tz-url
```

Content-Type: application/xml
Content-Length : nnnn

6.19.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.19.6 DELETE

This operation deletes an attribute of a profile, the pcc:tz-url. If successful, this SHALL delete the attribute from the specific user's profile.

6.19.6.1 Request

```
DELETE /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/profile/mariaPCC/attributes/~-pcc%5D%5B@pcc-
type=%22individual%22%5D%5B@index=%22gt4fd890bu8%22%5D/address/address-
entry%5B@index=%22dasfdhasl%22%5D%5B@addr-type=%22Work%22%5D/location/timezone/tz-url HTTP/1.1
```

Accept: application/xml
Host: example.com

6.19.6.2 Response

HTTP/1.1 204 No Content
Date: Fri, 10 Dec 2010 12:53:23 GMT

6.20 Resource: Authorization rules

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules

and it is specified within [Addressbook TS 1.0] sect.6.23.

This resource is used to retrieve rules for access to profile and contacts, as well as for the creation of new authorization rules.

6.21 Resource: Individual authorization rule

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}

and it is specified within [Addressbook TS 1.0] sect.6.24.

This resource is used to retrieve an individual authorization rule for access to profile or contacts.

6.22 Resource: Individual authorization rule data

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/authorizationRules/{ruleId}/[ResourceRelPath]

and it is specified within [Addressbook TS 1.0] sect.6.25.

This resource is used to retrieve an individual authorization rule for access to profile or contacts.

This Light-weight Resource is used to manage authorized identities in an existing authorization rule, which include creation, update, retrieval, and delete operations.

6.23 Resource: Collection of External Requests

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/externalRequest

This resource is used to retrieve the list of all active External Requests associated to a given user and to create new External Requests.

6.23.1 Request URL variables

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

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	Version of the API clients want to use. The value of this variable is defined in section 5.1.
userId	User identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 of [Addressbook TS 1.0] for a statement on the escaping of reserved characters in URL variables.

6.23.2 Response Codes and Exceptions

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Restful CAB APIs, see section 7.

6.23.3 GET

This resource is used to retrieve the list of all active External Requests associated to a given user.

6.23.3.1 Example 1: Retrieve all External Requests (Informative)

6.23.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalReq HTTP/1.1
Accept: application/xml
Host: example.com
```

6.23.3.1.2 Response

```
HTTP/1.1 200 OK
```

Date: Fri, 10 Feb 2012 12:51:59 GMT
 Content-Type: application/xml
 Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<cab:externalRequestCollection xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-
v1_0.xsd">

  <externalReq>
    <importProfile/>
    <nonCABSource>extSystem1</nonCABSource>
    <scheduledInterval>24</scheduledInterval>
    <username>maria</username>
    <password>maria123</password>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1234
    </resourceURL>
  </externalReq>
  <externalReq>
    <importNonCAB/>
    <nonCABSource>extSystem1</nonCABSource>
    <scheduledInterval>24</scheduledInterval>
    <username>maria</username>
    <password>maria123</password>
    <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1235
    </resourceURL>
  </externalReq>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest</resourceURL>
</cab:externalRequestCollection>
```

6.23.4 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 [RFC 2616].

6.23.5 POST

This resource is used to create new External Request

6.23.5.1 Example 1: Create a new External Request

(Informative)

6.23.5.1.1 Request

```
POST /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<cab:externalRequest xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1
http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-v1_0.xsd">
```

```
<exportProfile/>
<nonCABSource>extSystem2</nonCABSource>
<scheduledInterval>0</scheduledInterval>
<username>maria</username>
<password>maria345</password>
</cab:externalRequest>
```

6.23.5.1.2 Response

HTTP/1.1 201 Created
 Date: Fri, 10 Feb 2012 21:33:52 GMT
 Location: http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1236
 Content-Type: application/xml
 Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<cab:externalRequest xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-
v1_0.xsd">
  <exportProfile/>
  <nonCABSource>extSystem2</nonCABSource>
  <scheduledInterval>0</scheduledInterval>
  <username>maria</username>
  <password>maria345</password>
  <resourceURL> http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1236
  </resourceURL>
</cab:externalRequest>
```

6.23.6 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 [RFC 2616].

6.24 Resource: Individual External Request

The resource used is:

http://{serverRoot}/addressbook/{apiVersion}/{userId}/externalRequest/{reqId}

This resource is used to retrieve, update or delete individual External Requests from the collection of External Requests of a given user

6.24.1 Request URL variables

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

Name	Description
serverRoot	Server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: http://example.com/exampleAPI
apiVersion	Version of the API clients want to use. The value of this variable is defined in section 5.1.

userId	User identifier. Examples: tel:+19585550100, acr:pseudonym123
reqId	External Request identifier. Example: externalReq12356

See section 6 of [Addressbook TS 1.0] for a statement on the escaping of reserved characters in URL variables.

6.24.2 Response Codes and Exceptions

For HTTP response codes, see [REST_NetAPI_Common].

For Policy Exception and Service Exception fault codes applicable to Restful CAB APIs, see section 7.

6.24.3 GET

This resource is used to retrieve an individual External Request.

6.24.3.1 Example 1: Retrieve an individual External Request (Informative)

6.24.3.1.1 Request

```
GET /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.24.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Fri, 10 Feb 2012 12:51:59 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cab:externalRequest xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-
v1_0.xsd">
  <importProfile/>
  <nonCABSource>extSystem1</nonCABSource>
  <scheduledInterval>48</scheduledInterval>
  <username>maria</username>
  <password>maria123</password>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1234
  </resourceURL>
</cab:externalRequest>
```

6.24.4 PUT

This resource is used to update an individual External Request.

6.24.4.1 Example 1: Update an individual External Request (Informative)

6.24.4.1.1 Request

```
PUT /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1234 HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
```


Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<cab:externalRequest xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1
http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-v1_0.xsd">
  <importProfile/>
  <nonCABSource>extSystem1</nonCABSource>
  <scheduledInterval>48</scheduledInterval>
  <username>maria</username>
  <password>maria123</password>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1234
  </resourceURL>
</cab:externalRequest>
```

6.24.4.1.2 Response

HTTP/1.1 200 OK

Date: Fri, 10 Feb 2012 12:51:59 GMT

Content-Type: application/xml

Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<cab:externalRequest xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-
v1_0.xsd">
  <importProfile/>
  <nonCABSource>extSystem1</nonCABSource>
  <scheduledInterval>48</scheduledInterval>
  <username>maria</username>
  <password>maria123</password>
  <resourceURL>http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1234
  </resourceURL>
</cab:externalRequest>
```

6.24.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT, DELETE' field in the response as per section 14.7 of [RFC 2616].

6.24.6 DELETE

This resource is used to delete an individual External Request.

6.24.6.1 Example 1: Delete an individual External Request (Informative)

6.24.6.1.1 Request

```
DELETE /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1234 HTTP/1.1
Accept: application/xml
Host: example.com
```

6.24.6.1.2 Response

HTTP/1.1 204 No Content
Date: Fri, 10 Feb 2012 12:53:23 GMT

7. Fault definitions

7.1 Service Exceptions

For common Service Exceptions refer to [REST_NetAPI_Common].

In addition, all Service Exception codes specified in [Addressbook TS 1.0] sect. 7 SHALL apply herein.

Appendix A. Change History

(Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-REST_NetAPI_CAB-V1_0-20130702-A	02 Jul 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0208-INP_CAB_APIs_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for REST.CAB Server

Item	Function	Reference	Requirement
REST-CAB-SUPPORT-S-001-M	Support for the CAB RESTful API	5	
REST-CAB-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-CAB-SUPPORT-S-003-M	Support for the JSON request & response format	n/a	
REST-CAB-SUPPORT-S-004-O	Support for the application/x-www-form-urlencoded format	Appendix C	

B.1.1 SCR for REST.CAB.ContactCollection Server

Item	Function	Reference	Requirement
REST-AB-CONTACT-COL-S-001-M	Support contact collection with all of user's contacts	6.1	
REST-AB-CONTACT-COL-S-002-M	This operation retrieves all contacts in the contact collection - GET	6.1.3 of [Addressbook TS 1.0].	

B.1.2 SCR for REST.CAB.IndividualContact Server

Item	Function	Reference	Requirement
REST-AB-CONTACT-IND--S-001-M	Support individual contact in contact collection	6.2	
REST-AB-CONTACT-IND-S-002-M	This operation retrieves information of a contact in the contact collection - GET	6.17.3 of [Addressbook TS 1.0].	
REST-AB-CONTACT-IND-S-003-M	This operation adds or updates a contact in the contact collection - PUT	6.17.4 of [Addressbook TS 1.0].	
REST-AB-CONTACT-IND-S-004-M	This operation deletes a contact in the contact collection - DELETE	6.2.6 of [Addressbook TS 1.0].	

B.1.3 SCR for REST.CAB.AttributesForAContact Server

Item	Function	Reference	Requirement
REST-AB-ATTRIB-CONTACT-S-001-M	Supports retrieval of all attributes of a contact	6.3	
REST-AB-CONTACT-S-002-M	This operation retrieves all attributes for a contact - GET	6.18.3 of [Addressbook TS 1.0].	

B.1.4 SCR for REST.CAB.IndividualAttributeForAContact Server

Item	Function	Reference	Requirement
REST-AB-IND-ATTRIB-CONTACT-S-001-M	Support for management (create, update, retrieve and delete) of a contact's single attribute.	6.4	
REST-AB-IND-ATTRIB-CONTACT-S-002-M	This operation returns the value of the attribute for a contact or profile – GET	6.4.6 of [Addressbook TS 1.0].	
REST-AB-IND-ATTRIB-CONTACT-S-003-M	This operation creates or updates an attribute for a contact or profile – PUT	6.4.4 of [Addressbook TS 1.0].	
REST-AB-IND-ATTRIB-CONTACT-S-004-M	This operation removes an attribute for a contact or profile - DELETE	6.4.6 of [Addressbook TS 1.0].	

B.1.5 SCR for REST.CAB.Lists Server

Item	Function	Reference	Requirement
REST-AB- MEM-LS-S-001-M	Support for retrieving all lists belonging to a user.	6.5	
REST-AB- MEM-LS-S-002-M	This operation retrieves all lists the user has created - GET	6.5.3 of [Addressbook TS 1.0].	

B.1.6 SCR for REST.CAB.IndividualList Server

Item	Function	Reference	Requirement
REST-AB-IND-MEM-LS-001-M	Support for management (create, update, retrieve and delete) of an individual list.	6.6	
REST-AB-IND-MEM-LS-002-M	This operation retrieves the list - GET	6.6.3 of [Addressbook TS 1.0].	
REST-AB-IND-MEM-LS-003-M	This operation creates or updates the entire list - PUT	6.6.4 of [Addressbook TS 1.0].	
REST-AB-IND-MEM-LS-004-M	This operation removes the list including members - DELETE	6.6.6 of [Addressbook TS 1.0] with the exception of the list attributes.	

B.1.7 SCR for REST.CAB.MemberInAList Server

Item	Function	Reference	Requirement
REST-AB-MBR-MEM-LS-S-001-O	Supports retrieval of all members of a list belonging to a user	6.7	
REST-AB-MBR-MEM-LS-S-002-O	This operation retrieves the members in the list -	6.9.3 of [Addressbook	

Item	Function	Reference	Requirement
	GET	TS 1.0].	

B.1.8 SCR for REST.CAB.IndividualMemberInAList Server

Item	Function	Reference	Requirement
REST-AB-IND-MBR-MEM-LS-S-001-O	Support for management (create, update, retrieve and delete) of members on individual list.	6.8	
REST-AB-IND-MBR-MEM-LS-S-002-O	This operation retrieves a user from a list - GET	6.10.3 of [Addressbook TS 1.0].	
REST-AB-IND-MBR-MEM-LS-S-003-O	This operation creates and updates an entry in the list - PUT	6.10.4 of [Addressbook TS 1.0].	
REST-AB-IND-MBR-MEM-LS-S-004-O	This operation removes the member from the list - DELETE	6.17.6 of [Addressbook TS 1.0].	

B.1.9 SCR for REST.CAB.AttributesForAMemberInAList Server

Item	Function	Reference	Requirement
REST-AB-ATTRIB-MBR-MEM-LS-S-001-O	Supports retrieval of all attributes of a member of a list belonging to a user	6.9	
REST-AB-ATTRIB-MBR-MEM-LS-S-002-O	This operation returns all attributes for a member – GET	6.11.3 of [Addressbook TS 1.0].	

B.1.10 SCR for REST.CAB.IndividualAttributeForAMemberInAList Server

Item	Function	Reference	Requirement
REST-AB-IND-ATTRIB-MEM-LS-S-001-O	Support for management (create, update, retrieve and delete) of member attributes on individual list.	6.10	
REST-AB-IND-ATTRIB-MEM-LS-S-002-O	This operation returns the value of the attribute for a member – GET	6.12.3 of [Addressbook TS 1.0].	
REST-AB-IND-ATTRIB-MEM-LS-S-003-O	This operation creates or updates an attribute for a member – PUT	6.12.4 of [Addressbook TS 1.0].	
REST-AB-IND-ATTRIB-MEM-LS-S-004-O	This operation removes an attribute for a member - DELETE	6.12.6 of [Addressbook TS 1.0].	

B.1.11 SCR for REST.CAB.ListReferences Server

Item	Function	Reference	Requirement
REST-AB-MEM-LS-REF-S-001-O	Supports retrieval of a list of references for all nested lists.	6.11	
REST-AB-MEM-LS-REF-S-002-O	This operation returns a list of references to other	6.13.3 of [Addressbook	

Item	Function	Reference	Requirement
	lists. –GET	TS 1.0].	

B.1.12 SCR for REST.CAB.IndivListReference Server

Item	Function	Reference	Requirement
REST-AB-IND-MEM-LS-REF-S-001-O	Supports retrieval of a list of references for all nested lists.	6.12	
REST-AB-IND-MEM-LS-REF-S-002-O	This operation returns an individual list reference to another list –GET	6.14.3 of [Addressbook TS 1.0].	
REST-AB-IND-MEM-LS-REF-S-003-O	This operation creates or updates an individual reference to another list. –PUT	6.14.4 of [Addressbook TS 1.0].	
REST-AB-IND-MEM-LS-REF-S-004-O	This operation deletes an individual reference to another list. –DELETE	6.14.6 of [Addressbook TS 1.0].	

B.1.13 SCR for REST.CAB.List.Subscr Server

Item	Function	Reference	Requirement
REST-AB-SUBSCR-S-001-M	Support subscriptions for address book changes notifications for: CAB subscription-list changes, profile and AB changes.	6.13	
REST-AB-SUBSCR-S-002-M	This operation retrieves a list of subscriptions - GET	6.15.3 of [Addressbook TS 1.0].	
REST-AB-SUBSCR-S-003-M	This operation creates a subscription for list, profile or contact collection change notifications – POST (XML or JSON)	6.15.5 of [Addressbook TS 1.0].	
REST-AB-SUBSCR-S-004-O	This operation creates a subscription for list, profile or contact collection change notifications – POST (application/x-www-form-urlencoded)	C.1 of [Addressbook TS 1.0].	

B.1.14 SCR for REST.CAB.Individual.Subscr Server

Item	Function	Reference	Requirement
REST-AB-IND-SUBSCR-S-001-M	Support for managing an individual subscription for list or contact collection change notifications	6.14	
REST-AB-IND-SUBSCR-S-002-M	This operation retrieves an individual subscription for list, profile or contact	6.16.3 of [Addressbook TS 1.0].	

Item	Function	Reference	Requirement
	collection change notifications - GET		
REST-AB-IND-SUBSCR-S-003-M	This operation updates an individual subscription for list, profile or contact collection change notifications - PUT	6.16.4 of [Addressbook TS 1.0].	
REST-AB-IND-SUBSCR-S-004-M	This operation deletes an individual subscription for list, profile or contact collection change notifications - DELETE	6.16.6 of [Addressbook TS 1.0].	

B.1.15 SCR for REST.CAB.Notif Server

Item	Function	Reference	Requirement
REST-AB-NOTIF-S-001-M	Support for notifying application about list, profile or contact collection changes	6.15 of [Addressbook TS 1.0].	
REST-AB-NOTIF-S-002-M	This operation notifies the application about list, profile or contact collection changes - POST	6.17.5 of [Addressbook TS 1.0].	

B.1.16 SCR for REST.CAB.Authorization.RuleList Server

Item	Function	Reference	Requirement
REST-AB-AUTH-RULE-LS-S-001-O	Support authorization rules for sharing lists or contacts	6.20	
REST-AB-AUTH-RULE-LS-S-002-O	This operation retrieves all authorization rules owned by a given user - GET	6.23.3 of [Addressbook TS 1.0].	
REST-AB-AUTH-RULE-LS-S-003-O	This operation creates a new authorization rule – POST (XML or JSON)	6.23.5 of [Addressbook TS 1.0].	
REST-AB-AUTH-RULE-LS-S-004-O	This operation creates a new authorization rule – POST (application/x-www-form-urlencoded)	C.2 of [Addressbook TS 1.0].	

B.1.17 SCR for REST.CAB.Indiv.AuthorizationRule Server

Item	Function	Reference	Requirement
REST-AB-IND-AUTH-RULE-S-001-O	Support individual authorization rule	6.21	
REST-AB-IND-AUTH-RULE-S-002-O	This operation retrieves an individual authorization rule - GET	6.24.3 of [Addressbook TS 1.0].	
REST-AB-IND-AUTH-	This operation updates	6.24.4 of	

Item	Function	Reference	Requirement
RULE-S-003-O	an individual authorization rule - PUT	[Addressbook TS 1.0].	
REST-AB-IND-AUTH-RULE-S-004-O	This operation deletes an individual authorization rule - DELETE	6.24.6 of [Addressbook TS 1.0].	

B.1.18 SCR for REST.CAB.Indiv.AuthorizationRuleData Server

Item	Function	Reference	Requirement
REST-AB-IND-AUTH-RULE-DATA-S-001-O	Support individual authorization rule data	6.22	
REST-AB-IND-AUTH-RULE-DATA-S-002-O	This operation retrieves an individual authorization rule specific data - GET	6.25.3 of [Addressbook TS 1.0].	
REST-AB-IND-AUTH-RULE-DATA-S-003-O	This operation updates an individual authorization rule specific data - PUT	6.25.4 of [Addressbook TS 1.0].	
REST-AB-IND-AUTH-RULE-DATA-S-004-O	This operation deletes an individual authorization rule specific data - DELETE	6.25.6 of [Addressbook TS 1.0].	

B.1.19 SCR for REST.CAB.ProfileCollection Server

Item	Function	Reference	Requirement
REST-AB-PROFILE-COL-S-001-M	Support profile collection with all of user's profiles	6.16	
REST-AB-PROFILE-COL-S-002-M	This operation retrieves all profiles in the profile collection - GET	6.16.3	

B.1.20 SCR for REST.CAB.IndividualProfile Server

Item	Function	Reference	Requirement
REST-AB-PROFILE-IND-S-001-M	Support individual profile in profile collection	6.17	
REST-AB-PROFILE-IND-S-002-M	This operation retrieves information of a profile in the profile collection - GET	6.17.3	
REST-AB-PROFILE-IND-S-003-M	This operation adds or updates a profile in the profile collection - PUT	6.17.4	
REST-AB-PROFILE-IND-S-004-M	This operation deletes a profile in the profile collection - DELETE	6.17.6	

B.1.21 SCR for REST.CAB.AttributesForAProfile Server

Item	Function	Reference	Requirement
REST-AB-ATTRIBUTE-PROFILE-S-001-M	Supports retrieval of all attributes of a profile	6.18	

Item	Function	Reference	Requirement
REST-AB-PROFILE-S-002-M	This operation retrieves all attributes for a profile – GET	6.18.3	

B.1.22 SCR for REST.CAB.IndividualAttributeForAProfile Server

Item	Function	Reference	Requirement
REST-AB-IND-ATTRIB-PROFILE-S-001-M	Support for management (create, update, retrieve and delete) of a profile's single attribute.	6.19	
REST-AB-IND-ATTRIB-PROFILE-S-002-M	This operation returns the value of the attribute for a profile – GET	6.19.3	
REST-AB-IND-ATTRIB-PROFILE-S-003-M	This operation creates or updates an attribute for a profile – PUT	6.19.4	
REST-AB-IND-ATTRIB-PROFILE-S-004-M	This operation removes an attribute for a profile - DELETE	6.19.6	

B.1.23 SCR for REST.CAB.ExternalRequestCollection Server

Item	Function	Reference	Requirement
REST-CAB-EXT-REQ-COL-S-001-M	Support External Request collection with all requests issued by a user	6.23	
REST-CAB-EXT-REQ-COL-S-002-M	This operation retrieves a list of all External Requests - GET	6.23.3	
REST-CAB-EXT-REQ-COL-S-003-M	This operation creates a new External Request– POST (XML)	6.23.5	
REST-CAB-EXT-REQ-COL-S-004-O	This operation creates a new External Request– POST application/x-www-form-urlencoded)	C.1	

B.1.24 SCR for REST.CAB.IndividualExternalRequest Server

Item	Function	Reference	Requirement
REST-CAB-IND-EXT-REQ-S-001-M	Support individual External Request	6.24	
REST-CAB-IND-EXT-REQ-S-002-M	This operation retrieves information of an individual External Request - GET	6.24.3	
REST-CAB-IND-EXT-REQ-S-003-M	This operation updates an individual External	6.24.4	

Item	Function	Reference	Requirement
	Request - PUT		
REST-CAB-IND-EXT-REQ-S-004-M	This operation deletes an individual External Request - DELETE	6.24.6	

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

This section defines a format for the CAB REST API requests where the body of the request is encoded using the application/x-www-form-urlencoded MIME type.

Note: only the request body is encoded as application/x-www-form-urlencoded, the response is still encoded as XML or JSON depending on the preference of the client and the capabilities of the server.

The encoding is defined below for all CAB REST operations which are based on POST requests. In addition to the encodings defined in the [Addressbook TS 1.0] Appendix C, the following apply to CAB APIs:

- Create a new external request

C.1 Create a new External Request

This operation is used for creating a new External Request, see section 6.19.5.

The request parameters are as follows:

Name	Type/Values	Optional	Description
importNonCAB	xsd:boolean	Choice	Indicates the request is an import of a non-CAB Address Book contacts
importProfile	xsd:boolean	Choice	Indicates the request is an import of Profile data from 3 rd Party.
exportProfile	xsd:boolean	Choice	Indicates the request is an export of PCC data to a 3rd Party profile.
subscribeProfile	xsd:boolean	Choice	Indicates the request is subscription to CAB User's personal profile information in 3rd party system
nonCABSource	xsd:string	No	element indicating the source (e.g. domain name) of the external system
scheduledInterval	xsd:int	No	Number of hours between the successive exchanges with external system. It shall be set to '0' in case of subscription to CAB User's personal profile information in 3rd party system, or in case of a single request not intended for periodic repetition
username	xsd:string	Yes	User name that is used to access the external system. The use of username and password is one of the alternatives to access the external system.
password	xsd:string	Yes	Password that is used to access the external system. The use of username and password is one of the alternatives to access the external system.
token	xsd:string	Yes	A token that is used to access the external system. . The use of a token is one of the alternatives to access the external system.
filterSet	xsd:string	Yes	Filter to be applied to the imported/exported/subscribed Profile data. This element may be present in import-Profile, export-Profile and subscribe-Profile operations.

clientCorrelator	xsd:string	Yes	<p>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.</p> <p>This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations.</p> <p>In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.</p>
applicationTag	xsd:string	Yes	<p>A tag that the client MAY use to tag this particular resource on the server. In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.</p>

Table 7 External Request

C.1.1 Example 1: Create a new External Request (Informative)

C.1.1.1 Request

```
POST /exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

exportProfile=true&
nonCABSource=extSystem2&
scheduledInterval=0&
username=maria&
password=maria345&
```

C.1.1.2 Response

```
HTTP/1.1 201 Created
Date: Fri, 10 Feb 2012 21:33:52 GMT
Location: http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1236
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<cab:externalRequest xmlns:cab="urn:oma:xml:rest:netapi:cab:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oma:xml:rest:netapi:cab:1 http://www.openmobilealliance.org/tech/profiles/rest_netapi_convergedaddressbook-
v1_0.xsd">
  <exportProfile/>
  <nonCABSource>extSystem2</nonCABSource>
  <scheduledInterval>0</scheduledInterval>
  <username>maria</username>
```

```
<password>maria345</password>  
<resourceURL> http://example.com/exampleAPI/addressbook/v1/sip%3Amaria%40example.com/externalRequest/externalReq1236  
</resourceURL>  
</cab:externalRequest>
```

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 [RFC 4627].

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.

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 for CAB APIs (Informative)

The following table lists all CAB data structure elements that can be accessed individually as Light-weight Resources.

For each Light-weight Resource, the following information is provided: corresponding root element name, root element type and [ResourceRelPath] string.

All the Light-weight Resources specified in the [Addressbook TS 1.0] Appendix F apply.

Note: When appending [ResourceRelPath] string to its Heavy-weight Resource URL, all variables within curly brackets “{}” such as: serviceId, version, deviceId, watcherUserId, contactListId, and domainName have to be replaced by their real values.

In addition to the Light-weight Resources defined by the [Addressbook TS 1.0] Appendix F, the following new Light-weight Resources are specified by CAB APIs:

Type of Light-weight Resources (and references to data structures)	Element/attribute that can be accessed as Light-weight Resource	Root element name for the Light-weight Resource	Root element type for the Light-weight Resource	[ResourceRelPath] string that needs to be appended to the corresponding Heavy-weight Resource URL
Profile(5.2.2.15)	xcapNodeSelect or	profile	xsd:anyURI	profile/{xcapNodeSelector}

Table 8 Light-weight Resources for CAB APIs

Appendix G. Authorization aspects (Normative)

This appendix specifies how to use the RESTful Converged Address Book API in combination with some authorization frameworks.

G.1 Use with OMA Authorization Framework for Network APIs

The RESTful Converged Address Book API MAY support the authorization framework defined in [Autho4API_10], as specified in [Addressbook TS 1.0] Appendix G, with the clarifications and extensions added herein.

G.1.1 Scope values

G.1.1.1 Definitions

- The following clarifications and additions apply in addition to the [Addressbook TS 1.0] Appendix G.1.1:

Scope value	Description	For one-time access token
oma_rest_addressbook.profile	Provide access to all defined operations on profile.	No
oma_rest_addressbook.externalRequest	Provide access to all defined operations on lists.	No

Table 9 Scope values for RESTful Converged Address Book API

G.1.1.2 Downscoping

In the case where the client requests authorization for “oma_rest_addressbook.all_{apiVersion}” scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- “oma_rest_addressbook.contact”
- “oma_rest_addressbook.list”
- “oma_rest_addressbook.subscr”
- “oma_rest_addressbook.profile”
- “oma_rest_addressbook.externalRequest”

G.1.1.3 Mapping with resources and methods

The mapping of the scope values defined for the RESTful Converged Address Book API to the REST resources and methods of this API is based on the [Addressbook TS 1.0] Appendix G.1.1.3, with the following extensions and clarifications:

Note: In these tables, the root “oma_rest_addressbook.” of scope values is omitted for readability reasons.

Resource	URL Base URL: http://{serverRoot}/addressbook/ {apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of contacts	/ {userId}/contacts	6.1	all_{apiVersion} or contact	n/a	n/a	n/a
Individual contact	/ {userId}/contacts/{contactId}	6.2	all_{apiVersion} or contact	all_{apiVersion} or contact	n/a	all_{apiVersion} or contact
Attributes for a contact	/ {userId}/contacts/{contactId}/attribu tes	6.3	all_{apiVersion} or contact	n/a	n/a	n/a
Individual attribute for a contact	/ {userId}/contacts/{contactId}/attribu tes/[ResourceRelPath]	6.4	all_{apiVersion} or contact	all_{apiVersion} or contact	n/a	all_{apiVersion} or contact
Collection of shared contacts	/ {userId}/shared/{otherUserId}/cont acts	n/a	all_{apiVersion} or contact	n/a	n/a	n/a
Individual shared contact	/ {userId}/shared/{otherUserId}/cont acts/{sharedIdentity}	n/a	all_{apiVersion} or contact	n/a	n/a	n/a

Table 10 Required scope values for: managing contacts

Resource	URL Base URL: http://{serverRoot}/addressbook/ {apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of profiles	/ {userId}/profile	6.16	all_{apiVersion} or profile	n/a	n/a	n/a
Individual profile	/ {userId}/profile/{profileId}	6.17	all_{apiVersion} or profile	all_{apiVersion} or profile	n/a	all_{apiVersion} or profile
Attributes for a profile	/ {userId}/profile/{profileId}/attributes	n/a	all_{apiVersion} or profile	n/a	n/a	n/a
Individual attribute for a profile	/ {userId}/profile/{profileId}/attributes/ [ResourceRelPath]	n/a	all_{apiVersion} or profile	all_{apiVersion} or profile	n/a	all_{apiVersion} or profile

Table 11 Required scope values for: managing profile

Resource	URL Base URL: http://{serverRoot}/addressbook/ {apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Lists	/ {userId}/lists	6.5	all_{apiVersion} or list	n/a	n/a	n/a
Individual list	/ {userId}/lists/{listId}	6.6	all_{apiVersion} or list	all_{apiVersion} or list	n/a	all_{apiVersion} or list
Members in a list	/ {userId}/lists/{listId}/members	6.7	all_{apiVersion} or list	all_{apiVersion} or list	n/a	n/a
Individual member in a list	/ {userId}/lists/{listId}/members/{memberId}	6.8	all_{apiVersion} or list	all_{apiVersion} or list	n/a	all_{apiVersion} or list
Attributes for a member in a list	/ {userId}/lists/{listId}/members/{memberId}/attributes	6.9	all_{apiVersion} or list	n/a	n/a	n/a
Individual attribute for a member in a list	/ {userId}/lists/{listId}/members/{memberId}/attributes/[ResourceRelPath]	6.10	all_{apiVersion} or list	all_{apiVersion} or list	n/a	all_{apiVersion} or list
List references	/ {userId}/lists/{listId}/ listReferences	6.11	all_{apiVersion} or list	n/a	n/a	n/a
Individual list reference	/ {userId}/lists/{listId}/ listReferences/[ResourceRelPath]	6.12	all_{apiVersion} or list	all_{apiVersion} or list	n/a	all_{apiVersion} or list
Collection of shared lists	/ {userId}/shared/{otherUserId}/lists	n/a	all_{apiVersion} or list	n/a	n/a	n/a
Individual shared list	/ {userId}/shared/{otherUserId}/lists/{sharedListIdentity}	n/a	all_{apiVersion} or list	n/a	n/a	n/a
Individual member information from shared list	/ {userId}/shared/lists/{otherUserId}/{sharedListIdentity}/{memberId}	n/a	all_{apiVersion} or list	n/a	n/a	n/a

Table 12 Required scope values for: managing CAB subscription-list and entries in the list

Resource	URL Base URL: http://{serverRoot}/addressbook/ {apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Address Book changes subscriptions	/userId/subscriptions/abChanges	6.13	all_{apiVersion} or subscr	n/a	all_{apiVersion} or subscr	n/a
Individual address book changes subscription	/userId/ subscriptions/ abChanges/{subscriptionId}	6.14	all_{apiVersion} or subscr	all_{apiVersion} or subscr	n/a	all_{apiVersion} or subscr

Table 13 Required scope values for: managing subscriptions to notifications about contacts, profile and CAB subscription-list changes

Resource	URL Base URL: http://{serverRoot}/addressbook/{ apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Collection of external requests	/userId/externalRequest	6.19	all_{apiVersion} or externalRequest	n/a	all_{apiVersion} or externalRequest	n/a
Individual external request with a 3 rd party system	/userId/externalRequest/{reqId}	6.20	all_{apiVersion} or externalRequest	all_{apiVersion} or externalRequest	n/a	all_{apiVersion} or externalRequest

Table 14 Required scope values for: managing external requests

Resource	URL Base URL: http://{serverRoot}/addressbook/ {apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE

Authorization rules	/userId/authorizationRules	6.18	all_{apiVersion} or contact or profile	n/a	all_{apiVersion} or contact or list	n/a
Individual authorization rule	/userId/authorizationRules/{ruleId}	6.19	all_{apiVersion} or contact or profile	all_{apiVersion} or contact or profile	all_{apiVersion} or contact or profile	all_{apiVersion} or contact or profile
Individual authorization rule data	/userId/authorizationRules/{ruleId}/[ResourceRelPath]	6.20	all_{apiVersion} or contact or profile	all_{apiVersion} or contact or profile	n/a	all_{apiVersion} or contact or profile

Table 15 Required scope values for: managing authorization rules for shared contacts and profile

G.1.2 Use of 'acr:auth'

The use of 'acr:auth' is defined in the [Addressbook TS 1.0], Appendix G.1.2.

Appendix H. Exposure by 3rd Party Systems

A requestor (e.g. CAB User, an application) is able to request either by CAB native request or with help of the CAB APIs (i.e. POST on /{userId}/externalRequest) the following interactions with 3rd Party Systems:

- Import CAB User's personal profile information from 3rd party systems
- Export CAB User's PCC information to 3rd party systems
- Subscription to CAB User's personal profile information in 3rd party systems

The CAB Enabler needs to interact with the 3rd party system in order to fulfil these requests. This can be done by either:

- using any proprietary interface exposed by the 3rd party system.
- using CAB APIs provided they are exposed by the 3rd party system.

This section specifies the CAB APIs resources that can allow the fulfilment of these interactions with a 3rd party system in a standard way.

H.1 Import CAB User's Personal Profile Information from 3rd Party Systems

In order to implement this feature relying on CAB APIs, the 3rd party system SHALL support the following resources:

- Retrieval of the collection of profiles, as specified in section 6.16.3, i.e. GET /{userId}/profile
- Retrieval of a specific profile, as specified in section 6.17.3, i.e. GET /{userId}/profile/{profileId}

The procedure is as follows:

1. The CAB enabler retrieves the collection of profiles associated to a CAB User in a 3rd party system making use of the corresponding CAB API resource exposed by the 3rd party system.
2. The CAB enabler retrieves the desired profile(s) associated to a CAB User in a 3rd party system making use of the corresponding CAB API resource exposed by the 3rd party system. This step can be repeated periodically if requested by the CAB User.

H.2 Export CAB User's PCC Information to 3rd Party Systems

In order to implement this feature relying on CAB APIs, the 3rd party system SHALL support the following resources:

- Creation/update of a specific profile, as specified in section 6.17.4, with 'profileId' variable set to a value previously agreed with the 3rd party system, i.e. PUT /{userId}/profile/{profileId}

The procedure is as follows:

1. The CAB enabler creates/updates the profile making use of the corresponding CAB API resource exposed by the 3rd party system. This step can be repeated periodically if requested by the CAB User.

H.3 Subscription to CAB User's Personal Profile Information in 3rd Party Systems

In order to implement this feature relying on CAB APIs, the 3rd party system SHALL support the following resources:

- "Address Book changes subscriptions" specified in section 6.13. Just the subscription to profile information needs to be supported (i.e. AbChangesSubscription type containing the 'profile' element).
- "Individual list changes subscription" specified in section 6.14. Just the subscription to profile information needs to be supported (i.e. AbChangesSubscription type containing the 'profile' element).

- “Client resource for list changes notifications” specified in section 6.15 Just the notifications associated to a subscription to profile information need to be supported.

The procedure is as follows

1. The CAB enabler issues a subscription request to the personal profile information in a 3rd party system making use of the corresponding CAB API resource exposed by the 3rd party system.
2. The CAB Enabler acting as a client of CAB APIs will receive notification of any update in the personal profile information in a 3rd party system.
3. The CAB enabler may update/delete the subscription as needed making use of the corresponding CAB API resource exposed by the 3rd party system.