



CPM Message Store using RESTFul API

Candidate Version 1.0 – 26 Sep 2017

Open Mobile Alliance
OMA-TS-CPM_Message_Storage_Using_RESTFul_API-V1_0-
20170926-C

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

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

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

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

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

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

© 2017 Open Mobile Alliance All Rights Reserved.

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

Contents

1.	SCOPE	5
2.	REFERENCES	6
2.1	NORMATIVE REFERENCES	6
2.2	INFORMATIVE REFERENCES	7
3.	TERMINOLOGY AND CONVENTIONS	8
3.1	CONVENTIONS	8
3.2	DEFINITIONS	8
3.3	ABBREVIATIONS	9
4.	INTRODUCTION	10
4.1	CPM MESSAGE STORE USING RESTFUL API VERSION 1.0	10
5.	COMMON PROCEDURES	11
5.1	RESTFUL RESOURCES	11
5.2	AUTHENTICATION AND AUTHORIZATION	11
5.2.1	Authentication	11
5.2.2	Authorization	11
5.3	STORAGE FOLDER AND OBJECT RESOURCES	11
5.3.1	Message Object.....	12
5.3.2	File Transfer History Object	15
5.3.3	Session Info Object	15
5.3.4	Group State Object.....	15
5.3.5	Stand-alone Media Object.....	16
5.3.6	Conversation History Folder	16
5.3.7	Session History Folder	16
5.3.8	User Folder	16
5.3.9	Non-CPM Folder and Object resources	16
5.3.10	IMDN Object	17
5.4	IDENTIFICATION OF OBJECTS	18
5.5	SUBSCRIPTIONS AND NOTIFICATIONS	18
5.6	METADATA STRUCTURE	18
6.	PROCEDURES AT MESSAGE STORAGE CLIENT	20
6.1	GENERAL OPERATIONS	20
6.1.1	Authenticate Operations.....	20
6.1.2	Authorization Operations	20
6.1.3	Root Folder and Active Folder Operation.....	20
6.1.4	Access rights to Folders and sub Folders	20
6.1.5	Message and History Operations at the Client	20
6.2	SUBSCRIPTIONS AND NOTIFICATION OPERATIONS	20
6.2.1	Subscriptions	20
6.2.2	Notifications.....	21
6.3	SYNCHRONIZATION OPERATIONS	21
6.3.1	Managing local storage mirror (cache) at the client	21
6.3.2	Strict Synchronization.....	21
6.3.3	Types of Synchronization flows	21
6.4	CLIENT OPERATIONS AT THE OBJECT STORE	22
6.4.1	Object Resources available to Message Storage Clients.....	22
6.4.2	Folder Resources available to Clients	23
6.5	REFERENCE OPERATIONS	23
6.5.1	Generate Reference Operation	23
6.5.2	Fetch by Reference Operation.....	23
6.6	METADATA MANAGEMENT OPERATIONS	24
6.6.1	Metadata Update Operation	24
6.6.2	Bulk Update of Object Metadata	25

6.6.3 Metadata Fetch Operation 25

7. PROCEDURES AT MESSAGE STORAGE SERVER 26

7.1 GENERAL OPERATIONS..... 26

7.1.1 Authenticate Operations..... 26

7.1.2 Authorization Operations 26

7.1.3 Set Active Folder Operation 26

7.2 MESSAGE AND HISTORY SYNCHRONIZATION OPERATIONS..... 26

7.3 RESOURCES EXPOSED BY THE SERVER..... 26

7.3.1 Subscription and Notification Resources 26

7.3.2 Folder Resources at the Server..... 27

7.3.3 Object Resources at the Server 27

7.4 REFERENCE OPERATIONS..... 28

7.4.1 Generate Reference Operation 28

7.4.2 Fetch by Reference Operation..... 28

APPENDIX A. CHANGE HISTORY (INFORMATIVE) 29

A.1 APPROVED VERSION HISTORY 29

A.2 DRAFT/CANDIDATE VERSION 0.1 HISTORY 29

APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) 30

B.1 SCR FOR MESSAGE STORAGE CLIENT 30

B.2 SCR FOR MESSAGE STORAGE SERVER..... 30

APPENDIX C. EXAMPLE OF A MESSAGE OBJECT 34

APPENDIX D. EXAMPLE OF SESSION HISTORY FOLDER 38

APPENDIX E. EXAMPLE OF FILE TRANSFER HISTORY OBJECT..... 41

APPENDIX F. EXAMPLE OF SESSION INFO OBJECT 43

APPENDIX G. EXAMPLE OF GROUP STATE OBJECT..... 44

Tables

Table 1: Mapping of the REST field or attribute to the MIME Headers for the message object 15

Table 2: Call Log object attributes 17

1. Scope

This document provides a technical specification of how the CPM Message Storage functionality of the CPM Enabler [OMA-CPM-MSG] may be accessed using the RESTful interface defined in [OMA-REST-NMS]. The goal of this document is to present an alternative protocol binding of the CPM-MSG interface as defined in [OMA-CPM-AD] specifying procedures for Clients and Servers that communicate with the CPM Message Store.

As such, these technical specifications provide an alternative formal definition of the CPM-MSG interface that has been identified in [OMA-CPM-AD]. Also, these technical specifications formally define the expected behaviour of the Message Storage Client and Message Storage Server functional components that have been identified in [OMA-CPM-AD] when using the RESTful interface defined in [OMA-REST-NMS].

2. References

2.1 Normative References

[OMA-CPM-AD]	“Converged IP Messaging Architecture”, Open Mobile Alliance™, OMA-AD-CPM-V2_2, URL: http://www.openmobilealliance.org/
[OMA-CPM-MSG]	“CPM Message Store”, Open Mobile Alliance™, OMA-TS-CPM_Message_Store-V2_2, URL: http://www.openmobilealliance.org/
[OMA-CPM-SD]	“Converged IP Messaging System Description”, Open Mobile Alliance™, OMA-TS-CPM_System_Description-V2_1, URL: http://www.openmobilealliance.org/
[OMA-REST-NMS]	“RESTful Network API for Network Message Storage; ; Open Mobile Alliance™,” OMA-TS-REST_NetAPI_NMS-V1_0, URL: http://www.openmobilealliance.org
[OMA-CPM-IWF]	“CPM Interworking Function”, Open Mobile Alliance™, OMA-TS-CPM_Interworking_Function-V2_2, URL: http://www.openmobilealliance.org/
[OMA-CPM-RD]	“Converged IP Messaging Requirements”, Open Mobile Alliance™, OMA-RD-CPM-V2_2, URL: http://www.openmobilealliance.org/
[OMA_PUSH]	“OMA Push 2.3” Open Mobile Alliance™. OMA-ERP-Push-V2_3, URL: http://www.openmobilealliance.org/
[REST_NetAPI_ACR]	“RESTful Network API for Anonymous Customer Reference Management”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_ACR-V1_0, URL: http://www.openmobilealliance.org/
[REST_NetAPI_Common]	“Common definitions for RESTful Network APIs”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_Common-V1_0, URL: http://www.openmobilealliance.org/
[REST_NetAPI_NotificationChannel]	“RESTful Network API for Notification Channel”, Open Mobile Alliance™, OMA-TS-REST_NetAPI_NotificationChannel-V1_0, http://www.openmobilealliance.org/
[REST_SUP_NotificationChannel]	“XML schema for the RESTful Network API for Notification Channel”, Open Mobile Alliance™, OMA-SUP-XSD_rest_netapi_notificationchannel-V1_0, URL: http://www.openmobilealliance.org/
[OMA-SCRRULES]	“SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL: http://www.openmobilealliance.org/
[OMA-SEC-CF]	“Security Common Functions Requirements”, Open Mobile Alliance, OMA-RD-SEC_CF-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/
[IANA_Message_Headers]	“Message Headers”, IANA protocol registry, URL: http://www.iana.org/assignments/message-headers/message-headers.xhtml
[IETF_ACR_draft]	The acr URI for anonymous users”, S.Jakobsson, K.Smith, March 1, 2012, URL: http://tools.ietf.org/html/draft-uri-acr-extension-04 Note: The referenced IETF draft is a work in progress, subject to change without notice.
[RFC2045]	“Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”, N. Freed, N. Borenstein, November 1996, URL: http://tools.ietf.org/html/rfc2045
[RFC2046]	“Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types”, N. Freed, N. Borenstein, November 1996, URL: http://tools.ietf.org/html/rfc2046
[RFC2119]	“Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL: http://www.ietf.org/rfc/rfc2119
[RFC2387]	“The MIME Multipart/Related Content-type”, E. Levinson, August 1998, URL: http://www.ietf.org/rfc/rfc2387
[RFC2392]	“Content-ID and Message-ID Uniform Resource Locator”, E. Levinson, August 1998, URL: http://www.ietf.org/rfc/rfc2392
[RFC3458]	“Message Context for Internet Mail”, E. Burger, January 2003, URL: http://tools.ietf.org/rfc/rfc3458

- [RFC3862] “Common Presence and Instant Messaging (CPIM) : Message Format”, G. Klyne et al, August 2004, URL:<http://www.ietf.org/rfc/rfc3862>
- [RFC5246] “The TLS Protocol Version 1.2”, T. Dierks et al, January 1999, URL:<http://www.ietf.org/rfc/rfc5246>
- [RFC7519] “JSON Web Token (JWT)”, M. Jones, J. Bradley, N. Sakimura, May 2015. URL:<https://tools.ietf.org/html/rfc7519>
- [RFC7578] “Returning Values from Forms: multipart/form-data”, L. Masinter, August 1998, URL: <http://tools.ietf.org/html/rfc7578>
- [XMLSchema2] W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation 5 April 2012, URL: <http://www.w3.org/TR/xmlschema11-2/>

2.2 Informative References

- [OMADICT] “Dictionary for OMA Specifications”, Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2.9, 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 [OMADICT].

Box	The logical store belonging to a single subscriber, made up of one or more folders. See [OMA-REST-NMS] section 5.1.3.
Client-side Notification URL	See [REST_NetAPI_NotificationChannel].
CPM Address	See [OMA-CPM-RD].
CPM Chat Message	See [OMA-CPM-RD].
CPM Conversation	See [OMA-CPM-RD].
CPM Conversation History	See [OMA-CPM-RD].
CPM Conversation Identity	See [OMA-CPM-SD].
CPM File Transfer	See [OMA-CPM-RD].
CPM File Transfer History	See [OMA-CPM-RD].
CPM Message	See [OMA-CPM-RD].
CPM Participating Function	See [OMA-CPM-AD].
CPM Pre-defined Group	See [OMA-CPM-RD].
CPM Session	See [OMA-CPM-RD].
CPM Session History	See [OMA-CPM-RD].
CPM Session Invitation	See [OMA-CPM-RD].
CPM Standalone Message	See [OMA-CPM-RD].
CPM User	See [OMA-CPM-RD].
Media Object	See [OMA-CPM-AD].
Message Storage Client	See [OMA-CPM-AD].
Message Storage Server	See [OMA-CPM-AD].
Participant	See [OMADICT].
Principal	See [OMADICT].
Long Polling	See [REST_NetAPI_NotificationChannel].
NMS Object	A message object as defined in [OMA-CPM-MSG] section 5.3.1. with additional REST fields and attributes as defined in [OMA-REST-NMS]
Notification Server	See [REST_NetAPI_NotificationChannel].
Server-side Notification URL	See [REST_NetAPI_NotificationChannel].

3.3 Abbreviations

OMA	Open Mobile Alliance
ACL	See [OMADICT]
CPIM	Common Presence and Instant Messaging
CPM	See [OMADICT]
GSO	Group State Object
IMAP	See [OMADICT]
JWT	JSON Web Token. See [RFC7519]
MIME	See [OMADICT]
OMA	See [OMADICT]
PSK	Pre Shared Key
REST	Representational state transfer
SASL	Simple Authentication and Security Layer
SIO	session info object
TLS	See [OMADICT]
UID	Unique (Message) Identifier
URL	See [OMADICT]

4. Introduction

The CPM Message Storage functionality allows the storage of both legacy SMS/MMS messages as well as CPM Messages, CPM File Transfer Histories, CPM Session Histories, CPM Conversation Histories, and any potential Media objects either stand-alone or attached to CPM Messages and CPM Session Histories in a network-based storage on behalf of CPM Users, as described in [OMA-CPM-MSG].

The CPM Message Storage functionality authenticates and authorizes CPM Users to being able to retrieve, organize, set permissions, receive event notifications, synchronize with CPM Users device's local message storage and manage (e.g. copy, remove, move etc..) the storage objects that are stored on it. It also allows CPM Users to search the storage objects with key words.

This Specification provides users with an alternative RESTful interface and redefines the procedures and methods described in [OMA-CPM-MSG] which is optimized around the IMAP protocol. There are many benefits for selecting REST over IMAP (as an interface), but these are not to be discussed here, instead this specification defines how the RESTful API standards concepts [OMA-REST-NMS] are used for both Storage Server and Clients.

4.1 CPM Message Store using RESTful API Version 1.0

Version 1.0 offers binding of CPM Message Storage folders management and operations with RESTful APIs defined in [OMA-REST-NMS]. The intention of this binding is to provide an alternate mechanism to manage CPM Message Storage defines in [OMA-CPM-MSG].

[OMA-REST-NMS] provides the concepts of Object, Folder, Box and operation principle in the context of RESTful API as well as strict and simplified synchronization. Details of how relevant resources are defined and operated should refer to [OMA-REST-NMS]

5. Common Procedures

5.1 Restful Resources

5.2 Authentication and Authorization

5.2.1 Authentication

If a CPM client or server requires access to resources on the Network Message Storage Server using RESTful APIs, the CPM client or server MAY use OpenId Connect in conjunction with

1. OAuth 2.0 [RFC6749] to authenticate the client and end user in case of a client and the group in case of the server. The outcome of the authentication SHALL be an ID Token presented to the client, (the JWT) token as specified by [RFC 7519]. The ID Token SHALL contain a set of claims about the user, client or group that MUST be sufficient for the Network Message Storage Server to allow or deny access to a particular resource on the Network Message Storage Server. The CPM client or server SHALL present the ID Token on every request to the Network Message Store or
2. GBA [3GPP TS32.220] to authenticate the NMS Client, using the bootstrapping token in the OpenID Connect flow. If a GBA bootstrapping token was obtained by another client on the CPM User's device e.g. a CPM Client (for authentication with SIP/IP Core) then the NMS Client MAY reuse that token or
3. Digest or Basic authentication mechanisms managed by the CPM service. The Digest or Basic authentication mechanism and the required credentials MAY be pre-configured by the CPM system when the CPM user subscribes to the CPM service

5.2.2 Authorization

The Network Message Storage Server MUST use the ID Token presented in every OMA NMS API call to enforce access to stored resources. The ID Token presented by the CPM client or a server MUST be valid in time and MUST be signed by a valid trusted authority. Also, the ID Token MUST contain sufficient claims to allow the Network Message Storage Server to be able to grant access to a stored resource or a group of resources. In case the ID Token is invalid or the claims presented in the JWT token are insufficient to allow the CPM client access to the requested stored resource, the Network Message Storage Server MUST return an Error Response to the caller with sufficient information in order for the CPM client to determine why the request has been rejected.

5.3 Storage Folder and Object resources

A CPM Message Storage Server MAY contain the following items:

- message object,
- session history folder,
- file transfer history object,
- conversation history folder,
- stand-alone Media Object,
- user folder,
- session info object,
- group state object.

Other objects stored in the CPM Message Storage SHALL inherit the message object definition, such as:

- received disposition notifications are stored as message objects with no payload.

Note: on the originating side, the display notifications sent are stored as the \Seen flag.

In this section and the next, the storage objects are specified in terms of their names and identities, which can be used for various CPM Message Storage operations in accordance with NMS [OMA-REST-NMS].

The storage model of the CPM Message Storage SHALL include all the object and folder definitions from [OMA-CPM-MSG], and when accessed via NMS [OMA-REST-NMS] interface, it also:

1. SHALL provide the additional REST fields and attributes needed over the CPM-MSG with the NMS [OMA-REST-NMS] binding;
2. MAY store the additional REST fields and attributes for each object or folder.

This allows any CPM Client of the CPM User to reconstruct, based on objects and folders retrieved from the CPM Message Storage, the elements components of the CPM Conversation with the same level of detail as another CPM Client of the CPM User that has received or sent, the message via CPM SIP and MSRP interfaces. This is required in order to mirror the same CPM Conversation experience in a multi-device context, where the same message MAY be received, sent or obtained by the CPM Clients of the same CPM User via different protocols.

5.3.1 Message Object

A message object as defined in [OMA-CPM-MSG] is stored in the CPM Message Storage and can be retrieved via CPM-MSG interface with a NMS RESTful API protocol binding as defined in [OMA-REST-NMS].

[OMA-CPM-MSG] defines the storage model of the message object including:

- the message payload, and
- all metadata, SIP and MSRP session information that enables the retrieving entity (e.g. a CPM Client) to associate the message object with the CPM Conversation to which the message object belongs.

A CPM Client that retrieves a message object from the CPM Message Storage shall then be able to offer the same CPM Conversation continuation options to the CPM User as any other CPM Client that has received, or sent, that message via a CPM Message or a CPM Session method.

Additional REST fields and attributes as defined in [OMA-REST-NMS] SHALL be provided over the NMS compliant CPM-MSG interface, matching the overall NMS Object concept described in [OMA-REST-NMS].

5.3.1.1 Request URL

When accessing the message object via NMS RESTful API, the message object SHALL be identified by an object ID as defined in [OMA-REST-NMS]. The resource URL of the NMS object SHALL be constructed as defined in [OMA-REST-NMS] by appending the object ID to the current server root, API version, store name, box ID, and endpoint path. The {objectId} SHALL be a string which is unique in the context of the NMS Box.

An example of NMS object resource URL is:

<http://example.org/nms/v1/base/tel:+12125551212/objects/abcdef12345678>, where “abcdef12345678” is the object ID

5.3.1.2 Fields and attributes associated with a Message Object

When using the NMS RESTful API for the access to the message object, the following additional data SHALL be provided over the CPM-MSG interface with the NMS binding:

1. additional REST fields defined in [OMA-REST-NMS], Section 5.3.2.1 “Type: Object” SHALL be populated with the attribute names and definitions given in the appendix [OMA-REST-NMS], Appendix I “RCS Object Attributes Table”.
2. additional REST attributes defined in the [OMA-REST-NMS].

The additional REST fields and attributes for a message object SHALL be set as defined in the mapping table below:

[OMA-CPM-MSG] MIME headers	REST NMS field of Object	REST NMS attribute (in object.attributes)	Description of the value set for the REST field or attribute, or of its value mapping from a MIME header

		Direction	<p>Possible values:</p> <ul style="list-style-type: none"> - Mandatory. In this specification [OMA-REST-NMS] appendix I. defines the possible values:”In” and ”Out”. - ”In” is set for incoming traffic; - ”Out” is set for traffic outgoing from the CPM User.
		Is-CPM-Group	<p>Optional.</p> <p>In 1-1 CPM communication it SHALL NOT be present.</p> <p>It SHALL be mandatory when the object was exchanged during a CPM Group Session.</p>
		TextContent	<p>Defined in [OMA-REST-NMS] appendix, with the following clarifications that supercede [OMA-REST-NMS] :</p> <ul style="list-style-type: none"> - Contains the value of the MIME Message body header. For message objects delivered via SIP, the entire CPIM wrapped payload is stored. This includes the IMDN headers with the disposition notification requested. - Legacy SMS messages SHALL have a single text/plain payload part containing the content of the SMS encoded as UTF-8; see Section 5.4.2.1 - Legacy MMS messages SHALL be represented in standard MIME format, with each top-level MMS part represented as a payload part; see Section 5.4.2.2.

From		From	Set to the value of the MIME From header as specified in [OMA-CPM-MSG]. If the initiator withholds its details, (e.g. use of anonymous URI) then the “anonymous” String SHALL be provided.
To		To	Set to the value of the MIME To header as specified in [OMA-CPM-MSG].
		Cc	Used only in MMS objects, set to the value of the “Cc” header field, if present.
		Bcc	Used only in MMS objects, set to the value of the “Bcc” header field, if present.
Date As per [OMA-CPM-MSG]		Date As per [OMA-REST-NMS]	The REST Date field SHALL be set as defined in [OMA-REST-NMS]. Note: format differs from the MIME Date field.
Subject		Subject	As per MIME Subject field, as specified in [OMA-CPM-MSG].
P-Asserted-Service		P-Asserted-Service	As per P-Asserted-Service MIME field, as specified in [OMA-CPM-MSG].
Conversation-ID		Conversation-ID	As per Conversation-ID MIME field, as specified in [OMA-CPM-MSG].
Contribution-ID		Contribution-ID	As per Contribution-ID MIME field, as specified in [OMA-CPM-MSG].
Accept-Contact		Accept-Contact	As per Accept-Contact MIME field, as specified in [OMA-CPM-MSG].
InReplyTo-Contribution-ID		InReplyTo-Contribution-ID	As per InReplyTo-Contribution-ID MIME field, as specified in [OMA-CPM-MSG].
IMDN-Message-ID	correlationId		For CPM Messages the value is set to IMDN.Message-ID. For MMS is the MMS ID.

Message-Correlator	correlationTag		Value set to the SMS correlator computed value.
Message-Context		Message-Context	
Content-Type As per [OMA-CPM-MSG]		Content-Type [OMA-REST-NMS]	DEPENDING ON THE PAYLOAD STORING DISCUSSION ABOVE The Content-Type attribute SHALL be set as defined in [OMA-REST-NMS].
Message body			No mapping needed over NMS RESTful API.

Table 1: Mapping of the REST field or attribute to the MIME Headers for the message object

An example of the Message Object is provided in Appendix C

5.3.1.3 CPIM Encoding

A CPIM message [RFC3862] consists of

- MIME headers for the overall message, including Content-Type: Message/CPIM.
- CPIM message headers (message-metadata-headers).
- Encapsulated MIME message-body.

A message object which is a CPIM message is realized by an NMS object as follows:

- A single REST NMS attribute “CPIM” contains the entire CPIM message headers as a single string.
- The other fields and attributes are as defined above for all message objects, and are taken from the encapsulated MIME message body. In particular, the Content-Type attribute is the content type of the encapsulated MIME message body.
- The payload(s) of the message object are the body part(s) of the encapsulated MIME message body of the CPIM message.

An example of the a CPIM Message Object and how it is deposited and retrieved is provided in Appendix C.

5.3.2 File Transfer History Object

The File Transfer History object is a special type of a message object defined in [OMA-CPM-MSG]. It is realized by the message object defined in Section 5.3.1 “Message Object”.

The Message-Context attribute MUST be set to “file-message”.

5.3.3 Session Info Object

A session info object as defined in [OMA-CPM-MSG] is realized by the object concept of NMS as a message object according to Section 5.3.1 “Message Object”.

5.3.4 Group State Object

A Group State Object as defined in [OMA-CPM-MSG] is realized by the object concept of NMS as a message object according to Section 5.3.1 “Message Object”.

5.3.5 Stand-alone Media Object

The standalone Media Object is realized by the object concept of [OMA- REST-NMS], whereby the Media Object is included as a payload part of an object. Other than the formatting of the contents, the standalone Media Object is the same as the message object defined in section 5.3.1 “*Message Object*”, including the definition and naming specification of the message object. The CPM attributes and MIME headers defined for the message object are not applicable to the standalone Media Objects. Standalone Media Objects are further stored in the conversation folder.

5.3.6 Conversation History Folder

The conversation history folder defined in [OMA-CPM-MSG] is realized via the folder concept of [OMA-REST-NMS].

5.3.7 Session History Folder

The session history folder is viewed as a special kind of sub-folder according to [OMA-CPM-MSG] and is realized via the folder concept of [OMA-REST-NMS].

5.3.8 User Folder

The user folder is a Message Storage folder realized by the folder concept of [OMA- REST-NMS]. The user folder is identified by the name given to it. The CPM user folder aligns with the rules and procedures for names of the folder concept of [OMA- REST-NMS].

5.3.9 Non-CPM Folder and Object resources

5.3.9.1 Non-CPM Folders

Non-CPM folders are as defined in [OMA-CPM-MSG], and SHOULD be accessible by external entities to allow direct read/write of application-specific data using [OMA-REST-NMS].

5.3.9.2 Non-CPM Objects

Non-CPM objects are supported as defined in section 5.2.9.1. of [OMA-CPM-MSG]

5.3.9.3 Call Log Objects

A call log object contains the relevant information about a call that has been made.

The call log object is defined as in Table below.

A call log object MUST NOT have any body (payload parts).

Note: The headers and attributes values for group calls cases is FFS. Needs to be harmonized with the Multi-device Voice & Video TF.

Attribute name, Applicable to NMS API	Parameter status	Content
From	Mandatory	The address of the Originator of call as received in the P-Asserted-Id. If anonymous value is received (e.g. sending User blocked the address to be shown) then the String value: “unknown” SHALL be used. If the originator’s address is received but there’s a restriction indication, then the “restricted” String value SHALL be used.
To	Mandatory	Recipient of call, the address from the P-Called-Party-Id as specified in [3GPP TS24.229].
Cc	Optional	Must be present only in case of a group/conference calling. It contains the list of recipients for the group call.

Call-Type	Mandatory	Possible values are: “voice” and “video”.
Call-subtype	Optional	Note : It is FFS: Possible values: cellular, VoWiFi
Date		SHALL be set as defined in [OMA-REST-NMS].
Device-Name	Optional	Provided for log objects stored by devices only. Includes the name of the device that logged the call object, as per the name assigned by the CPM User (e.g. “Alice’s iPad”, “Cristina’s iPhone”).
Call-Disposition	Mandatory	The result of the call, i.e.: <ul style="list-style-type: none"> For incoming calls is one of “ANSWERED”, “LEFT_MESSAGE”, “HUNG_UP”, “REJECTED”, “BUSY” or “BLACKLISTED”. For outgoing calls is “OUTGOING ANSWERED”, “OUTGOING UNANSWERED”, “HUNG_UP_BEFORE_ANSWER”, “BUSY”. These values reflect the call result from all user devices, e.g. a call missed on one device and answered on another will be logged as “ANSWERED” and SHALL be stored by the device that answered the call.
Data-Usage	Optional	FFS: to determine if we keep it, depends on if the client can have the full data usage info at the end of a call
Call-Duration	Mandatory	The duration of the call in seconds.
Enriched-Content	Optional	It shall be a multi-part content-type including the enriched information shared. Note: This element can be defined later on.

Table 2: Call Log object attributes

5.3.10 IMDN Object

An IMDN object (i.e., a disposition notification) as defined in [RFC5438] is realized by the object concept of NMS as a message object with no payload according to Section 5.3.1 “Message Object”. The content of the IMDN is realized by NMS attributes. The Message-Context MUST be set to “imdn-message”. The following additional attributes MUST all be set:

Attribute name (on NMS API)	Content (see [RFC5438])
DispositionType	The type of the disposition, one of “delivery”, “processing” or “display”.
DispositionStatus	The status of the disposition, one of “delivered”, “failed”, “forbidden”, “error”, “processed”, “stored” or “displayed”. Legal values depend on the DispositionType: Delivery: delivered, failed, forbidden, error. Processing: processed, stored, forbidden, error. Display: displayed, forbidden, error.
DispositionOriginalMessageID	The ID of the message that this disposition refers to.

DispositionOriginalTo	The ID of the receiver client.
-----------------------	--------------------------------

5.4 Identification of Objects

The combination of a box ID and object ID MUST together permanently and persistently refer to one and only one stored object in a Message Storage Server. In particular, the attributes, correlationId, correlationTag, payload parts, and payload part info MUST never change according to [OMA-REST-NMS]. This requirement does not include folder location (path, parentFolder, parentFolderPath), nor does it include object attributes that can be set by an object POST request (e.g. flags).

Associated with any object or folder there is a lastModSeq value, which is intended to provide a means to determine whether any immediate properties of this object (e.g. flags, location) or folder (e.g. location) have changed since the previous time it checked this value.

5.5 Subscriptions and Notifications

[OMA-REST-NMS] defines a Strict Synchronization mechanism by which a client can discover changes to the storage, such as new messages delivered, messages moved between folders, flag changes, message deletions). Associated with this is a mechanism for subscription for change notifications, and a server-provided string called the restartToken which is used to identify a point in time at which the client last checked for changes.

The Message Storage Server and Message Storage Client SHALL implement Strict Synchronization as defined in [OMA-REST-NMS].

The Message Storage Server and Message Storage Client MAY implement Simplified Synchronization as defined in [OMA-REST-NMS].

5.6 Metadata Structure

CPM's message storage functionality supports a metadata model that consists of three distinct parts:

1. A set of metadata flags (i.e. system flags and keywords) that are associated with message objects, file transfer history objects and standalone Media Objects. These flags indicate additional state information about the stored object, and
2. A set of metadata annotations that can be associated with folder objects (i.e. mailboxes) and message objects (i.e. messages) stored in the Message Storage Server. These annotations provide system or user-defined information that the system or the user associates with these stored objects.
3. A set of metadata annotations that can be associated with the server (i.e. server annotations). These annotations provide system- or user-defined information that the system or the user associates with the server rather than individual objects.

The Message Storage Client and the Message Storage Server SHALL at least support metadata flags, folder metadata annotations and server annotations, and, in addition, MAY support message metadata annotations.

With respect to the metadata flags, the Message Storage Client and Message Storage Server SHALL support at least the following flags defined in [OMA-REST-NMS]:

- \Seen (message has been read),
- \Answered (message has been answered),
- \Flagged (message is "flagged" for urgent and/or special attention),
- \Deleted (message is "deleted" for removal by later EXPUNGE),
- \Draft (message has not completed composition (marked as a draft)),
- \Recent (message is "recently" arrived in this mailbox),
- \$Forwarded (message has been forwarded)
- \read-report-sent (A read receipt has been sent for this message)

- Archived (message is archived).

[OMA-REST-NMS] has no mechanism comparable to [RFC5464] for metadata annotations, and so server metadata and folder metadata annotations are not exposed in an [OMA-REST-NMS] based implementation of Message Storage Server.

[OMA-REST-NMS] has no mechanism comparable to the experimental protocol defined in [RFC5257] for message annotations, and so message metadata annotations are not exposed in an [OMA-REST-NMS] based implementation of Message Storage Server.

6. Procedures at Message Storage Client

The Message Storage Server enables Message Storage Clients to keep complete synchronization between themselves and the folders on the Message Storage Server. To do this, there are recommended approaches for how a Message Storage Client becomes aware of changes the contents of the Message Storage Server, and how synchronization takes place, the majority of this information is defined in [OMA-REST-NMS].

6.1 General Operations

Every client and server access to the Message Storage Server via the [OMA-REST-NMS] MUST be authenticated and authorized before allowing access to data stored on the server.

6.1.1 Authenticate Operations

See Section 5.2.1 “Authenticate operations”.

6.1.2 Authorization Operations

See Section 5.2.2 “Authorization operations”.

6.1.3 Root Folder and Active Folder Operation

A Message Storage Client SHALL be able to perform traversal of the storage hierarchical structure, after it discovers the root folder of the hierarchy (i.e. the starting point(s) for traversal). The search by folder attributes operation SHALL be used to identify root folder in the Message Storage Server. The root folder SHALL have an attribute named “Root” with the value “Yes” which designates such a starting point. The Message Storage Server MUST provide exactly one root folder per box. For further information, see [OMA-REST-NMS] section 5.4.7.

When the Message Storage Client retrieves the root folder it will discover its name. By default, the name of the root folder is an empty string unless specifically assigned to be some other name by the Message Storage Server.

Renaming a folder (e.g. the root folder) MAY be prohibited subject to service provider policy.

6.1.4 Access rights to Folders and sub Folders

Access rights to resources on the message store are described in Appendix G of [OMA-REST-NMS].

6.1.5 Message and History Operations at the Client

The Message Storage Client MUST keep a copy of all object and folder URLs, flags, and threaded/conversation view information. In particular, the Message Storage Client SHOULD follow the Object Search call:

`objects/operations/search` and SHOULD only be used in the cases where bulk download of content is required (for example, where the user clears the cache, is out of sync for an extended period, reinstalls the client or replaces the handset). In the Steady State case, the client SHOULD instead use the polling system for catching up the state of the box and keeping in sync.

The Message Storage Client MAY retain all header information for all messages to better user experience; subscribers will be able to access messages while offline. If it does not retain all the header information, the Message Storage Client SHOULD, at minimum, retain the headers for the most recent messages within each thread. The Message Storage Client SHALL be able to provide CPM User the summary information and the most recent messages for each thread without having to download information from Message Storage Server.

As for optimization, the client SHOULD cache some binary data, particularly for recent messages. By retaining all or nearly all recent binary data the client SHALL be able to assemble and render whole picture messages without retrieving from Message Storage Server.

6.2 Subscriptions and Notification Operations

6.2.1 Subscriptions

Message Storage Clients MUST subscribe for asynchronous notification of changes in the Message Storage Server.

See sections 5.1.4 and 5.4.1 of [OMA-REST-NMS] for details on using subscriptions and notifications and section 6.19 for details on creating and managing subscriptions within the object store.

6.2.2 Notifications

Notifications are generated as a result of active subscriptions described above in conjunction with changes to the mailbox. Message Storage Clients SHALL receive a change notification whenever the storage is considered to be changed (subject to any filtering rules specified when the subscription was created).

See section 5.1.4.1 of [OMA-REST-NMS] for details on using notifications to achieve and maintain synchronization.

6.2.2.1 Client usage of advanced Notification Filters

As described in section 5.4.3 of [OMA-REST-NMS] a filter is a mechanism which may be used by the Message Storage Client to indicate what kind of network storage changes it is interested to receive notifications about (e.g. only SMS messages or SMS messages from a particular contact/user ID).

If the Message Storage Client subscribes using filter criteria that the Message Storage Server does not support, the Message Storage Server MAY omit to indicate that it has not accepted the instruction and instead respond with a 200 OK (see RFC 2119).

6.3 Synchronization Operations

6.3.1 Managing local storage mirror (cache) at the client

See section 5.1.5 of [OMA-REST-NMS] for details on managing the local client message cache.

The NMS API offers two alternatives for synchronizing their local cache: strict synchronization and simplified synchronization. The CPM Message Storage Server MUST offer strict synchronization, which the Message Storage Client MUST utilize. The CPM Message Storage Client SHOULD NOT attempt to use simplified synchronization.

6.3.2 Strict Synchronization

As stated in section 6.3.1 “*Managing local storage mirror (cache) at the client*” above the Message Storage Client SHALL use strict synchronization and SHALL use subscriptions and notifications to keep itself informed of changes to the Message Storage Server.

Strict synchronization SHALL be used both online (where a client receives a stream of change notifications) and offline (where a client asks the server to be told of changes that have occurred since it was last connected). Strict synchronization is reliable as it can recover from notification loss, reordering, or duplication. The Message Storage Client SHALL receive, during synchronization, only the changes required and MUST use a per user folder token that is exchanged with the server.

6.3.3 Types of Synchronization flows

The Message Storage Client SHOULD implement the followings type of synchronization process

- **First Time Sync** - Running the Message Storage Client application for the first time having either had no previous contact with the Network Message Store or having had to clear and reinitialize the local message store.
- **Steady State Sync** – Ongoing sync with the Network Storage Server. This happens during normal operation; it also covers the case when connections time out, network signal is lost, the application or phone is restarted etc.

6.3.3.1 First Time Sync

First Time Sync SHOULD be used to perform the following to get the Message Storage Client’s local store and Message Storage Server in sync with one another. This is in two stages and result in both stores having the same set of messages.

- The Message Storage Client SHOULD download existing messages from Message Storage Server to the handset.
- The Message Storage Client SHOULD upload any handset messages that are not present in Message Storage Server.

Once First Time Sync has completed then the Message Storage Client SHOULD use Steady State Sync.

6.3.3.2 Steady State Sync

The purpose of the Steady State Sync process is to update the handset with changes made to the subscriber's mailbox. The Message Storage Client SHALL wait for notifications of incoming messages using a polling request, and SHALL process the notifications, upload and downloading messages and performing all correlation of Message Storage Server with the local handset store.

NOTE: The Message Storage Client MUST have an 'escape' method out of Steady state synchronization should an error occur that it cannot recover from. This 'escape' procedure may nominally be a retry of the affected objects followed the Message Storage Client SHOULD perform a first time synchronization.

6.4 Client Operations at the Object Store

See section 5.2 of [OMA-REST-NMS] for a summary of resources available to Message Storage Client.

See section 5.4.5 of [OMA-REST-NMS] for detailed description on operations on objects.

6.4.1 Object Resources available to Message Storage Clients

6.4.1.1 Object Search Operation

A Message Storage Client SHALL query the Message Storage Server to get object information using the resource. See section 6.7 of [OMA-REST-NMS] for details on search criterion. Additionally, a Message Storage Client MAY search using the location and Object identifier. See section 6.8 of [OMA-REST-NMS] for details on search using pathToID.

6.4.1.2 Object GET Operation

When a Message Storage Client needs to fetch an object, (e.g. a message object, a file transfer history object or standalone Media Object) from the active folder on the Message Storage Server, the Message Storage Client SHALL call the Message Storage Server using the RESTful Resource for managing the stored object. See sections of [OMA-REST-NMS] for further details on using GET on applicable resources.

6.4.1.3 Create Object

When a Message Storage Client needs to store an object (e.g. a message object, a file transfer history object or a standalone Media Object) into a folder on the Message Storage Server, the Message Storage Client SHALL send to the Message Storage a Restful Command to execute the Resource for creating a new object, supporting POST only, see section 6.1.5 of [OMA-REST-NMS] for details creating a new object.

6.4.1.4 Object Move Operation

If a Message Storage Client moves a referenced source object(s) and/or folder(s) (including recursive folders' content) to a designated target folder, the resource "movetoFolder" SHALL be used and is shown in section 6.18 of [OMA-REST-NMS].

Same restriction applies for Move operation, as specified in section 6.3.5. "*Object Remove Operation*" of the [OMA-CPM-MSG] regarding the move of CPM specific objects containing CPM Session metadata, such as session info object, or the latest group state object or conference info object for a CPM Group Session: these objects SHALL NOT be moved to another folder or be removed from its original folder until all the associated messages, their associated disposition notifications and media objects from that CPM Session folder have been moved or removed from the original CPM Session folder.

6.4.1.5 Object DELETE Operation

When a Message Storage Client needs to remove a stored message object, a flag on an object, a file transfer history object, a Group State Object or a stored standalone Media Object, the Message Storage Client SHALL send to the Message Storage Server a DELETE command. Deletes are described in detail in section 5.1.7 of [OMA-REST-NMS]. See also section 6.7 of [OMA-REST-NMS].

Same restriction for special CPM objects containing metadata applies for DELETE operation, as specified in section 6.4.1.4. "*Object Move Operation*".

6.4.1.6 Bulk Object creation

A Message Storage Client MAY create multiple objects in a single POST. See section 6.1.9 of [OMA-REST-NMS] for details on bulk creation.

6.4.1.7 Bulk DELETE objects

A Message Storage Client MAY delete more than one object in a single DELETE. The resource used is described section 5.4.2 of [OMA-REST-NMS] and examples are shown in section 6.11.5 of [OMA-REST-NMS].

In a bulk DELETE operation, the same restriction for special CPM objects containing metadata applies for DELETE operation, as specified in section 6.4.1.5. “*Object DELETE Operation*”.

6.4.2 Folder Resources available to Clients

6.4.2.1 Folder Search Operation

The Message Storage Client SHALL use the resource for retrieving information about a set of selected folders according to section 6.15 of [OMA-REST-NMS].

6.4.2.2 Folder Create Operation

The Message Storage Client MAY create a folder on the server. See section 7.3.2.2 as it is more common that the folders shall be created by the Message Storage Server through the process of depositing of messages.

6.4.2.3 List Folder Operation

A Message Storage Client SHALL discover the folder hierarchy according to the process described in section of 6.16 [OMA-REST-NMS].

6.4.2.4 Folder Move Operation

A Message Storage Client MAY move a folder and its contents and sub folders as per section 6.18.5 of [OMA-REST-NMS].

6.4.2.5 Folder DELETE Operation

The DELETE operation is allowed on folders, and is described in section 6.13.6 of [OMA-REST-NMS]. All the contained folders and objects (including their payload) in the targeted folder SHALL be deleted as well.

The Message Storage Server SHALL respond to a DELETE request with an HTTP 204 No Content response.

6.5 Reference Operations

6.5.1 Generate Reference Operation

When a unique reference for a message object is available, it MUST appear in the correlationId field as described in [OMA-REST-NMS].

When a unique reference is not available, the client MUST use the correlationTag field as described in [OMA-REST-NMS], if present, to correlate messages received from the Message Storage Server with those received via the RAN.

6.5.1.1 CorrelationTag generation

Within CPM, when the correlationTag is provided the Message Storage Client and Message Storage Server MUST calculate the correlationTag field using the algorithm described in section 3.2.4.7.3 of [RCS-RCC.07].

6.5.2 Fetch by Reference Operation

The Message Storage Client SHALL query (search) a message object using the /objects/operations/search resource and apply a SearchCriterion that may reference any of the message fields and attributes, including but not limited to

- correlationId
- correlationTag
- Contribution-ID

- Conversation-ID
- InReplyTo-Contribution-ID

See section 5.3.2.19 of [OMA-REST-NMS] for details on search criterion.

6.6 Metadata Management Operations

6.6.1 Metadata Update Operation

The following operations (and only these operations) SHALL be considered to cause a change in an object in the storage:

- creation (whether direct or as part of a recursive operation)
- user-initiated deletion (whether direct or as part of a recursive operation)
- expiry
- “parentFolder” change
- flag change

The following operations (and only these operations) SHALL be considered to cause a change in a folder in the storage:

- creation (whether direct or as part of a recursive operation)
- user-initiated deletion (whether direct or as part of a recursive operation)
- expiry
- “parentFolder” change
- change of any Light-weight Resource within a folder, as described in Section 6.14.1.1 “Light-weight relative resource paths”, e.g. changing its “folderName”.

This means that an object or folder is considered to have changed only when the change affects the object or folder itself. That is, changes to the object’s or folder’s derived properties (such as its children or the name or location of one of its parent folders) is not considered a change to the object or folder itself. In particular:

- Changing the “folderName” of a folder SHALL be considered a change to that folder, but SHALL NOT be considered a change to any subfolders or objects within it (even though their “path” and “parentFolderPath” values are now different).
- Adding or deleting an object within a folder SHALL be considered a change to that object, but SHALL NOT be considered a change to the folder itself (even though its “objects” value is now different).
- Adding or deleting a subfolder within a parent folder SHALL be considered a change to the subfolder, but SHALL NOT be considered a change to the parent folder itself (According to [OMA-REST-NMS], only certain changes to objects and folders will be notified to the client).
- Moving a folder to a new location SHALL be considered a change to the moved folder (since its parentFolder changes), but SHALL NOT be considered a change to any subfolders or objects within it (According to [OMA-REST-NMS], only certain changes to objects and folders will be notified to the client), nor the target (folder) (even though its “subFolders” value is now different).

In each of these cases, the Message Storage Client SHALL infer any derived changes it needs to capture the overall change to the storage. For example, when a parent “folderName” changes, even though the storage does not report a change against the subfolders’ path values, the Message Storage Client SHALL derive this change locally based on the notification of the parent folderName change.

6.6.1.1 Updating an Object’s Flags

The Message Storage Server SHALL support a resource for updating an Object’s flags. This is the most common use of updating an Object’s metadata and SHALL be accomplished using the PUT method

://{nmsHost}/nms/v1/base/{boxId}/objects/{objectId}/flags

Note that Message Storage Server MAY support the flag the Message Storage Client is updating, but if it does not the result code (response) MAY still be the same (200ok). Message Storage Client SHOULD check if flags are supported by Message Storage Server before requesting UPDATE operation.

6.6.2 Bulk Update of Object Metadata

A Message Storage Client MAY wish to change the status of all its messages in Message Storage Server to “read”. Such an action would generate many updates to the server to set the individual object’s flags to accordingly. The Message Storage Server SHALL support a Bulk UPDATE resource as follows:

```
//{{serverRoot}}/nms/{apiVersion}/{storeName}/{boxId}/objects/operations  
/bulkUpdate
```

This resource is used for updating multiple objects using a single request.

NOTE: if the update of all the objects failed the Message Storage Server SHALL return an appropriate HTTP 4xx or 5xx response code.

Otherwise the Message Storage Server SHALL return an appropriate HTTP 2xx response code, even if the update of some (but not all) objects failed.

For HTTP response codes, see [REST_NetAPI_Common].

The response body SHALL include a list of success or failure status for each object in the request list (if objects are specified in the request) or for each matching object (if selectionCriteria are specified in the request) respectively.

The maximum size of bulk update request MAY be limited subject to server provider policy, e.g. by number of objects, object size, total request size. For this reason, the Message Storage Client SHOULD NOT make unreasonably large bulk update requests.

6.6.3 Metadata Fetch Operation

The Message Storage Client SHALL receive the metadata of objects within the store using the following methods:

1. Notifications sent from the Message Storage Server SHALL contain the metadata of the objects that have changed in the store
2. Object search: the Message Storage Client SHALL search the server for objects and the result shall be a list of objects and their associated metadata.

7. Procedures at Message Storage Server

The Message Storage Server is a functional component of the CPM enabler, which allows authorized and/or authenticated principals (such as Message Storage Clients or CPM Participating Functions) to access a resource in the Message Storage Server.

The Message Storage Server SHALL act as RESTful Resource server as defined in [OMA-REST-NMS], so the Message Store Server SHALL advertise the resources necessary to the supported to the Message Store Client.

7.1 General Operations

7.1.1 Authenticate Operations

See Section 5.2.1 “Authenticate operations”.

7.1.2 Authorization Operations

See Section 5.2.2 “Authorization operations”.

7.1.3 Set Active Folder Operation

The Message Storage Server SHALL provide a root folder at the location as described in section 5.4.7 [OMA-REST-NMS] i.e.

http://{serverRoot}/nms/{apiVersion}/{storeName}/{boxId}/folders

There SHALL only be one root folder which is identified by having an attribute named “Root” with the value “Yes”.

The Message Storage Server SHALL support Message Storage Client access to the folder structure as described in section 6.3.1 “Object Store Operation” of [OMA-CPM-MSG]

7.2 Message and History Synchronization Operations

The Message Storage Server SHALL support synchronization as per section 5.4.2 of [OMA-REST-NMS]. The Message Storage Server MUST allow Message Storage Client to exercise their synchronization process as described in section 5.1.5 of [OMA-REST-NMS] and the Message Storage Server MUST support strict synchronization as described in section 5.1.5.1 of [OMA-REST-NMS].

The Message Storage Server SHALL support Message Storage Client’s requirements for first time synchronization and steady state synchronization as per section 6.3.3. In addition, the Message Storage Server SHALL perform a ‘box reset’ as described in section 5.1.4.2.1 of [OMA-REST-NMS].

7.3 Resources exposed by the Server

Please see [OMA-REST-NMS] for a full list of supported resources for Subscriptions, Notifications, Objects and Folders.

7.3.1 Subscription and Notification Resources

The Message Storage Server SHALL provide RESTful resources to allow Message Storage Client to subscribe for Notifications about changes to contents of the user’s BOX as described in section 5.1.4.2 [OMA-REST-NMS]. The Message Storage Server SHALL be responsible for managing notifications according to the tracked changes and SHALL provide support for “lastModSeq” as described in section 5.1.4.4 of [OMA-REST-NMS] and for “restartToken” as described in section 5.1.4.3 of [OMA-REST-NMS].

7.3.1.1 Subscriptions

The Message Storage Server SHALL support the mechanisms to allow a Message Storage Client to subscribe to box events via a RESTful subscription as per section 5.1.4 of [OMA-REST-NMS].

7.3.1.2 Notifications

The Message Storage Server SHALL send notifications as per section 5.1.4.1 of [OMA-REST-NMS].

7.3.2 Folder Resources at the Server

The Message Storage Server SHALL provide RESTful resources that allow messages to be stored and searched for under a hierarchical folder structure as per section 5.1.2 of [OMA-REST-NMS].

7.3.2.1 Folder Search Operation

The Message Storage Server SHALL support search of a folder as per section 6.15 of [OMA-REST-NMS]

7.3.2.2 Create Folder Operation

The Message Storage Server SHALL support folder creation as per section 6.12.5 of [OMA-REST-NMS].

7.3.2.3 List Folders Operation

The Message Storage Server SHALL support folder discovery as per section 5.1.6 of [OMA-REST-NMS].

7.3.2.4 Folder Move Operation

The Message Storage Server SHALL support the RESTful resource described in section 6.18 of [OMA-REST-NMS] “moveToFolder” which allows a Message Storage Client to move the folder and all its contents. Some folders SHALL NOT be allowed to be moved, subject to service provider policies. If a folder cannot be moved per service provider policy, Message Storage Server SHALL reply with the appropriate error as per 6.18.5.3 of [OMA-REST-NMS].

7.3.2.5 Folder Copy Operation

The Message Storage Server SHALL support the RESTful resource described in section 6.17.5.2 of [OMA-REST-NMS] “copyToFolder” which allows a Message Storage Client to copy a folder and all its contents to a new folder.

7.3.2.6 Folder Remove Operation

The Message Storage Server SHALL support folder deletion as per section 6.13.6 of [OMA-REST-NMS]. The Message Storage Server SHALL delete a folder due to expiry as per section 5.1.7 of [OMA-REST-NMS].

7.3.3 Object Resources at the Server

7.3.3.1 Object creation

The Message Storage Server SHALL support Object creation as per section 6.1 of [OMA-REST-NMS].

If neither parentFolder nor parentFolderPath are included by the Message Storage Client in an object creation request (as described in section 5.3.2.1 “Type: Object” of [OMA-REST-NMS]), the Message Storage Server SHALL choose a parent folder according to the folder structure described in section 6.3.1 “Object Store Operation” of [RCS-RCC.09].

7.3.3.2 Object Bulk creation

The Message Storage Server SHALL support bulk Object creation as per section 6.9 of [OMA-REST-NMS].

7.3.3.3 Object Search operations

The Message Storage Server SHALL support Object search using a set of search criterion as per section 6.7 of [OMA-REST-NMS]. Additionally, the Message Storage Server SHALL support Object search using its location and Object ID as per section 6.8 of [OMA-REST-NMS].

7.3.3.4 Object Copy Operation

The Message Storage Server SHALL support bulk Object creation as per section 6.17.5.1 of [OMA-REST-NMS].

7.3.3.5 Object Remove Operation

The Message Storage Server SHALL support Object delete as per section 6.2.6 of [OMA-REST-NMS]. The Message Storage Server SHALL delete an object due to expiry as per section 5.1.7 of [OMA-REST-NMS].

The Message Storage Client SHALL NOT remove a stored session info object unless it is done in the context of the entire session history folder itself being removed or moved. All associated IMDNs SHALL also be deleted along with the Object.

7.3.3.6 Object Bulk Delete Operation

The Message Storage Server SHALL support Object bulk delete as per section 6.2.11 of [OMA-REST-NMS]. All associated IMDNs related to the deleted objects SHALL also be deleted.

7.3.3.7 Object Bulk Update Operation

The Message Storage Server SHALL support Object bulk update as per section 6.2.10 of [OMA-REST-NMS].

7.4 Reference Operations

7.4.1 Generate Reference Operation

When a unique reference for a message object is available, it MUST appear in the correlationId field as described in [OMA-REST-NMS].

When a unique reference is not available, and in cases where the Message Storage Client may not receive the unique reference over the RAN (e.g. CPM messages interworked to SMS), the Message Storage Server MUST provide the correlationTag field as described in [OMA-REST-NMS], to enable the Message Storage Client to correlate messages received from the Message Storage SServer with those received via the RAN.

7.4.1.1 Correlation Tag generation

Within CPM, when the correlationTag is provided the Message Storage Server MUST calculate the correlationTag field using the algorithm described in section 3.2.4.7.3 of [RCS-RCC.07].

7.4.2 Fetch by Reference Operation

The Message Storage Server SHALL support the Message Storage Client's to query (search) for a message object using the /objects/operations/search resource and applying a Search Criterion that may reference any of the message fields and attributes, including but not limited to

- correlationId
- correlationTag
- Contribution-ID
- Conversation-ID
- InReplyTo-Contribution-ID

See section 5.3.2.19 of [OMA-REST-NMS] for details on search criterion.

Appendix A. Change History

(Informative)

A.1 Approved Version History

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

A.2 Draft/Candidate Version 0.1 History

Document Identifier	Date	Sections	Description	
Draft Versions OMA-TS-CPM-Message_- Storage_Using_RESTful_API-V1_0	22 Apr 2016	All	New Document	
	09 Jul 2016	All	Official draft version created and uploaded after CPM 2.2 approved; same as the April 22 approved version with title alignment on first page	
	31 Dec 2016	All	OMA-COM-CPM-2016-0065-CR_MSAPI_Sec_3_Updates OMA-COM-CPM-2016-0079R02- CR_Sec_5_3_2_File_Transfer_History_Object OMA-COM-CPM-2016-0080R01- CR_Sec_5_3_3_Session_Info_Object OMA-COM-CPM-2016-0081R01- CR_Sec_5_3_4_Group_State_Object OMA-COM-CPM-2016-0082R02- CR_Sec_5_3_5_File_Transfer_Media_Object OMA-COM-CPM-2016-0083R03- CR_Sec_5_3_6_Standalone_Media_Object OMA-COM-CPM-2016-0084R01- CR_Sec_5_3_7_Session_History_Folder OMA-COM-CPM-2016-0085R01-CR_Sec_5_3_8_User_Folder OMA-COM-CPM-2016-0090R01-CR_Sec_2_Reference OMA-COM-CPM-2016-0093-CR_Sec_5_4_Identification_of_Objects OMA-COM-CPM-2016-0094R02- CR_Sec_5_5_subscriptions_n_notifications OMA-COM-CPM-2016-0095R03-CR_Sec_5_6_Metadata OMA-COM-CPM-2016-0096R01-CR_Sec_6_1_general_operations OMA-COM-CPM-2016-0097R01-CR_Sec_6_2_and_6_3 OMA-COM-CPM-2016-0098R01-CR_Sec_6_4_Folder_Operations OMA-COM-CPM-2016-0099R01-CR_Sec_6_5_Reference_Operations	
	20 Jan 2017	5.3.9	OMA-COM-CPM-2016-0091R01-CR_Sec_5_2_Auth_Auth OMA-COM-CPM-2016-0092R03-CR_Sec_5_3_9_Non_CPM_F older_Object OMA-COM-CPM-2016-0111-CR_Appendix_B_Content OMA-COM-CPM-2016-0112-CR_Appendix_C_Content OMA-COM-CPM-2016-0113-CR_Appendix_D_Content OMA-COM-CPM-2016-0115-CR_Appendix_F_Content	
	01 Feb 2017	5.3 Appendix C	OMA-COM-CPM-2016-0078R02 CR_Sec_5_3_1_Message_Object OMA-COM-CPM-2016-0110R01-CR_Sec_7_Content	
	20 Feb 2017	All	OMA-COM-CPM-2017-0029-CR_editorial_corrections	
	09 May 2017	All	Apply all CONR comments in OMA-CONRR-CPM-V2_2-20170505-D for final version before approval	
	10 May 2017	5.2.1	Correct "Direct authentication" to "Basic authentication" missed from the email correction	
	15 May 2017	2.1, 4.1, 7.1.3	Correct the OMA-REST-NMS references	
	16 May 2017	2.1	Correct the CPM IWF version used	
	05 Sep 2017	7.3.3.5, 7.3.3.6	OMA-COM-CPM-2017-0094-CR_Clarify_Msg_Object_Removal	
	Candidate Version OMA-TS-CPM-Message_- Storage_Using_RESTful_API-V1_0	26 Sep 2017	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2017-0040-INP_CPM- V2_2_ERP_for_1st_Candidate_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for Message Storage Client

B.2 SCR for Message Storage Server

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

The CPM Message Storage Server SCRs are inherited from the NMS SCRs in [OMA-REST-NMS]. The following table describes the SCRs for the CPM Message Storage Server when using the NMS RESTful API.

Item	Function	Status	Requirement
REST-NMS-SUPPORT-S-001-M	Support for the RESTful NMS API	Mandatory	
REST- NMS-SUPPORT-S-002-M	Support for the XML request & response format	Optional	
REST- NMS-SUPPORT-S-003-M	Support for the JSON request & response format	Mandatory	
REST-NMS-OBJECTS-S-001-M	Support for object creation (object upload into NMS)	Mandatory	
REST-NMS-OBJECTS-S-002-M	Upload (create) a new object into an identified folder of NMS – POST	Mandatory	
REST-NMS-AOBJECT-S-001-M	Support for managing individual stored object	Mandatory	
REST-NMS-AOBJECT-S-002-M	Retrieve metadata of an object – GET	Mandatory	
REST-NMS-AOBJECT-S-003-M	Delete an object – DELETE	Mandatory	
REST-NMS-AOBJECT-FLAGS-S-001-M	Support for managing flags associated with an object	Mandatory	
REST-NMS-AOBJECT-FLAGS-S-002-M	Retrieve flag list of an object – GET	Mandatory	
REST-NMS-AOBJECT-FLAGS-S-003-O	Add a new flag to the flag list of an object – PUT	Optional	
REST-NMS-AOBJECT-FLAGS-S-004-M	Update the entire flag list of an object – PUT	Mandatory	
REST-NMS-AOBJECT-INDFLAG-S-001-M	Support for managing an individual flag associated with an object	Mandatory	
REST-NMS-AOBJECT-INDFLAG-S-002-M	Test for existence of a flag in a given object – GET	Mandatory	
REST-NMS-AOBJECT-INDFLAG-S-003-M	Add a new flag to the flag list of an object – PUT	Mandatory	

Item	Function	Status	Requirement
REST-NMS-AOBJECT-INDFLAG-S-004-M	Remove a flag from flag list of an object – DELETE	Mandatory	
REST-NMS-AOBJECT-PAYLOAD-S-001-M	Support for retrieving object's entire payload	Optional	
REST-NMS-AOBJECT-PAYLOAD-S-002-M	Retrieve the entire payload of a given object – GET	Optional	
REST-NMS-AOBJECT-PAYLOADPART-S-001-M	Support for retrieving an individual payload part	Mandatory	
REST-NMS-AOBJECT-PAYLOADPART-S-002-M	Retrieve an individual payload part of a given object – GET	Mandatory	
REST-NMS-OBJECTS-SEARCH-S-001-M	Support for searching for objects meeting certain criteria	Mandatory	
REST-NMS-OBJECTS-SEARCH-S-002-M	Retrieve information about objects meeting certain criteria – POST	Mandatory	
REST-NMS-OBJECTS-PATHTOID-S-001-M	Support for looking up object(s) resource URL(s) using its (their) location/path	Mandatory	
REST-NMS-OBJECTS-PATHTOID-S-002-M	Retrieve an object's Id (resource URL) using its location/path – GET	Mandatory	
REST-NMS-OBJECTS-PATHTOID-S-003-M	Retrieve a list of object Ids (resource URLs) using their location/path – POST	Mandatory	
REST-NMS-OBJECTS-bulkCreation-S-001-O	Support for bulk upload of objects	Mandatory	
REST-NMS-FOLDERS-S-001-M	Support for folder creation	Mandatory	
REST-NMS-FOLDERS-S-002-M	Create a new folder under an identified parent folder in NMS – POST	Mandatory	
REST-NMS-AFOLDERS-S-001-M	Support for folder creation	Mandatory	
REST-NMS-AFOLDERS-S-002-M	Retrieve information about a folder – GET	Mandatory	
REST-NMS-AFOLDERS-S-003-M	Delete a folder – DELETE	Mandatory	
REST-NMS-FOLDERNAME-S-001-M	Support for managing individual folder data	Mandatory	
REST-NMS-FOLDERNAME-S-002-M	Retrieve folder name – GET	Mandatory	
REST-NMS-FOLDERNAME-S-003-M	Change folder name – PUT	Mandatory	
REST-NMS-FOLDERS-SEARCH-S-001-M	Support search operation for folders	Mandatory	

Item	Function	Status	Requirement
REST-NMS-FOLDERS-SEARCH-S-002-M	Search for the root folder(s) – POST	Mandatory	
REST-NMS-FOLDERS-SEARCH-S-003-O	Support additional search operations other than searching for root – POST	Optional	
REST-NMS-FOLDERS-PATHTOID-S-001-M	Support for looking up folder(s) resource URL(s) using its (their) location/path	Mandatory	
REST-NMS-FOLDERS-PATHTOID-S-002-M	Retrieve an folder's Id (resource URL) using its location/path – GET	Mandatory	
REST-NMS-FOLDERS-PATHTOID-S-003-M	Retrieve a list of folders Ids (resource URLs) using their location/path – POST	Mandatory	
REST-NMS-FOLDERS-COPY-S-001-M	Support for copying objects and/or folders to a target folder	Mandatory	
REST-NMS-FOLDERS-COPY-S-002-M	Copy objects into a target folder – POST	Mandatory	
REST-NMS-FOLDERS-COPY-S-003-M	Copy a folder containing other folders and objects into a target folder (recursive copy) – POST	Mandatory	
REST-NMS-FOLDERS-MOVE-S-001-M	Support for moving objects and/or folders to a target folder	Mandatory	
REST-NMS-FOLDERS-MOVE-S-002-M	Move objects into a target folder – POST	Mandatory	
REST-NMS-FOLDERS-MOVE-S-003-M	Move a folder containing other folders and objects into a target folder (recursive move) – POST	Mandatory	
REST-NMS-SUBSCR-S-001-O	Support for subscriptions to NMS event notifications as well as synchronization with NMS	Mandatory	
REST-NMS-SUBSCR-S-002-O	Read the list of active subscriptions to NMS event notifications – GET	Optional	
REST-NMS-SUBSCR-S-003-O	Create new subscription to NMS event notifications – POST	Mandatory	
REST-NMS-SUBSCR-S-004-O	Create new subscription to NMS event notifications while it syncs the local storage with NMS – POST	Mandatory	

Item	Function	Status	Requirement
REST-NMS-SUBSCR-S-005-O	Create new subscription to NMS event notifications with filter setup to receive only certain event (e.g. SMS's)- POST	Optional	
REST-NMS-INDSUBSCR-S-001-O	Support for access to an individual subscription to NMS event notifications	Mandatory	
REST-NMS-INDSUBSCR-S-002-O	Read an individual subscription to NMS event notifications – GET	Mandatory	
REST-NMS-INDSUBSCR-S-003-O	Update an individual subscription to NMS event notifications – POST	Mandatory	
REST-NMS-INDSUBSCR-S-004-O	Cancel subscription and stop corresponding notifications – DELETE	Mandatory	
REST-NMS-NOTIF-S-001-O	Support for notifications about NMS events	Mandatory	
REST-NMS-NOTIF-S-002-O	Notifications about NMS changes – GET	Mandatory	

Appendix C. Example of a Message Object

Example of a CPIM message deposited by the CPM Client in the NMS Message Store:

```
Content-Type: message/CPIM
From: +1818118181...
To: +1818118181...
Date: 1 January 2014 14:30:30+00:00
[... other SIP or MSRP framing omitted...]

From: MR SANDERS <im:piglet@100akerwood.com>
To: Dopey Donkey <im:eyore@100akerwood.com>
DateTime: 2014-12-13T13:40:00-08:00
Subject: the weather will be fine today
Subject;;lang=fr beau temps prevu pour aujourd'hui
NS: MyFeatures <mid:MessageFeatures@id.foo.com>
Require: MyFeatures.VitalMessageOption
MyFeatures.VitalMessageOption: Confirmation-requested
MyFeatures.WackyMessageOption: Use-silly-font

Content-Type: multipart/mixed; boundary=7fekyzC-uQhXhZCGky5kU7wP

--7fekyzC-uQhXhZCGky5kU7wP
Content-Disposition: attachment; filename=body.txt; name=body.txt
Content-Type: text/plain; charset=UTF-8

See attached photo

--7fekyzC-uQhXhZCGky5kU7wP
Content-Disposition: attachment; filename=picture.gif; name=picture.gif
Content-Type: image/gif

[binaryData]
--7fekyzC-uQhXhZCGky5kU7wP--
--3UOXhVDKZAwJuefmkecOX5gI--
```

Example of a message object deposited by the CPM Client in the NMS Message Store:

```
POST /nms/v1/base/tel:+12021308085/objects HTTP/1.1
Host: server
Authorization: Bearer SBS9V8OWX4o7GDv9DRhux0hLcGzc7q165S/iU+Ri9vJC7iw=
MIME-Version: 1.0
Accept: application/json, */*
Content-Type: multipart/form-data; boundary=3UOXhVDKZAwJuefmkecOX5gI
Content-Length: 1210

--3UOXhVDKZAwJuefmkecOX5gI
Content-Disposition: form-data; name=root-fields
Content-Type: application/json; charset=UTF-8

{
  "object": {
    "parentFolder":
"http://server/nms/v1/base/tel:+12021308085/folders/UGZa5FY4o1S1oTSQSQC",
```

```

"attributes": {
  "attribute": [{
    "name": "Subject",
    "value": ["the weather will be fine today "]
  }, {
    "name": "From",
    "value": ["+1818118181"]
  }, {
    "name": "Message-Context",
    "value": ["multimedia-message"]
  }, {
    "name": "Date",
    "value": ["2014-02-01T14:30:30Z"]
  }, {
    "name": "To",
    "value": ["+1818118181"]
  }, {
    "name": "Direction",
    "value": ["IN"]
  }, {
    "name": "CPIM",
    "value": ["From: MR SANDERS <im:piglet@100akerwood.com>\r\nTo: Dopey Donkey
<im:eyore@100akerwood.com>\r\nDateTime: 2014-12-13T13:40:00-08:00\r\nSubject: the
weather will be fine today\r\nSubject;;lang=fr beau temps prevu pour
aujourd'hui\r\nNS: MyFeatures <mid:MessageFeatures@id.foo.com>\r\nRequire:
MyFeatures.VitalMessageOption\r\nMyFeatures.VitalMessageOption: Confirmation-
requested\r\nMyFeatures.WackyMessageOption: Use-silly-font"]
  }]
},
"correlationId": "someId"
}
}
--3UOXhVdKZAwJuefmkecOX5gI
Content-Disposition: form-data; name=attachments
Content-Type: multipart/mixed; boundary=7fekyzC-uQhXhZCGky5kU7wP

--7fekyzC-uQhXhZCGky5kU7wP
Content-Disposition: attachment; filename=body.txt; name=body.txt
Content-Type: text/plain; charset=UTF-8

See attached photo

--7fekyzC-uQhXhZCGky5kU7wP
Content-Disposition: attachment; filename=picture.gif; name=picture.gif
Content-Type: image/gif

[binaryData]
--7fekyzC-uQhXhZCGky5kU7wP--
--3UOXhVdKZAwJuefmkecOX5gI--

```

When the object is retrieved from NMS, the metadata (retrieved by GET <http://server/nms/v1/base/tel:+12021308085/objects/NNNN>) is as follows:

```

GET /nms/v1/base/tel:+12021308085/objects/obj542 HTTP/1.1
Host: server
Authorization: Bearer SBS9V8OWX4o7GDv9DRhux0hLcGzc7q165S/iU+Ri9vJC7iw=

```

```

MIME-Version: 1.0
Accept: application/json, */*

HTTP/1.1 200 OK
Date: Fri, 03 Mar 2017 00:09:33 GMT
Content-Type: application/json
Content-Length: nnn

{ "object": {
  "parentFolder":
"http://server/nms/v1/base/tel:+12021308085/folders/UGZa5FY4o1S1oTSQSQ",
  "attributes": {
    "attribute": [{
      "name": "Subject",
      "value": ["the weather will be fine today "]
    }, {
      "name": "From",
      "value": ["+1818118181"]
    }, {
      "name": "Message-Context",
      "value": ["multimedia-message"]
    }, {
      "name": "Date",
      "value": ["2014-02-01T14:30:30Z"]
    }, {
      "name": "To",
      "value": ["+1818118181"]
    }, {
      "name": "Direction",
      "value": ["IN"]
    }, {
      "name": "Content-Type",
      "value": [ "multipart/mixed" ]
    }, {
      "name": "CPIM",
      "value": ["From: MR SANDERS <im:piglet@100akerwood.com>\r\nTo: Dopey Donkey
<im:eyore@100akerwood.com>\r\nDateTime: 2014-12-13T13:40:00-08:00\r\nSubject: the
weather will be fine today\r\nSubject:;lang=fr beau temps prevu pour
aujourd'hui\r\nNS: MyFeatures <mid:MessageFeatures@id.foo.com>\r\nRequire:
MyFeatures.VitalMessageOption\r\nMyFeatures.VitalMessageOption: Confirmation-
requested\r\nMyFeatures.WackyMessageOption: Use-silly-font"]
    }
  ]
},
  "flags": {
    "flag": [
      "\\Seen",
      "\\Flagged"
    ]
  }
},
  "resourceURL": "http://server/nms/v1/base/tel:+12021308085/objects/obj542",
  "path": "/main/conversation5/obj542",
  "payloadURL":
"http://server/nms/v1/base/tel:+12021308085/objects/obj542/payload",
  "payloadPart": [
    { "contentType": "text/plain",
      "contentDisposition": "attachment; filename=body.txt; name=body.txt",
      "size": 48,

```

```
    "href": " http://server/nms/v1/base/tel:+12021308085/objects/obj542/payload
Parts/blob123"
  },
  { "contentType": "image/gif",
    "contentDisposition": "attachment; filename=picture.gif; name=picture.gif",
    "size": 1024,
    "href": "
http://server/nms/v1/base/tel:+12021308085/objects/obj542/payloadParts/blob457 "
  }
],
"lastModSeq": 48
}
}
```

Appendix D. Example of Session History Folder

Below is an example of a 1-1 session history folder with the following content:

- One session info object.
- One CPM Chat Message with text content in message object 1.

Another example is provided in Appendix F.

The session info object as deposited into NMS

```
POST /nms/v1/base/tel:+12021308085/objects HTTP/1.1
Host: server
Authorization: Bearer zSBS9V8OWX4o7GDv9DRhux0hLcGzc7q165S/iU+Ri9vJC7iw=
MIME-Version: 1.0
Accept: application/json, */*
Content-Type: multipart/form-data; boundary=LH6uxl4xe5sJ93i57iJLOyNl
Content-Length: 1024

--LH6uxl4xe5sJ93i57iJLOyNl
Content-Disposition: form-data; name=root-fields
Content-Type: application/json; charset=UTF-8

{
  "object": {
    "parentFolder": "http://server/nms/v1/base/tel:+12021308085/folders/UGza5FY4o1S1oTSQSQC",
    "attributes": {
      "attribute": [{
        "name": "Subject",
        "value": ["the weather will be fine today"]
      }, {
        "name": "From",
        "value": ["+1818118181"]
      }, {
        "name": "Message-Context",
        "value": ["session-info-message"]
      }, {
        "name": "Date",
        "value": ["2014-02-01T14:29:00Z"]
      }, {
        "name": "To",
        "value": ["+1818118181"]
      }, {
        "name": "Direction",
        "value": ["IN"]
      }, {
        "name": "Conversation-Id",
        "value": ["f81d4fae-7dec-11d0-a765-00a0c91e6bf5"]
      }, {
        "name": "Contribution-Id",
        "value": ["abcdef-1234-5678-90ab-cdef01234566"]
      }, {
        "name": "InReplyTo-Contribution-Id",
        "value": ["01234567-89ab-cdef-0123-456789abcdee"]
      }
    ]
  },
  "correlationId": "654131a654131a654131a654131a8994653"
}
--LH6uxl4xe5sJ93i57iJLOyNl
Content-Disposition: form-data; name=message
Content-Type: multipart/mixed; boundary=4vImznJai6jqZFelH2zuHe+P

--4vImznJai6jqZFelH2zuHe+P
Content-Type: Application/X-CPM-Session
```

```
<?xml version="1.0" encoding="UTF-8"?>
<session>
<session-type>l-1</session-type>
</session>
```

```
--4vImznJai6jqZFelH2zuHe+P--
--LH6uxl4xe5sJ93i57iJLOyNl--
```

The message object 1 carrying a CPM Chat Message as deposited into NMS

```
POST /nms/v1/base/tel:+12021308085/objects HTTP/1.1
```

```
Host: server
```

```
Authorization: Bearer zSBS9V8OWX4o7GDv9DRhux0hLcGzc7q165S/iU+Ri9vJC7iw=
```

```
MIME-Version: 1.0
```

```
Accept: application/json, */*
```

```
Content-Type: multipart/form-data; boundary=LH6uxl4xe5sJ93i57iJLOyNl
```

```
Content-Length: 1024
```

```
--LH6uxl4xe5sJ93i57iJLOyNl
```

```
Content-Disposition: form-data; name=root-fields
```

```
Content-Type: application/json; charset=UTF-8
```

```
{
  "object": {
    "parentFolder": "http://server/nms/v1/base/tel:+12021308085/folders/UGZa5FY4o1S1oTSQSQ",
    "attributes": {
      "attribute": [{
        "name": "Subject",
        "value": ["ObjectsPostSpec test message"]
      }, {
        "name": "From",
        "value": ["+1818118181"]
      }, {
        "name": "Message-Context",
        "value": ["chat-message"]
      }, {
        "name": "Date",
        "value": ["2014-02-01T14:30:30Z"]
      }, {
        "name": "To",
        "value": ["+1818118181"]
      }, {
        "name": "Direction",
        "value": ["IN"]
      }, {
        "name": "Conversation-Id",
        "value": ["a2133b-654131a-f564321-c5d655"]
      }, {
        "name": "Contribution-Id",
        "value": ["683135-a654bb2-35c641d-84679f"]
      }
    ]
  },
  "correlationId": "654131a654131a654131a654131a8994657"
}
```

```
--LH6uxl4xe5sJ93i57iJLOyNl
```

```
Content-Disposition: form-data; name=message
```

```
Content-Type: multipart/mixed; boundary=4vImznJai6jqZFelH2zuHe+P
```

```
--4vImznJai6jqZFelH2zuHe+P
```

```
Content-Disposition: attachment; filename=sms; name=sms
```

```
Content-Type: text/plain; charset=utf-8
```

Here is the text of my message.

```
--4vImznJai6jqZFelH2zuHe+P--  
--LH6uxl4xe5sJ93i57iJLOyNl--
```


Appendix E. Example of File Transfer History Object

Below is an example of a 1-1 File Transfer History Object in which one file was received.

```
POST /nms/v1/base/tel:+12021308085/objects HTTP/1.1
Host: server
Authorization: Bearer zSBS9V8OWX4o7GDv9DRhux0hLcGzc7q165S/iU+Ri9vJC7iw=
MIME-Version: 1.0
Accept: application/json, */*
Content-Type: multipart/form-data; boundary=LH6uxl4xe5sJ93i57iJLOyN1
Content-Length: 1024

--LH6uxl4xe5sJ93i57iJLOyN1
Content-Disposition: form-data; name=root-fields
Content-Type: application/json; charset=UTF-8

{
  "object": {
    "parentFolder": "http://server/nms/v1/base/tel:+12021308085/folders/UGZa5FY4o1S1oTSQSQC",
    "attributes": {
      "attribute": [{
        "name": "From",
        "value": ["+1818118181"]
      }, {
        "name": "Message-Context",
        "value": ["file-message"]
      }, {
        "name": "Date",
        "value": ["1997-11-21T15:55:06Z"]
      }, {
        "name": "To",
        "value": ["+1818118181"]
      }, {
        "name": "Direction",
        "value": ["IN"]
      }, {
        "name": "Conversation-Id",
        "value": ["f81d4fae-7dec-11d0-a765-00a0c91e6bf6"]
      }, {
        "name": "Contribution-Id",
        "value": ["abcdef-1234-5678-90ab-cdef01234567"]
      }, {
        "name": "InReplyTo-Contribution-Id",
        "value": ["01234567-89ab-cdef-0123-456789abcdef"]
      }
    ]
  },
  "correlationId": "654131a654131a131bfrufh37846r44tcbrfb94656"
}
--LH6uxl4xe5sJ93i57iJLOyN1
Content-Disposition: form-data; name=message
Content-Type: multipart/related; boundary=4vImznJai6jqZFelH2zuHe+P; type="Application/X-CPM-File-Transfer"

--4vImznJai6jqZFelH2zuHe+P
Content-Disposition: attachment; filename=sms; name=sms
Content-Type: Application/X-CPM-File-Transfer; charset=utf-8

<?xml version="1.0" encoding="UTF-8"?>
<file-transfer>
  <file-transfer-type>1-1</file-transfer-type>
  <imdn></imdn>
  <sdp>
```

```
i=This is my latest picture
a=sendonly
a=file-selector:name:"My picture.jpg" type:image/jpeg size:4092
a=file-disposition:render
a=file-date:creation:"Mon, 15 May 2006 15:01:31 +0300"
a=file-icon:cid:mythumbnail@example.com
</sdp>
<file-object>
  <cid>cid:1234@example.com</cid>
</file-object>
</file-transfer>

--4vImznJai6jqZFelH2zuHe+P
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <mythumbnail@example.com>

  ... mythumbnail.jpg...

--4vImznJai6jqZFelH2zuHe+P
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <1234@example.com>

  ... My picture.jpg...

--4vImznJai6jqZFelH2zuHe+P--
--LH6uxl4xe5sJ93i57iJLOyNl--
```

Appendix F. Example of Session Info Object

Below is an example of a Session Info object, illustrating the alternative form for uploading a message object with only a single part. In this case the inner multipart may be omitted, as described in [OMA-REST-NMS].

```
POST /nms/v1/base/tel:+12021308085/objects HTTP/1.1
Host: server
Authorization: Bearer zSBS9V8OWX4o7GDv9DRhux0hLcGzc7q165S/iU+Ri9vJC7iw=
MIME-Version: 1.0
Accept: application/json, */*
Content-Type: multipart/form-data; boundary=LH6uxl4xe5sJ93i57iJLOyN1
Content-Length: 1024

--LH6uxl4xe5sJ93i57iJLOyN1
Content-Disposition: form-data; name=root-fields
Content-Type: application/json; charset=UTF-8

{
  "object": {
    "parentFolder": "http://server/nms/v1/base/tel:+12021308085/folders/UGZa5FY4o1S1oTSQSQC",
    "attributes": {
      "attribute": [{
        "name": "From",
        "value": ["+1818118181"]
      }, {
        "name": "Message-Context",
        "value": ["session-info-message"]
      }, {
        "name": "Date",
        "value": ["1997-11-21T15:55:06Z"]
      }, {
        "name": "Subject",
        "value": ["the weather will be fine today"]
      }, {
        "name": "To",
        "value": ["+1818118181"]
      }, {
        "name": "Direction",
        "value": ["IN"]
      }, {
        "name": "Conversation-Id",
        "value": ["f81d4fae-7dec-11d0-a765-00a0c91e6bf6"]
      }, {
        "name": "Contribution-Id",
        "value": ["abcdef-1234-5678-90ab-cdef01234567"]
      }, {
        "name": "InReplyTo-Contribution-Id",
        "value": ["01234567-89ab-cdef-0123-456789abcdef"]
      }
    ]
  }
}
--LH6uxl4xe5sJ93i57iJLOyN1
Content-Disposition: form-data; name=message
Content-Type: Application/X-CPM-Session

<session-type>1-1</session-type>
<sdp>
a=sendonly
m=audio 49170 RTP/AVP 0 97
</sdp>
--LH6uxl4xe5sJ93i57iJLOyN1--
```

Appendix G. Example of Group State Object

Below is an example of a Group State object, showing the alternative form for uploading a message object with only a single part. In this case the inner multipart may be omitted, as described in [OMA-REST-NMS].

```
POST /nms/v1/base/tel:+16135551212/objects HTTP/1.1
Host: server
Authorization: Bearer zSBS9V8OWX4o7GDv9DRhux0hLcGzc7q165S/iU+Ri9vJC7iw=
MIME-Version: 1.0
Accept: application/json, */*
Content-Type: multipart/form-data; boundary=LH6uxl4xe5sJ93i57iJLOyN1
Content-Length: 1024

--LH6uxl4xe5sJ93i57iJLOyN1
Content-Disposition: form-data; name=root-fields
Content-Type: application/json; charset=UTF-8

{
  "object": {
    "parentFolder": "http://server/nms/v1/base/tel:+12021308085/folders/UGZa5FY4o1S1oTSQSQC",
    "attributes": {
      "attribute": [{
        "name": "From",
        "value": ["+1818118181"]
      }, {
        "name": "Message-Context",
        "value": ["group-state-message"]
      }, {
        "name": "Date",
        "value": ["2012-06-13T16:39:57-05:00"]
      }, {
        "name": "To",
        "value": ["+15195551212", "+15145551212"]
      }, {
        "name": "Direction",
        "value": ["IN"]
      }
    ]
  }
}

--LH6uxl4xe5sJ93i57iJLOyN1
Content-Disposition: form-data; name=message
Content-Type: application/group-state-object+xml

<groupstate timestamp="2012-06-13T16:39:57-05:00"
lastfocussessionid="da9274453@company.com"
  group-type="closed">
  <participant name="bob" comm-addr="tel:+16135551212"/>
  <participant name="alice" comm-addr="tel:+15195551212"/>
  <participant name="bernie" comm-addr="tel:+15145551212"/>
</groupstate>

--LH6uxl4xe5sJ93i57iJLOyN1--
```