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.

© 2012 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 .......................................................................................................................... 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 .................................................................................................... 8

4. INTRODUCTION ............................................................................................................. 10
   4.1 VERSION 1.0 .......................................................................................................... 10

5. COMMON PROCEDURES ................................................................................................. 11
   5.1 AUTHORIZATION AND AUTHENTICATION .......................................................... 11
      5.1.1 Authentication ............................................................................................... 11
      5.1.2 Authorization ............................................................................................... 11
   5.2 STORAGE OBJECTS ................................................................................................. 11
      5.2.1 Message Object .............................................................................................. 11
      5.2.2 Session History Object .................................................................................. 12
      5.2.3 File Transfer History Object ......................................................................... 13
      5.2.4 Conversation History Object .......................................................................... 15
      5.2.5 Stand-alone Media Object .............................................................................. 15
      5.2.6 Storage Mailbox ............................................................................................. 16
   5.3 IDENTIFICATION OF STORAGE OBJECTS ......................................................... 16
   5.4 NOTIFICATIONS ....................................................................................................... 16
   5.5 METADATA STRUCTURE .......................................................................................... 16

6. PROCEDURES AT MESSAGE STORAGE CLIENT ....................................................... 18
   6.1 GENERAL OPERATIONS ....................................................................................... 18
      6.1.1 Authenticate Operation ................................................................................... 18
      6.1.2 Set Active Folder Operation ........................................................................... 18
   6.2 ACCESS CONTROL LIST OPERATIONS .................................................................. 18
      6.2.1 Set Access Control List ................................................................................... 18
      6.2.2 Get Access Control List .................................................................................. 18
      6.2.3 Delete Access Control List .............................................................................. 19
      6.2.4 Access Rights Retrieval Operations .................................................................. 19
   6.3 MESSAGE AND HISTORY OPERATIONS ................................................................ 19
      6.3.1 Object Store Operation .................................................................................... 19
      6.3.2 Object Fetch Operation .................................................................................... 19
      6.3.3 Object Preview Fetch Operation ...................................................................... 19
      6.3.4 Object Copy Operation .................................................................................... 20
      6.3.5 Object Remove Operation ............................................................................... 20
   6.4 FOLDER OPERATIONS ............................................................................................. 20
      6.4.1 Folder Create Operation ................................................................................... 20
      6.4.2 List Folder Operation ....................................................................................... 20
      6.4.3 Folder Move Operation .................................................................................... 20
      6.4.4 Folder Remove Operation ............................................................................... 20
      6.4.5 Folder Search Operation ................................................................................. 20
   6.5 REFERENCE OPERATIONS ...................................................................................... 21
      6.5.1 Generate Reference Operation ......................................................................... 21
      6.5.2 Fetch by Reference Operation ......................................................................... 21
   6.6 MESSAGE AND HISTORY METADATA MANAGEMENT OPERATIONS ................. 21
      6.6.1 Metadata Operation ........................................................................................ 21
   6.7 SYNCHRONIZATION ............................................................................................... 21
   6.8 NOTIFICATION OPERATIONS ................................................................................. 21
7. PROCEDURES AT MESSAGE STORAGE SERVER ............................................................... 23
   7.1 GENERAL OPERATIONS ..................................................................................... 23
      7.1.1 Authenticate Operation ............................................................................ 23
      7.1.2 Set Active Folder Operation .................................................................. 23
   7.2 ACCESS CONTROL LIST OPERATIONS ............................................................. 23
      7.2.1 Set Access Control List .......................................................................... 23
      7.2.2 Get Access Control List ......................................................................... 23
      7.2.3 Delete Access Control List .................................................................... 23
      7.2.4 Access Rights Retrieval Operations ...................................................... 23
   7.3 MESSAGE AND HISTORY OPERATIONS ............................................................. 24
      7.3.1 Object Store Operation .......................................................................... 24
      7.3.2 Object Fetch Operation .......................................................................... 24
      7.3.3 Object Preview Fetch Operation ............................................................. 24
      7.3.4 Object Copy Operation ........................................................................... 24
      7.3.5 Object Remove Operation ...................................................................... 24
   7.4 FOLDER OPERATIONS .......................................................................................... 24
      7.4.1 Folder Create Operation ......................................................................... 24
      7.4.2 List Folders Operation ............................................................................ 25
      7.4.3 Folder Move Operation ........................................................................... 25
      7.4.4 Folder Remove Operation ...................................................................... 25
      7.4.5 Folder Search Operation ........................................................................ 25
   7.5 REFERENCE OPERATIONS ................................................................................... 25
      7.5.1 Generate Reference Operation ................................................................ 25
      7.5.2 Fetch by Reference Operation ................................................................. 25
   7.6 METADATA UPDATE OPERATIONS ..................................................................... 25
   7.7 MESSAGE AND HISTORY SYNCHRONIZATION OPERATIONS ......................... 25
   7.8 NOTIFICATIONS OPERATIONS .......................................................................... 26
APPENDIX A. CHANGE HISTORY (INFORMATIVE) .............................................................. 27
   A.1 APPROVED VERSION 1.0 HISTORY .................................................................. 27
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ......................... 28
   B.1 SCR FOR MESSAGE STORAGE CLIENT .......................................................... 28
   B.2 SCR FOR MESSAGE STORAGE SERVER ......................................................... 29
APPENDIX C. APPENDIX C. CPM-DEFINED IMAP FLAG EXTENSIONS ............................ 30
   C.1 \READ-REPORT-SENT .................................................................................. 30
APPENDIX D. EXAMPLE OF SESSION HISTORY OBJECT ............................................ 31
APPENDIX E. EXAMPLE OF FILE TRANSFER HISTORY OBJECT ................................... 32

Tables

Table 1: CPM session history object format .................................................................... 13
Table 2: CPM file transfer history object format ............................................................ 15
1. Scope

This document provides the technical specifications for the message storage functionality of the CPM Enabler. The document covers the storage of Media Objects, CPM Messages, CPM File Transfer Histories, CPM Session Histories and CPM Conversation Histories in the network and the interactions between client and server components to access the network storage. The technical specifications are designed to fulfill the requirements, architecture and system concepts that are described in [OMA-CPM-RD], [OMA-CPM-AD] and [OMA-CPM-SD] respectively.

As such, these technical specifications provide the formal definitions 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].
2. References

2.1 Normative References


[OMA-CPM-SD] “Converged IP Messaging System Description”, Open Mobile Alliance™, OMA-TS-CPM_System_Description-V1_0, URL:http://www.openmobilealliance.org/


2.2 Informative References


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

<table>
<thead>
<tr>
<th>Term</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPM Address</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Chat Message</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Conversation</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Conversation History</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Conversation Identity</td>
<td>See [OMA-CPM-SD].</td>
</tr>
<tr>
<td>CPM File Transfer</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM File Transfer History</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Message</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Participating Function</td>
<td>See [OMA-CPM-AD].</td>
</tr>
<tr>
<td>CPM Pre-defined Group</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Session History</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Standalone Message</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM User</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>Media Object</td>
<td>See [OMA-CPM-AD].</td>
</tr>
<tr>
<td>CPM Session</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>CPM Session Invitation</td>
<td>See [OMA-CPM-RD].</td>
</tr>
<tr>
<td>Message Storage Client</td>
<td>See [OMA-CPM-AD].</td>
</tr>
<tr>
<td>Message Storage Server</td>
<td>See [OMA-CPM-AD].</td>
</tr>
<tr>
<td>Principal</td>
<td>See [OMADICT].</td>
</tr>
</tbody>
</table>

**Unique IDentifier**

A 32-bit value assigned to each stored object and used with the UID validity value to form a 64-bit value that is unique to a stored object and cannot be referred to any other stored object in the folder or any subsequent folder of the message storage [RFC3501].

**UIDVALIDITY**

A 32-bit representation of the creation date and time of a folder

3.3 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>Access Control List</td>
</tr>
<tr>
<td>CPIM</td>
<td>Common Presence and Instant Messaging</td>
</tr>
<tr>
<td>CPM</td>
<td>Converged IP Messaging</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IMAP</td>
<td>Internet Message Access Protocol</td>
</tr>
<tr>
<td>OMA</td>
<td>Open Mobile Alliance</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>PSK</td>
<td>Pre Shared Key</td>
</tr>
<tr>
<td>SASL</td>
<td>Simple Authentication and Security Layer</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>UID</td>
<td>Unique (Message) Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
</tbody>
</table>
4. Introduction

The CPM message storage functionality allows the storage of 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.

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.

4.1 Version 1.0

The CPM 1.0 offers:

- **Storage objects**
  - CPM Standalone Messages, CPM File Transfer Histories, CPM Session Histories and CPM Conversation Histories
  - any potential Media objects either stand-alone or attached to CPM Standalone Messages, CPM File Transfer Histories and CPM Session Histories

- **User authentication and authorization mechanisms**

- **Operations**
  - folder operations e.g., create, list, set active folder, move, remove, search
  - stored object operations e.g., store, fetch, copy, remove, preview
  - metadata operations on stored objects e.g., update metadata
  - access rights on stored objects e.g., set, get, delete

- **generate references to stand-alone or attached to CPM Standalone Messages and CPM Session History objects**

- **allows fetch of stored Media objects by reference**

- **synchronize between network-based storage and device’s local storage**

- **Notifications**
  - notifications about changes in stored resources
5. Common Procedures

5.1 Authorization and Authentication

The IMAPv4 [RFC3501] protocol is used to access the Message Storage Server. This section defines authentication and authorization mechanisms of IMAPv4 used by the Message Storage Server.

5.1.1 Authentication

CPM’s message storage functionality supports the two authentication mechanisms listed below, as defined for IMAPv4 [RFC3501];

1. SASL authentication via the AUTHENTICATE command ([RFC3501]); and
2. Username/password in plain text authentication via the LOGIN command ([RFC3501]).

The username/password for the second authentication mechanism is separately managed by the CPM service. The password may be pre-configured by the CPM system when the CPM user subscribes to the CPM service.

In addition to these authentication mechanisms, TLS/PSK-TLS, as defined in [RFC2246] and [OMA-SEC-CF], is optional and complementary to simple authentication-only SASL mechanisms or deployed clear-text password login commands. In this way, IMAPv4 can be immune to eavesdropping and hijacking attacks. Using TLS/PSK-TLS, the Message Storage Client also can authenticate the Message Storage Server by checking the certificate supplied by the Message Storage Server.

5.1.2 Authorization

The Message Storage Server uses standard IMAP4 [RFC3501] functionality to enforce access to the stored resources, extended with the possibility for CPM Users to define access control lists (ACL) for their own stored resources.

The ACL management operations and related standard rights are defined in IMAPv4 ACL extension [RFC4314].

In addition to that, the Message Storage Server also allows the use of the IMAP4 URLAUTH extension, as defined in [RFC4467] and [RFC5092]. This URLAUTH extension provides a means by which a Message Storage Client can use URLs carrying authorization information to access limited data on the Message Storage Server.

5.2 Storage Objects

Per the description provided in the CPM System Description [OMA-CPM-SD], there can be five different categories of “storage objects” and one special folder named as “storage mailbox” present in a CPM Message Storage Server. The storage objects are message, session history, file transfer history, conversation history and stand-alone media. In this section, these storage objects as well as the storage mailbox are specified in terms of their names and identities, which can be used for various CPM message storage operations in accordance with IMAP4 [RFC3501] and its extensions.

5.2.1 Message Object

A message object, matching the message concept described in [RFC3501], is a Message Storage Server stored item. For execution of various IMAP commands, the message object is identified by a 3-component identifier consisting of a folder name, a message identification number associated to the message object and a folder validity value as specified below:

1. The folder name is the name of the folder in the Message Storage Server where the message is stored,
2. The message identification number is either a message sequence number or a 32-bit Unique Identifier (UID), which is specified according to [RFC3501],
3. The folder validity value or the Unique IDentifier validity (UIDVALIDITY) is another 32-bit value as defined in [RFC3501] distinguishing folders of the same name from each other.
5.2.2 Session History Object

The definition and identity specification of the message object SHALL be applicable to the session history object, which consists of a UID, UIDVALIDITY and their values. The session history object is a message formatted according to [] and the clarifications given in below.

The content inserted in the headers of this message is retrieved from the CPM Session Invitation.

The body of this message SHALL be a “multipart/related” body, structured as follows:

- the first body part SHALL contain metadata about the CPM Session History and is formatted according to section 5.2.2.1 “Application/X-CPM-Session Content Type Definition”
- the other body parts SHALL contain the CPM Chat Messages and media that were sent and received via MRSP and RTP during the CPM Session.

An example of a session history object is shown in Appendix D.

<table>
<thead>
<tr>
<th>Internet Message Format [RFC5322] header</th>
<th>Internet Message Format Parameter status</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Mandatory</td>
<td>Set to the address of the initiator of the CPM Session, retrieved from the authenticated originator’s CPM Address in the SIP INVITE request.</td>
</tr>
<tr>
<td>Date</td>
<td>Mandatory</td>
<td>Set to, in order of preference:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- the Date of the SIP INVITE request if available, otherwise,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- the date and time when the CPM Session recording was started.</td>
</tr>
<tr>
<td>Subject</td>
<td>Optional</td>
<td>Set only if “Subject” header is set in the SIP INVITE request.</td>
</tr>
<tr>
<td>Conversation-ID</td>
<td>Mandatory</td>
<td>Set to the Conversation-ID of the SIP INVITE request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTE: this is a new field, extending the Internet Message Format in accordance with 3.6.8 of[].</td>
</tr>
<tr>
<td>Contribution-ID</td>
<td>Mandatory</td>
<td>Set to the Contribution-ID of the SIP INVITE request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTE: this is a new field, extending the Internet Message Format in accordance with 3.6.8 of[].</td>
</tr>
<tr>
<td>InReplyTo- Contribution-ID</td>
<td>Optional</td>
<td>Set to the InReplyTo- Contribution-ID of the SIP INVITE request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTE: this is a new field, extending the Internet Message Format in accordance with 3.6.8 of[].</td>
</tr>
</tbody>
</table>
| Content type | Mandatory | Set to the following value: multipart/related;boundary=cpm; type="Application/X-CPM-Session"

NOTE: the new “Application/X-CPM-Session” content-type is defined in section 5.2.3.1. |
| Message body | Mandatory | Contains a list of MIME entities separated by the “cpm” boundary.

The first MIME entity is the “root”, as defined in [RFC2387] and its content-type is “Application/X-CPM-Session”.

The other MIME entities are:
- the CPM Chat Messages, each formatted as a CPIM MIME entity in accordance with [RFC3862], and,
- the Media Objects that were exchanged during the CPM Session. |

Table 1: CPM session history object format

5.2.2.1 Application/X-CPM-Session Content Type Definition

This new MIME type is used to store metadata about the CPM Session.

The body of a message with the content-type header set to Application/X-CPM-Session SHALL be formatted as follows:

```xml
<session-type>: this element indicates the type of session. It may be set to “Ad-Hoc”, “Pre-Defined” or “1-1”.

<invited-participants>: this element contains the list of addresses of the invited participants, separated by a semi-colon, or the address of the Pre-defined Group.

Next, for each Media Object, metadata information MAY be stored within a <media-object> element as follows:

<media-object>

  <cid>

  cid: set to the Content-ID, as defined in [RFC2392], of the corresponding stored Media Object.

  </cid>

  <sdp>SDP parameters associated with the corresponding stored Media Object.</sdp>

</media-object>
```

The format of the body MAY be extended to store additional metadata information (e.g., participants who joined or left the session).

5.2.3 File Transfer History Object

The definition and identity specification of the message object SHALL be applicable to the file transfer history object, which consists of a UID, UIDVALIDITY and their values.
The file transfer history object is a message formatted according to [] and the clarifications given in below.

The content inserted in the headers of this message is retrieved from the CPM File Transfer invitation.

The body of this message SHALL be a “multipart/related” body, structured as follows:

- the first body part SHALL contain metadata about the CPM File Transfer History and is formatted according to section 5.2.3.1 “Application/X-CPM-File-Transfer Content Type Definition”
- the other body parts SHALL contain the files that were sent and received via MRSP during the CPM File Transfer.

An example of a file transfer history object is shown in Appendix E.

<table>
<thead>
<tr>
<th>Internet Message Format [RFC5322] header</th>
<th>Internet Message Format Parameter status</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Mandatory</td>
<td>Set to the address of the initiator of the CPM File Transfer, retrieved from the authenticated originator’s CPM Address in the SIP INVITE request.</td>
</tr>
<tr>
<td>Date</td>
<td>Mandatory</td>
<td>Set to, in order of preference:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- the Date of the SIP INVITE request if available, otherwise,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- the date and time when the CPM File Transfer recording was started.</td>
</tr>
<tr>
<td>Subject</td>
<td>Optional</td>
<td>Set only if “Subject” header is set in the SIP INVITE request.</td>
</tr>
<tr>
<td>Conversation-ID</td>
<td>Mandatory</td>
<td>Set to the Conversation-ID of the SIP INVITE request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTE: this is a new field, extending the Internet Message Format, as defined in 3.6.8 of [].</td>
</tr>
<tr>
<td>Contribution-ID</td>
<td>Mandatory</td>
<td>Set to the Contribution -ID of the SIP INVITE request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTE: this is a new field, extending the Internet Message Format, as defined in 3.6.8 of [].</td>
</tr>
<tr>
<td>InReplyTo- Contribution-ID</td>
<td>Optional</td>
<td>Set to the InReplyTo -Contribution -ID of the SIP INVITE request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTE: this is a new field, extending the Internet Message Format, as defined in 3.6.8 of [].</td>
</tr>
<tr>
<td>Content type</td>
<td>Mandatory</td>
<td>Set to the following value: multipart/related;boundary=cpm; type=“Application/X-CPM-File-Transfer”</td>
</tr>
</tbody>
</table>
|                                       |                                      | NOTE: the new Application/X-CPM File-
<table>
<thead>
<tr>
<th>Message body</th>
<th>Transfer content-type is defined in section 5.2.4.1.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>Contains a list of MIME entities separated by the “cpm” boundary.</td>
</tr>
<tr>
<td></td>
<td>The first MIME entity is the “root”, as defined in [RFC2387] and its content-type is “Application/X-CPM-File-Transfer”.</td>
</tr>
<tr>
<td></td>
<td>The other MIME entities are the files that were exchanged during the CPM File Transfer.</td>
</tr>
</tbody>
</table>

Table 2: CPM file transfer history object format

5.2.3.1 Application/X-CPM-File-Transfer Content Type Definition

This new MIME type is used to store metadata about the CPM File Transfer.

The body of a message with the content-type header set to Application/X-CPM-File-Transfer SHALL be formatted as follows:

```xml
<file-transfer-type>: this element indicates the type of file transfer. It may be set to “Ad-Hoc”, “Pre-Defined” or “1-1”.

<invited-participants>: this element contains the list of addresses of the invited participants, separated by a semi-column, or the address of the Pre-defined Group.

Next, for each file, metadata information MAY be stored within a <file-object> element as follows:

<file-object>

  <cid>cid: set to the Content-ID, as defined in [RFC2392], of the corresponding stored file.</cid>
  <sdp>SDP parameters associated with the corresponding stored file (e.g., file name).</sdp>

</file-object>
```

5.2.4 Conversation History Object

The conversation history object is viewed as a special kind of sub-folder and is realized via the mailbox concept of IMAP4. A conversation history object stores all items related to a single CPM Conversation as elements (i.e. message objects, session history objects and standalone Media Objects) within the conversation history object. The name of the conversation history object is set to the CPM Conversation Identity associated with the CPM Conversation History that is stored in the conversation history object.

NOTE: The conversation history objects are exposed by the Message Storage Server to the Message Storage Clients; however, the Message Storage Client only shows the contents of the CPM Conversation History (represented as message objects and session history objects) in a manner selected by the CPM User (e.g. in a threaded view).

5.2.5 Stand-alone Media Object

The standalone Media Object is realized by the message concept of IMAP4, whereby the Media Object is wrapped into a MIME formatted object in order to fit into an IMAP4 message. Other than the formatting of the contents, the standalone Media Object is the same as the message object defined in section 5.2.1 “Error! Reference source not found.”, including the definition and naming specification of the message object in terms of UID, next UID and their values.
5.2.6 Storage Mailbox

The folder object storage mailbox is a Message Storage folder realized by the mailbox concept of IMAP4, described in [RFC3501]. The folder objectstorage mailbox is identified by the name given to it. The CPM folder objectstorage mailbox aligns with the rules and procedures for names of the mailbox concept of IMAP4, as described in [RFC3501].

5.3 Identification of Storage Objects

The combination of a folder name, its stored object’s UID, and the folder’s UIDVALIDITY MUST permanently and persistently refer to one and only one stored object in a Message Storage Server. In particular, the internal date, size, envelope, body structure, and message texts (HEADER, TEXT and all BODY fetch data items) MUST never change according to [RFC3501]. This requirement does not include message numbers, nor does it include object attributes that can be set by a STORE command (e.g., FLAGS).

Associated with any folder object and conversation history object, there is a next UID value, which is the predicted value that will be assigned to a new storage object in that folder or conversation history. Provided by the CPM Message Storage Server, this next UID is intended to provide a means for the CPM Message Storage Client to determine whether or not any messages have been delivered to the folder since the previous time it checked this value.

NOTE: The “next UID” value of a folder or conversation history changes whenever a new object is stored in that folder or conversation history.

5.4 Notifications

The Message Storage Server SHALL notify the Message Storage Client of the changes in the stored resources either solicited as a result of a client request or unsolicited and unilaterally according to the IMAP extension for the general notification model [RFC5551].

The data content of a notification may include a wide range of information as follows:

1. Folder size and stored objects status updates, e.g., addition and/or removal of messages to prevent synchronization errors,
2. “Next UID” for the changes to the storage requested by a third entity, e.g., CPM PF,
3. Stored objects’ attribute flags, e.g., “\Recent” flag for recently arrived objects in a folder.

5.5 Metadata Structure

CPM’s message storage functionality supports a metadata model that consists of two distinct parts:

1. A set of metadata flags that are associated with message objects, file transfer history objects, session 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 all the objects stored in the Message Storage Server. These annotations provide user-defined information that the user associates with these stored objects.

The Message Storage Client and the Message Storage Server SHALL at least support the metadata flags and MAY support the 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 [RFC3501], [RFC5788] and Appendix C:

- \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),
• $MDNSent (A disposition notification has been sent for this message),
• $Forwarded (message has been forwarded), and
• \read-report-sent (A read receipt has been sent for this message) as defined in Appendix C.

NOTE: All flag assignments and operations SHALL be handled according to the procedures specified in [RFC3501].

With respect to the metadata annotations, if supported, the Message Storage Client and the Message Storage Server SHALL support the structure defined in [RFC5464] for metadata annotations for conversation history objects and for folder objects, and the Message Storage Client and the Message Storage Server SHALL support the structure defined in [RFC5257] for message objects, for file transfer history objects, for session history objects and for standalone Media Objects.
6. Procedures at Message Storage Client

The Message Storage Client is a functional component of the CPM enabler, which allows the CPM User to view and manage (store, fetch, delete etc) the resources stored in the Message Storage Server. In addition to that the Message Storage Client notifies the CPM User of any changes to the stored resources in the Message Storage Server (e.g. new message arrived, message got read on another client).

The Message Storage Client SHALL act as an IMAP4 client as defined in [RFC3501]. In addition to that, the Message Storage Client SHALL support the “ACL” IMAP4 extension as defined in [RFC4314], the “URLAUTH” IMAP4 extension as defined in [RFC4467], the “CONSDSTORE” IMAP4 extension as defined in [RFC4551], the “ENABLE” IMAP4 extension as defined in [RFC5161 RFC5161] and the “QRESYNC” IMAP4 extension as defined in [RFC5162]. Also, the Message Storage Client MAY support the “ANNOTATE” IMAP4 extension as defined in [RFC5257], the “CONVERT” IMAP4 extension as defined in [RFC5259] and/or the “METADATA” IMAP4 extension as defined in [RFC5464].

6.1 General Operations

6.1.1 Authenticate Operation

When the CPM Client needs to authenticate with the Message Storage Server, it SHALL either use the SASL method or the plain-text username/password method.

When the Message Storage Client wants to use the SASL method, the Message Storage Client SHALL send to the Message Storage Server an AUTHENTICATE request as defined in [RFC3501] with the authentication mechanism that it wants to use and then SHALL complete the authentication process as defined in [RFC3501].

When the Message Storage Client wants to use the plain-text username/password method, the Message Storage Client SHALL send to the Message Storage Server a LOGIN request as defined in [RFC3501] with the username and password associated with the Message Storage Client’s CPM User.

The CPM Client MAY set up a TLS session prior to authenticating with the Message Storage Server. In order to do so, the Message Storage Client SHALL send to the Message Storage Server a STARTTLS request as defined in [RFC3501] and complete the TLS session setup. The Message Storage Server SHALL support either TLS or PSK-TLS as defined in [RFC2246] and [OMA-SEC-CF] for the TLS protocol negotiations.

6.1.2 Set Active Folder Operation

When a Message Storage Client needs to set a particular folder as the active folder, the Message Storage Client SHALL send to the Message Storage Server a SELECT request as defined in [RFC3501] with the folder name of the folder that is to be set as the active folder.

6.2 Access Control List Operations

6.2.1 Set Access Control List

When a Message Storage Client needs to set the access rights for another Principal on one of the folders of its associated CPM User, the Message Storage Client SHALL send to the Message Storage Server a SETACL request as defined in [RFC4314] with the folder name, the granted access rights and the identifier of the Principal to which access is given.

6.2.2 Get Access Control List

When a Message Storage Client needs to get the access control list on one of the folders of its associated CPM User, the Message Storage Client SHALL send to the Message Storage Server a GETACL request as defined in [RFC4314] with the folder name.
6.2.3 Delete Access Control List

When a Message Storage Client needs to delete the access rights for another Principal on one of the folders of its associated CPM User, the Message Storage Client SHALL send to the Message Storage Server a DELETEACL request as defined in [RFC4314] with the folder name and the identifier of the Principal to which access is given.

6.2.4 Access Rights Retrieval Operations

When a Message Storage Client needs to retrieve the access rights for another Principal on one of the folders of its associated CPM User, the Message Storage Client SHALL send to the Message Storage Server a LISTRIGHTS request as defined in [RFC4314] with a folder name and the identifier of the Principal whose access rights are to be retrieved.

6.3 Message and History Operations

6.3.1 Object Store Operation

When a Message Storage Client needs to store a message object, a file transfer history object, a session 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 Server an APPEND request as defined in [RFC3501] including the name of the folder and the data of the message object, file transfer history object, session history object, or standalone Media Object data. The Message Storage Client MAY include an initial set of metadata flags in the APPEND request towards the Message Storage Server, as defined in [RFC3501]. The Message Storage Client also MAY include a set of metadata annotations in the APPEND request towards the Message Storage Server, as defined in [RFC5257].

NOTE: The set of metadata flags and metadata annotations associated with the stored object can be changed later using the metadata update operation defined in section Error! Reference source not found. “Error! Reference source not found.”.

6.3.2 Object Fetch Operation

When a Message Storage Client needs to fetch a message object, file transfer history object, session history object or standalone Media Object from the active folder on the Message Storage Server, the Message Storage Client SHALL send to the Message Storage Server a FETCH or a UID FETCH request as defined in [RFC3501] with the UID(s) pointing to a stored object(s) in Message Storage Server.

NOTE: The Message Storage Client may specify that only specific parts of a message object, file transfer history object, session history object or standalone Media Object are to be fetched, as defined in [RFC3501].

6.3.3 Object Preview Fetch Operation

When a Message Storage Client needs to fetch a preview of a message object, file transfer history object, session history object or standalone Media Object from the active folder on the Message Storage Server, the Message Storage Client SHALL send to the Message Storage Server a CONVERT or a UID CONVERT request as defined in [RFC5259] with the UID pointing to the stored object on the Message Storage Server to be previewed.

NOTE 1: The Message Storage Client may specify that only a preview of specific parts of a message object, file transfer history object, session history object or standalone Media Object is to be fetched, as defined in [RFC3501] and [RFC5259].

NOTE 2: The CONVERT and UID CONVERT commands can be used to transcode the media type of a MIME part into another media type and/or into the same media type with different encoding parameters.

NOTE 3: A “Preview Fetch” operation MAY involve a server-side content adaptation in response to the Message Storage Client’s request for the stored object in a compacted or digested form rather than in its original full size and shape.
NOTE 4: Conversions only affect what is sent to the Message Storage Client; the original data in the Message Storage Server SHALL NOT be converted.

6.3.4 Object Copy Operation

When a Message Storage Client needs to copy an existing message object, an existing file transfer history object, an existing session history object or an existing standalone Media Object in the Message Storage Server, the Message Storage Client SHALL send to the Message Storage Server a COPY request as defined in [RFC3501] with the UID pointing to an object to be copied from the Message Storage Server and specifying the destination folder.

6.3.5 Object Remove Operation

When a Message Storage Client needs to remove a stored message object, a file transfer history object, a stored session history object or a stored standalone Media Object, the Message Storage Client SHALL send to the Message Storage Server a STORE request as defined in [RFC3501] with the UID pointing to the stored object to update the flag list associated with the object’s data and setting the “\Deleted” flag.

After setting the “\Deleted” flag, the Message Storage Client SHALL send to the Message Storage Server an EXPUNGE request as defined in [RFC3501] in order to permanently remove the message(s) that have been identified for removal from the list of objects with the “\Deleted” flag set.

NOTE: The Message Storage Client can use the EXPUNGE request to permanently remove multiple messages, i.e. there may be multiple STORE commands to set the “Deleted” flag before an EXPUNGE command is executed.

6.4 Folder Operations

6.4.1 Folder Create Operation

When a Message Storage Client needs to create a new folder or a new conversation history object, the Message Storage Client SHALL send to the Message Storage Server a CREATE request as defined in [RFC3501] including the name of the folder.

6.4.2 List Folder Operation

When a Message Storage Client needs to list the contents of the currently selected folder, or conversation history object, the Message Storage Client SHALL send to the Message Storage Server a LIST request as defined in [RFC3501].

6.4.3 Folder Move Operation

NOTE: The folder move operation is also used for renaming folders.

When a Message Storage Client needs to rename a folder or a conversation history object or to move a folder or a conversation history object, the Message Storage Client SHALL send to the Message Storage Server a RENAME request as defined in [RFC3501] including the old name and the new name of the folder or conversation history object to be renamed or moved.

6.4.4 Folder Remove Operation

When a Message Storage Client needs to delete a folder or a conversation history object, the Message Storage Client SHALL send to the Message Storage Server a DELETE request as defined in [RFC3501] including the name of the folder or conversation history object that is to be deleted.

6.4.5 Folder Search Operation

When a CPM Client needs to search in the active folder on the Message Storage Server, the Message Storage Client SHALL send to the Message Storage Server SEARCH request as defined in [RFC3501] including one or more search key data.
6.5 Reference Operations

6.5.1 Generate Reference Operation

When a Message Storage Client needs to generate a reference for (part of) a message object, a file transfer history object, a session history object or a stand-alone Media Object, the Message Storage Client SHALL send to the Message Storage Server a GENURLAUTH request as defined in [RFC4467] including an IMAP URL (as per [RFC5092] and [RFC4467]) pointing to the (part of) the object for which a reference needs to be created.

6.5.2 Fetch by Reference Operation

When a Message Storage Client needs to fetch the (part of) a message object, a file transfer history object, a session history object or a stand-alone media object on the basis of the reference, the Message Storage Client SHALL send to the Message Storage Server a URLFETCH request as defined in [RFC4467].

6.6 Message and History Metadata Management Operations

6.6.1 Metadata Operation

The Message Storage Client SHALL support updating the metadata flags defined for IMAP4 message objects in section Error! Reference source not found. “Error! Reference source not found.” and MAY support updating the metadata annotations defined for IMAP4 mailboxes and IMAP4 message objects in section Error! Reference source not found. “Error! Reference source not found.”.

When a Message Storage Client needs to update the meta data flags of a message object, a file transfer history object, a session history object or a standalone Media Object, the Message Storage Client SHALL send to the Message Storage Server a STORE request as defined in [RFC3501] including the UID of the message object, file transfer history object, session history object or standalone Media Object and the changes to the flags (e.g., seen flag, read flag.)

Additionally, the CPM Message Storage Client MAY use the ANNOTATE extensions as defined in [RFC5257] for the STORE request to provide metadata annotations update.

When a CPM Client needs to update the meta annotations of a folder object or a conversation history object, the Message Storage Client SHALL send to the Message Storage Server a SETMETADATA request as defined in [RFC5464] with the name of the folder and the updated metadata annotations.

6.7 Synchronization

[OMA-CPM-SD] gives a description of the synchronization process between the Message Storage Client and the Message Storage Server. This process only consists of operations described above and therefore needs no further explanation in this section.

While executing these operations, the Message Storage Client SHALL support and use the IMAP4 extensions described in [RFC4551], [RFC5161] and [RFC5162] to get an optimized and quick synchronization between the – potentially offline – Message Storage Client and the Message Storage Server.

6.8 Notification Operations

The IMAP Extension [RFC5465] allows the Message Storage Client to request for solicited notifications about events in specified folders of a Message Storage Server. For making this request, the Message Storage Client SHALL send a NOTIFY command to the Message Storage Server to limit its unsolicited notifications to certain selected folders and certain events such as objects being added to or deleted from those selected folders. Without this Message Storage Client NOTIFY command, an IMAP server will only send information about the changes in the Message Storage Server to the client in the following cases:

1. as the result of a Message Storage Client command such as FETCH responses to a FETCH or STORE command),
2. as unsolicited responses sent just before the end of a command (e.g., EXISTS or EXPUNGE) as the result of changes in other sessions, and

3. during the Message Storage Client’s IDLE command

NOTE: Per [RFC2177], the IDLE command provides a way for the client to go into a mode where the Message Storage Server pushes its notifications about the server events in selected folders.

Upon registration and receiving of either solicited or unsolicited notifications from the Message Storage Server, the Message Storage Client SHALL update its corresponding information in the client(s)’ locally stored resources.
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 an IMAP4 server as defined in [RFC3501]. In addition to that, the Message Storage Server SHALL support the “ACL” IMAP4 extension as defined in [RFC4314], the “URLAUTH” IMAP4 extension as defined in [RFC4467], the “CONDSTORE” IMAP4 extension as defined in [RFC4551], the “ENABLE” IMAP4 extension as defined in [RFC5161] and the “QRESYNC” IMAP4 extension as defined in [RFC5162]. Also, the Message Storage Server MAY support the “CONVERT” IMAP4 extension as defined in [RFC5259] and/or the “METADATA” IMAP4 extension as defined in [RFC5464].

7.1 General Operations

7.1.1 Authenticate Operation

Upon receiving a STARTTLS request, the Message Storage Server SHALL process the request and return a response according to the STARTTLS command as defined in [RFC3501]. The Message Storage Server SHALL at least support TLS and PSK-TLS as defined in [RFC2246] and [OMA-SEC-CF] for the TLS protocol negotiations.

Upon receiving an AUTHENTICATE request, the Message Storage Server SHALL process the request and return a response according to the AUTHENTICATE command as defined in [RFC3501] and then complete the authentication process as defined in [RFC3501].

Upon receiving a LOGIN request, the Message Storage Server SHALL process the request and return a response according to the LOGIN command as defined in [RFC3501].

7.1.2 Set Active Folder Operation

Upon receiving a SELECT request, the Message Storage Server SHALL process the request and return a response according to the SELECT command as defined in [RFC3501].

7.2 Access Control List Operations

7.2.1 Set Access Control List

Upon receiving a SETACL request, the Message Storage Server SHALL process the request and return a response according to the SETACL command as defined in [RFC4314].

7.2.2 Get Access Control List

Upon receiving a GETACL request, the Message Storage Server SHALL process the request and return a response according to the GETACL command as defined in [RFC4314].

7.2.3 Delete Access Control List

Upon receiving a DELETEACL request, the Message Storage Server SHALL process the request and return a response according to the DELETEACL command as defined in [RFC4314].

7.2.4 Access Rights Retrieval Operations

Upon receiving a LISTRIGHTS request the Message Storage Server SHALL process the request and return response according to the descriptions as defined in [RFC4314].
7.3 Message and History Operations

7.3.1 Object Store Operation

Upon receiving an APPEND request including the object to store, the Message Storage Server SHALL handle the request according to the APPEND command as defined in [RFC3501] and store the specified object to the end of the specified destination folder.

7.3.1.1 Handling Deferred CPM Message Objects

Upon receiving a request for storing a Deferred CPM Message object that is associated with an expiry time, the Message Storage Server SHALL:

1. handle the request according to the APPEND command as defined in [RFC3501] and store the object to the end of the specified destination folder designated for temporarily holding deferred messages and
2. notify the Message Storage Client of the arrival of the new message.

The stored message object will remain in the designated folder until either it is moved by the Message Storage Client’s corresponding commands or it reaches its deferred expiry time and is deleted from the folder.

7.3.2 Object Fetch Operation

Upon receiving a FETCH request with the UID pointing to a stored object, the Message Storage Server SHALL handle the request according to the FETCH or UID FETCH command as defined in [RFC3501].

7.3.3 Object Preview Fetch Operation

Upon receiving a CONVERT or UID CONVERT request with the UID pointing to a stored object of the Message Storage Server, the Message Storage Server SHALL handle the preview request for the supplied format and dimensions according to the CONVERT or UID CONVERT command as defined in [RFC5259]

7.3.4 Object Copy Operation

Upon receiving a COPY request with the UID pointing to a stored object of the Message Storage Server, the Message Storage Server SHALL copy the specified Message(s) or Message History to the end of the specified destination folder according to the COPY command as defined in [RFC3501]

7.3.5 Object Remove Operation

Upon receiving an IMAP STORE command with a UID pointing to the stored object to update the flag list of the object’s data to include the “\Deleted” flag, the Message Storage Server SHALL handle the flagging request by setting the stored object’s “\Deleted” flag according to the STORE command as defined in [RFC3501].

Upon receiving an EXPUNGE request to remove all stored objects from the CPM User’s Message Storage Server that have the “\Deleted” flag set for removal, the Message Storage Server SHALL handle the request to permanently remove these stored objects according to the EXPUNGE command as defined in [RFC3501].

7.4 Folder Operations

7.4.1 Folder Create Operation

Upon receiving a CREATE request, the Message Storage Server SHALL create the folder (e.g., mailbox) with the requested name according to the CREATE command as defined in [RFC3501].
7.4.2 List Folders Operation

Upon receiving a LIST request, the Message Storage Server SHALL determine and return the names of all folders of the CPM User on the Message Storage Server according to the LIST command as defined in [RFC3501].

7.4.3 Folder Move Operation

NOTE: The folder move operation is also used for renaming folders.

Upon receiving a RENAME request with the request-folder name and its new name, the Message Storage Server SHALL rename the indicated folder according to the RENAME command as defined in [RFC3501].

7.4.4 Folder Remove Operation

Upon receiving a DELETE request with the request-folder name, the Message Storage Server SHALL remove the indicated folder according to the DELETE command as defined in [RFC3501].

7.4.5 Folder Search Operation

Upon receiving a SEARCH request, with one or more search key data, the Message Storage Server SHALL handle the request according to the IMAP SEARCH command as defined in [RFC3501].

7.5 Reference Operations

7.5.1 Generate Reference Operation

Upon receiving a GENURLAUTH request, the Message Storage Server SHALL process the request and return a response according to the GENURLAUTH command as defined in [RFC4467].

7.5.2 Fetch by Reference Operation

Upon receiving a URLFETCH request, the Message Storage Server SHALL process the request and return a response according to the URLFETCH command as defined in [RFC4467].

7.6 Metadata Update Operation

The Message Storage Server SHALL support the metadata flags defined for IMAP4 message objects in section Error! Reference source not found. “Error! Reference source not found.” and MAY support the metadata annotations defined for IMAP4 mailboxes and IMAP4 message objects in section Error! Reference source not found. “Error! Reference source not found.”.

Upon receiving a STORE request with the UID addressing a stored object of the Message Storage Server, the Message Storage Server SHALL update the metadata flags as indicated in the request according to the STORE command as defined in [RFC3501]. The Message Storage Server MAY handle any metadata annotations specified in the STORE request according to the ANNOTATE IMAP4 extension defined in [RFC5257].

Upon receiving a SETMETADATA request with the UID addressing a stored object of the Message Storage Server, the Message Storage Server SHALL update the metadata annotations for the indicated updates (add, remove & modify) according to the SETMETADATA command as defined in [RFC5464]

7.7 Message and History Synchronization Operations

[OMA-CPM-SD] gives a description of the synchronization process between the Message Storage Client and the Message Storage Server. This process only consists of operations described above and therefore needs no further explanation in this section.
In addition to that, the Message Storage Server SHALL support the IMAP4 extensions described in [RFC4551], [RFC5161] and [RFC5162] to allow the Message Storage Client to have an optimized and quick synchronization process.

7.8 Notifications Operations

Upon receiving the Message Storage Client’s NOTIFY command and/or occurrence of changes in the Message Storage Server’s resources, the Message Storage Server SHALL reflect the presence of changes via sending Notification messages. The changes in the Message Storage Server’s resources MAY include one or more of the following specific operations per [RFC5423]:

1. message addition and deletion
2. Message flags, e.g., read, clear
3. Access accounting, e.g., login, logout
4. Folder management, e.g., create, delete, rename

The notification messages SHALL be sent by the Message Storage Server to the Message Storage Client upon the occurrence of the following events:

1. Changes in the stored objects due to performing any operations by the Message Storage Server on the stored objects during the period when the CPM Message Storage Client is registered and the notification feature is activated.
2. Changes occurred since the CPM Message Storage Client’s last de-registration or CPM User’s deactivation of the notification feature.

NOTE: While IMAP IDLE, NOTIFY, or other status change indications serve the purpose of notification, they are not necessarily for notifications only. They are part of the greater synchronization mechanism.
Appendix A. Change History

A.1 Approved Version 1.0 History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA-TS-CPM_MessageStorage-V1.0-20120612-A</td>
<td>12 Jun 2012</td>
<td>Status changed to Approved by TP:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OMA-TP-2012-0221R01-INP_CPM_V1.0_ERP_for_Final_Approval</td>
</tr>
</tbody>
</table>
Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [Error! Reference source not found.].

### B.1 SCR for Message Storage Client

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPM-TS-MS-C-001_M</td>
<td>Authentication – setting up TLS connection</td>
<td>6.1.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-002_M</td>
<td>Authentication – SASL method</td>
<td>6.1.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-003_M</td>
<td>Authentication – username/password method</td>
<td>6.1.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-004_M</td>
<td>Requesting to set an active folder</td>
<td>6.1.2</td>
</tr>
<tr>
<td>CPM-TS-MS-C-005_M</td>
<td>Setting an ACL</td>
<td>6.2.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-006_M</td>
<td>Getting an ACL</td>
<td>6.2.2</td>
</tr>
<tr>
<td>CPM-TS-MS-C-007_M</td>
<td>Deleting an ACL</td>
<td>6.2.3</td>
</tr>
<tr>
<td>CPM-TS-MS-C-008_M</td>
<td>Retrieving access rights</td>
<td>6.2.4</td>
</tr>
<tr>
<td>CPM-TS-MS-C-009_M</td>
<td>Requesting to store an object(i.e. Message and Message histories)</td>
<td>6.3.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-010_M</td>
<td>Requesting to fetch message(s)</td>
<td>6.3.2</td>
</tr>
<tr>
<td>CPM-TS-MS-C-011_O</td>
<td>Requesting to fetch a preview object</td>
<td>6.3.3</td>
</tr>
<tr>
<td>CPM-TS-MS-C-012_M</td>
<td>Requesting to copy objects to another folder</td>
<td>6.3.4</td>
</tr>
<tr>
<td>CPM-TS-MS-C-013_M</td>
<td>Requesting to remove objects</td>
<td>6.3.5</td>
</tr>
<tr>
<td>CPM-TS-MS-C-014_M</td>
<td>Requesting to create a folder</td>
<td>6.4.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-015_M</td>
<td>Requesting to get a list of folders</td>
<td>6.4.2</td>
</tr>
<tr>
<td>CPM-TS-MS-C-016_M</td>
<td>Requesting to rename the folder</td>
<td>6.4.3</td>
</tr>
<tr>
<td>CPM-TS-MS-C-017_M</td>
<td>Requesting to remove the folder</td>
<td>6.4.4</td>
</tr>
<tr>
<td>CPM-TS-MS-C-018_M</td>
<td>Requesting to search contents by a few key words</td>
<td>6.4.5</td>
</tr>
<tr>
<td>CPM-TS-MS-C-019_M</td>
<td>Updating metadata flags of a stored object</td>
<td>6.5.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-020_O</td>
<td>Updating metadata annotations of a stored object (ANNOTATE extension)</td>
<td>6.5.1</td>
</tr>
<tr>
<td>CPM-TS-MS-C-021_O</td>
<td>Updating metadata annotations of a folder</td>
<td>6.6</td>
</tr>
<tr>
<td>Item</td>
<td>Function</td>
<td>Reference</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>CPM-TS-MS-C-022_M</td>
<td>(METADATA extension)</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-C-023_O</td>
<td>Notifications</td>
<td>6.7</td>
</tr>
</tbody>
</table>

### B.2 SCR for Message Storage Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPM-TS-MS-S-001_M</td>
<td>Authentication – setting up TLS connection</td>
<td>7.1.1</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-002_M</td>
<td>Authentication – SASL method</td>
<td>7.1.1</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-003_M</td>
<td>Authentication – username/password method</td>
<td>7.1.1</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-004_M</td>
<td>Requesting to set an active folder</td>
<td>7.1.2</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-005_M</td>
<td>Setting an ACL</td>
<td>7.2.1</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-006_M</td>
<td>Getting an ACL</td>
<td>7.2.2</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-007_M</td>
<td>Deleting an ACL</td>
<td>7.2.3</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-008_M</td>
<td>Retrieving access rights</td>
<td>7.2.4</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-009_M</td>
<td>Storing objects to a folder</td>
<td>7.3.1</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-010_M</td>
<td>Sending objects</td>
<td>7.3.2</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-011_O</td>
<td>Converting the object as requested</td>
<td>7.3.3</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-012_M</td>
<td>Copying the objects to the another folder</td>
<td>7.3.4</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-013_M</td>
<td>Removing the object</td>
<td>7.3.5</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-014_M</td>
<td>Creating a folder with a requested folder name</td>
<td>7.4.1</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-015_M</td>
<td>Sending the name of all folders</td>
<td>7.4.2</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-016_M</td>
<td>Renaming the folder</td>
<td>7.4.3</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-017_M</td>
<td>Deleting the folder</td>
<td>7.4.4</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-018_M</td>
<td>Searching objects by a few keywords</td>
<td>7.4.5</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-019_M</td>
<td>Updating metadata flags for a stored object</td>
<td>7.5</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-020_O</td>
<td>Updating metadata annotations of a stored object (ANNOTATE extension)</td>
<td>6.2.3</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-021_O</td>
<td>Updating metadata annotations of a folder (METADATA extension)</td>
<td>7.5</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-022_M</td>
<td>Processing the synchronization</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>CPM-TS-MS-S-023_O</td>
<td>Notifications</td>
<td>7.7</td>
<td></td>
</tr>
</tbody>
</table>
Appendix C. Appendix C. CPM-defined IMAP Flag Extensions

C.1 \read-report-sent

\read-report-sent is a CPM-defined IMAP flag extension associated with a Message Object stored in Message Storage Server.

The definition of the flag is:
\begin{verbatim}
\read-report-sent
Read report is sent
\end{verbatim}

Formal syntax of \read-report-sent is:
\begin{verbatim}
flag-extension = "\read-report-sent"
\end{verbatim}

Example:
\begin{verbatim}
C: A003 STORE 2:4 +FLAGS (\read-report-sent)
\end{verbatim}

\read-report-sent status flag is visible to the Message Storage Client but is masked to the CPM User. This flag is used to record whether a read report has been sent or not for a message object in the Message Storage Server.
Appendix D.  Example of Session History Object

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

- 1 CPM Chat Message.
- 1 audio stream.

From: John Doe <jdoe@machine.example>
Date: Fri, 21 Nov 1997 09:55:06 -0600
Subject: the weather will be fine today
Conversation-ID: f81d4fae-7dec-11d0-a765-00a0c91e6bf6
Contribution-ID: abcd0ef-2345-7689-0ab-cdef01234567
InReplyTo-Conversion-ID: 01234567-89ab-cdef-0123-456789abcdef
Content-type: multipart/related; boundary=cpm; type="Application/X-CPM-Session"

--cpm
Content-Type: Application/X-CPM-Session

<session-type>1-1</session-type>
<invited-participants>jdoe@machine.example; sip:alice@example.com</invited-participants>

<media-object>
  <cid>cid:<1234@example.com></cid>
  <sdp>
    a=sendonly
    m=audio 49170 RTP/AVP 0 97
  </sdp>
</media-object>

--cpm
Content-type: Message/CPIM
From: John Doe <jdoe@machine.example>
To: Alice <sip:alice@example.com>
DateTime: 2000-12-13T13:40:00-08:00
Content-type: text/plain; charset=utf-8

Here is the text of my message.

--cpm
Content-Type: audio/basic
Content-Transfer-Encoding: base64
Content-ID:<1234@example.com>

    ... base64-encoded 8000 Hz single-channel
    mu-law-format audio data goes here ...
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.

From: John Doe <jdoe@machine.example>
Date: Fri, 21 Nov 1997 09:55:06 -0600
Conversation-ID: f81d4fae-7dec-11d0-a765-00a0c91e6bf6
Contribution-ID: abcdef-1234-5678-90ab-cdef01234567
InReplyTo-Contribution-ID: 01234567-89ab-cdef-0123-456789abcdef
Content-type: multipart/related;boundary=cpm; type="Application/X-CPM-File-Transfer"

--cpm
Content-Type: Application/X-CPM-File-Transfer

<file-transfer-type>1-1</file-transfer-type>
<invited-participants>jdoe@machine.example; sip:alice@example.com</invited-participants>

<file-object>
  <cid>cid:<1234@example.com></cid>
  <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"
  </sdp>
</file-object>

--cpm
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <1234@example.com>

... My picture.jpg...