



Technical Specifications for Mobile Email

Approved Version 1.0 – 02 Aug 2011

Open Mobile Alliance
OMA-TS-Mobile_Email-V1_0-20110802-A

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

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

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

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

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

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

© 2011 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	9
5.	ARCHITECTURE OVERVIEW	10
6.	OUT-BAND NOTIFICATIONS	11
6.1	EMAIL EVENT TYPES	11
6.2	PAYLOAD	11
6.3	NOTIFICATION MECHANISMS	11
6.3.1	SMS	12
6.4	NOTIFICATION PREFERENCES	12
6.4.1	Client-ID	12
6.4.2	Subscribed Email Event Types	13
6.4.3	Preferences for Notification Mechanisms	13
7.	CLIENT CAPABILITY INFORMATION	14
8.	CLIENT CONFIGURATION	15
8.1	PROVISIONING	15
9.	CONTENT TRANSCODING	16
9.1	GENERIC UNICODE FOR HEADER FIELDS	16
9.2	REQUIRED CONVERSIONS	16
9.2.1	Character Set Conversions	16
9.2.2	Image Conversions	16
9.3	OPTIONAL CONVERSIONS	16
9.3.1	Audio Conversions	16
9.3.2	Video Conversions	17
9.3.3	Text Conversions	17
9.3.4	Image-to-Text Conversions	17
10.	BINDINGS FOR MEM PROTOCOL	18
10.1	IETF LEMONADE	18
10.1.1	MEM Protocol	18
10.1.2	MEM-1 Interface	19
10.1.3	MEM-2 Interface	19
10.1.4	Content Transcoding	19
10.1.5	Out-band Notifications	22
10.2	OMA-DS	23
10.2.1	Component Mapping	23
10.2.2	MEM Protocol	23
10.2.3	MEM-1 Interface	23
10.2.4	MEM-2 Interface	23
10.2.5	Out-band Notifications	23
APPENDIX A.	CHANGE HISTORY (INFORMATIVE)	24
A.1	APPROVED VERSION HISTORY	24

APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....25

B.1 SCR FOR MEM CLIENT25

B.2 SCR FOR MEM SERVER.....25

APPENDIX C. ABNF DEFINITIONS27

C.1 ABNF RULES27

C.2 ABNF TERMINAL VALUES.....27

Figures

Figure 1 - Logical architecture for OMA MEM enabler.....10

Tables

Table 1 – OMA DS Enabler component mapping23

1. Scope

This document defines the technical specification for the OMA Mobile Email Enabler to ensure an inter-operable solution.

The goal of the MEM Enabler is to provide quasi-instantaneous and secure updates of the MEM Client with new emails and MEM server changes, optimized online and off-line usage and capability to securely send email from the appropriate MEM server.

While mobile email is defined in the requirement document [MEM RD] and the logical architecture is described in architecture document [MEM AD], this document defines technical specification for the fundamental events and notification mechanisms, management interfaces, client capabilities, client configuration, content transcoding, binding of IETF Lemonade based and OMA DS based MEM protocols.

Management Object and charging for the OMA MEM Enabler are defined respectively in two separate technical specifications [MEM MO] [MEM Charging].

2. References

2.1 Normative References

- [E.164] “ITU-T Recommendation E.164”, “The international public telecommunication numbering plan”, Telecommunication Standardization Sector of ITU, February 2005.
URL: <http://www.itu.int/rec/T-REC-E.164-200502-I/en>
- [DS Enabler] “OMA Data Synchronization”, Version 1.2.1, Open Mobile Alliance™,
OMA-ERP-DS-V1_2_1-20070810-A
URL: <http://www.openmobilealliance.org>
- [IMAP-FILTERS] “IMAP4 extension for named searches (filters)”, A. Melnikov and C. King,
URL: <http://www.ietf.org/internet-drafts/draft-melnikov-imapext-filters>
- [IOPPROC] “OMA Interoperability Policy and Process”, Version 1.1, Open Mobile Alliance™,
OMA-IOP-Process-V1_1,
URL: <http://www.openmobilealliance.org>
- [Lemonade Profile bis] “Lemonade Profile bis”, Stephane H. Maes, Alexey Melnikov and Dave Cridland,
URL: <http://www.ietf.org/internet-drafts/draft-ietf-lemonade-profile-bis>
- [MANAGESIEVE] “A Protocol for Remotely Managing Sieve Scripts”, A. Melnikov and T. Martin,
<http://www.ietf.org/internet-drafts/draft-ietf-sieve-managesieve>
- [MEM AD] “Mobile Email Architecture”, Version 1.1, Open Mobile Alliance™,
OMA-AD-Mobile_Email-V1_0_0-20070614-D
URL: <http://www.openmobilealliance.org>
- [MEM Charging] “OMA MEM Charging Specification”, Version 1.0, Open Mobile Alliance™,
OMA-TS-MEM_Charging-V1_0_0-20081007-D
URL: <http://www.openmobilealliance.org>
- [MEM MO] “Mobile Email Management Object”, Version 1.0, Open Mobile Alliance™,
OMA-TS-MEM_MO-V1_0_0-20080603-D,
URL: <http://www.openmobilealliance.org>
- [MEM RD] “Mobile Email Requirements”, Version 1.0, Open Mobile Alliance™,
OMA-RD-MobileEmail-V1_0-20051018-C
URL: <http://www.openmobilealliance.org>
- [METADATA] “IMAP METADATA Extension”, C. Daboo,
<http://www.ietf.org/internet-drafts/draft-daboo-imap-annotatemo>
- [Push OTA AD] “Push Architecture”, Version 2.2, Open Mobile Alliance™,
OMA-AD-Push-V2_2-20071002-C
URL: <http://www.openmobilealliance.org>
- [Push OTA TS] “Push Over The Air”, Version 2.2, Open Mobile Alliance™,
OMA-TS-PushOTA-V2_2-20071002-C
URL: <http://www.openmobilealliance.org>
- [RFC2015] “MIME Security with Pretty Good Privacy (PGP)”, M. Elkins, October 1996,
URL: <http://www.ietf.org/rfc/rfc2015.txt>

- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997,
URL: <http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2633] “S/MIME Version 3 Message Specification”, B. Ramsdell, June 1999,
URL: <http://www.ietf.org/rfc/rfc2633.txt>
- [RFC3685] “SIEVE Email Filtering: Spamtest and VirusTest Extensions”, C. Daboo, February 2004,
URL: <http://www.ietf.org/rfc/rfc3685.txt>
- [RFC3840] “Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)”, J. Rosenberg, H. Schulzrinne, P. Kyzivat, August 2004,
URL: <http://www.ietf.org/rfc/rfc3840.txt>
- [RFC3986] “Uniform Resource Identifier (URI): Generic Syntax”, T. Berners-Lee, R. Fielding, L. Masinter, January 2005,
URL: <http://www.ietf.org/rfc/rfc3986.txt>
- [RFC5234] “Augmented BNF for Syntax Specifications: ABNF”, Dave Crocker and Paul Overell,
January 2008,
URL: <http://www.ietf.org/rfc/rfc5234.txt>
- [RFC5259] “Internet Message Access Protocol - CONVERT Extension”, A. Melnikov, P. Coates, July
2008,
URL: <http://www.ietf.org/rfc/rfc5259.txt>
- [SIEVE-METADATA] “The SIEVE mail filtering language - extension for accessing mailbox metadata”, A.
Melnikov,
<http://www.ietf.org/internet-drafts/draft-melnikov-sieve-imapext-metadata>
- [SIP Push AD] “Push using SIP Architecture”, Version 1.0, Open Mobile Alliance™,
OMA-AD-SIP_Push-V1_0-20071203-D
URL: <http://www.openmobilealliance.org>
- [SIP Push TS] “Push using SIP”, Version 1.0, Open Mobile Alliance™,
OMA-TS-SIP_Push-V1_0-20080108-D
URL: <http://www.openmobilealliance.org>
- [OMA-DSPROTO] “Data Sync Protocol, Version 1.2.1”, Open Mobile Alliance™.
OMA-TS-DS_DS_Protocol-V1_2_1. [URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [OMA-SCOMO] “Software Component Management Object”, Version 1.0, Open Mobile Alliance™,
OMA-TS-SCOMO-V1_0,
URL: <http://www.openmobilealliance.org>
- [Store-Events] “Internet Message Store Events”, Chris Newman, Randall Gellens,
URL: <http://tools.ietf.org/html/draft-ietf-lemonade-msgevent>
- [UNICODE] “The Unicode Standard, Version 5.0”, The Unicode Consortium. The Unicode Standard,
Version 5.0.0, defined by: The Unicode Standard, Version 5.0
Boston, MA, Addison-Wesley, 2007. ISBN 0-321-48091-0
URL: <http://www.unicode.org/versions/Unicode5.0.0>

2.2 Informative References

This section left intentionally blank.

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

This technical specification refers other, external specifications. The external definitions are not copied here – the reader shall refer to the original definitions in the referenced documents.

Email Event	Change to the status of an email (e.g. read/unread, moved, flagged, deleted, etc...) that results, for example, from performing an operation on an email message. The event may be considered as server or client side events depending on where the change took place. A new email is also considered as an event.
MEM Alignment	The process and mechanisms by which the MEM Client is updated to an appropriate view of the MEM Server and the MEM Server is updated to an appropriate view of the MEM Client where appropriate view means a subset of the corresponding data filtered based on configuration and user preferences.
NIL	Special marker used by the MEM client to specify as target MIME type, therefore requesting the MEM server to pick a suitable target media type, in accordance with procedures in [RFC 5259], instead of specifying the exact target MIME media type the client wants to convert to.

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form, see [RFC5234]
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IP	Internet Protocol
ISDN	Integrated Service Digital Network
MDN	Mobile Directory Number
MSISDN	Mobile Station International ISDN Number
OMA	Open Mobile Alliance
OTA	Over The Air
SAS	Server Alerted Sync
SMTP	Simple Mail Transfer Protocol

4. Introduction

The Mobile Email (MEM) Enabler aims to support efficient access to email from a mobile device. Email may be personal email provided by an email service provider, or corporate email.

While mobile email is defined in the requirement document [MEM RD] and the logical architecture is described in architecture document [MEM AD], this document provides detailed technical specification of the mobile email service. This document also covers the IETF Lemonade and OMA DS bindings of MEM protocol.

5. Architecture Overview

The MEM logical architecture illustrated in Figure 1 is composed by some components and protocol.

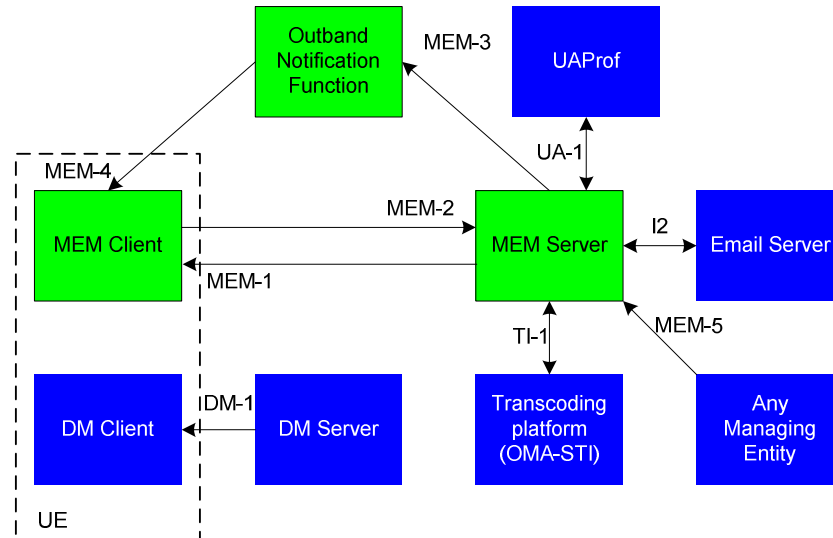


Figure 1 - Logical architecture for OMA MEM enabler.

MEM Client: Embedded Client for devices that enable synchronization of email using MEM protocol. The client also supports notification.

MEM Server: synchronization server supporting MEM protocol that enables direct synchronization with the Email Server. MEM Server supports filtering, notification, email.

Outband Notification Enabler: allows the notification of the MEM Client that an event has occurred at the Email Server via channels other than the MEM Protocol such as: SMS, MMS, WAP Push, SIP Push, etc

Others Enablers: stand for any enabler utilized by the MEM Server or the MEM Client to support additional MEM functionality, e.g. outband notifications, provisioning/device management, transcoding etc

MEM Protocol: allows the exchange of messages between the MEM Client and MEM Server that includes control of mobile operations, notifications, etc.

Email server: provides the user email data storage access (Mail Store) and means of email submission (Message Transfer Agent).

Any Managing Entity: management of MEM Server settings (globally and per account).

6. Out-band Notifications

While the MEM Client is not connected to the MEM Server (e.g. offline), the MEM Server informs the MEM Client about Email Events using out-band notifications. This is achieved by sending a specific payload to the MEM Client using a particular notification mechanism according to the notification preferences of the MEM Client. The Email Events are described in section 6.1; the payload is described in section 6.2; the notification mechanisms are described in section 6.3; the notification preferences are described in section 6.4.

The MEM Server **MUST** use the notification preferences [6.4] to identify the list of MEM Clients that are candidates for a out-band notification, and refine this list based on the following requirements (the order is not significant):

The MEM Server **MUST** refine the list of candidates based on the online state of the MEM Clients:

- The MEM Server **MUST NOT** send out-band notifications to online MEM Clients.
- The MEM Server **MUST** send out-band notifications to offline MEM Clients.

The MEM Server **MUST** refine the list of candidates using the type of the Email Event:

- The MEM Server **MUST NOT** send out-band notifications to MEM Clients that are not interested in the Email Event.
- The MEM Server **MUST** send out-band notifications to MEM Clients that are interested in the Email Event.

Once list of the MEM Clients to be notified have been finalized, the MEM Server **MUST** create the notification payload [6.2] and send it to each MEM Client on the list, using the appropriate notification mechanism individually selected for each MEM Client. The MEM Server **MUST** take into account the characteristics and the capabilities of the notification mechanism when it is creating the payload.

6.1 Email Event Types

We differentiate client-side and server-side Email Events based on the origin of the Email Event; however out-band notifications deal with server-side Email Events only. A single Email Event is supported, as follows:

MessageNew: This allows a MEM Server to send new mail notifications to an offline MEM Client, requesting it to connect, receive new email messages, and synchronize with the MEM Server.

6.2 Payload

The payload is basically a description of the Email Event that took place. The payload is not limited to describing one Email Event; it **MAY** include two or more Email Event descriptions. To describe an Email Event, the following details **MUST** be included in the payload:

Mailbox identifier – to indicate where the Email Event took place.

Event type – to indicate what kind of event took place. See section 6.1.

Event-specific parameters – to provide more details of the event (when available). The event-specific parameters are as follows: [TBD]

The exact format of the payload is defined in sections 10.1 and 10.2 for Lemonade-based MEM and DS-based MEM, respectively.

6.3 Notification Mechanisms

The notification mechanism has only one purpose: to deliver the payload to the MEM Client.

The available notification mechanisms are described in the following sub-sections. The MEM Client **MAY** support any of these mechanisms, however in order to receive notifications, the MEM Client **MUST** store its notification preferences on the MEM Server. See notification preferences in section 6.4.

6.3.1 SMS

Upon receipt of new mail, the MEM Server SHALL retrieve the client identification that identifies the OMA MEM Client application. The MEM Server SHALL then alert the MEM Client on the device by sending an SMS message with the reference of the message that triggers the synchronization.

Upon receipt of the SMS message, the MEM Client SHALL check the sender number against specified SMS-Service-Number defined in [MEM MO] and configured by the MEM server. If the compared result is not equal, then the MEM Client SHALL ignore this SMS notification. Otherwise, the MEM Client shall process this SMS notification according to procedures in sections 10.1 and 10.2 for Lemonade-based MEM and DS-based MEM, respectively.

6.4 Notification Preferences

The notification preferences contain information about each MEM Client including the means of contacting them using various protocols whenever out-band notifications are to be sent. Each MEM Client maintains its own set of notification preferences. These preferences are stored within the MEM Server. Each set of preferences MUST include:

One and exactly one Client-ID that identifies the MEM Client. It MUST be unique within the scope of the user. See [6.4.1].

One and exactly one list of subscribed Email Event types. See [6.4.2].

One or more preferences for each notification mechanism that is supported by the MEM Client. See [6.4.3].

- The priority indicator that tells the MEM Server which protocol to try next if the previous attempt has failed. See [6.4.3.1].

The MEM Clients are responsible for updating their own preferences, except the ClientID, since that is persistent. In order to avoid conflicts, each MEM Client MUST use its own Client-ID to update their notification preferences.

The exact means to store the preferences is technology-specific. According to the [MEM AD], in order to facilitate a mobile email service, the MEM-1 and MEM-2 interfaces MUST be provided for the MEM Protocol. This specification offers service providers two alternatives for the MEM Protocol. Service providers should take into account that neither of these two technologies meets all the requirements defined in the [MEM RD].

6.4.1 Client-ID

The Client-ID identifies the MEM Client. It also allows for storing client-specific preferences at the MEM Server. The Client-ID MUST be unique.

If the MEM Client does not have a Client-ID, it MUST request a new one from the MEM Server during initial Email Service setup. If the MEM Client has a Client-ID, it MUST inform the MEM Server about it whenever an Email Session is established.

In order to avoid reserving multiple Client-IDs for the same MEM Client, the MEM Client MUST store the Client-ID persistently. The MEM Server MUST store the Client-ID persistently as well.

The Client-ID MUST be a free-form string containing only US-ASCII alpha-numeric characters and is case-sensitive. The syntax for a Client-ID is defined using ABNF [RFC5234] as follows:

```
Client-ID = 1*(ALPHA / DIGIT)
```

For example:

```
Client-ID=1234FEED5678BEEF
```

6.4.2 Subscribed Email Event Types

In order to instruct the MEM Server to filter the notifications, a preference needs to be set. This preference is the list of subscribed Email Event types. The list contains all allowed Email Event types that the MEM Client will receive, while the non-listed Email Events will be discarded by the MEM Server.

The MEM Server **MUST** filter the notifications based on the type of the Email Event that has occurred:

all notifications carrying an Email Event of a subscribed type **SHALL** be passed to the MEM Client, while

all notifications carrying an Email Event of a non-subscribed type **SHALL** be discarded.

The list of subscribed Email Events **MUST** be a non-case sensitive list of Email Event types [6.1] separated by a plus sign ('+'). An empty event list **MUST** contain the text "nil". The syntax for the SubscribedEventTypeList is defined using ABNF [RFC5234] as follows:

```
SubscribedEventTypeList = nil / EventType / EventList
nil = "nil"
EventType = 1*ALPHA
                ; Defined in Section 6.1 of this document
EventList = EventType 1*("+" EventType)
```

For example:

```
SubscribedEventTypeList=MessageNew
```

6.4.3 Preferences for Notification Mechanisms

The MEM Client **MAY** support more than one out-band notification mechanism. It is **REQUIRED** to define settings for each out-band notification mechanism that the MEM Client desires to use.

The list of notification mechanisms can be found in [6.3].

For example: SMS

6.4.3.1 Priority Indicator

The MEM Client **MAY** configure any number of outband notification mechanisms. Since it is a good idea to conserve resources on the MEM Server, it is desired to choose only one notification mechanism. In order to make such decision the MEM Server needs to know which notification mechanism is preferred by the MEM Client. This is achieved by assigning a specific priority indicator to each mechanism, and by doing so, setting up a ranked list.

The priority indicator **MUST** be a positive decimal integer number and the smaller the number, the higher the priority; therefore '1' indicates the highest possible priority. The syntax for the PriorityIndicator is defined using ABNF [RFC5234] as follows:

```
PriorityIndicator = 1*DIGIT
```

The priority indicator **MUST** be unique within the scope of all notification preferences of the MEM Client, meaning that it is not possible to assign the same priority twice.

Whenever a notification needs to be sent, the MEM Server **MUST** take into account the notification priority defined by the MEM Client and use the highest priority mechanism that is available.

For example:

```
PriorityIndicator=2
```

7. Client Capability Information

The Client capability information includes capability information for both the device and MEM Client..

The device capability information includes hardware platform and software platform, etc.

Hardware Platform: A collection of properties that adequately describe the hardware characteristics of the terminal device. This includes the type of device, model number, display size, input and output methods, etc.

Software Platform: A collection of attributes associated with the operating environment of the device. Attributes provide information on the operating system software, video and audio encoders supported by the device, client settings, and other user profile information (e.g., user's preference on language, content transcoding, etc.).

The MEM Client capability information includes MEM Client version, supported protocols, supported optional commands and extensions, etc.

After the Client updates its MEM client capability information, the MEM Client MAY publish the updated information to the MEM Server.

After the Client updates its device capability information, the Client MAY publish the updated information to the MEM Server. Alternatively, the MEM server MAY obtain the updated device information using UAProf enabler [OMA UAProf].

8. Client Configuration

8.1 Provisioning

The OMA MEM Enabler allows provisioning of the MEM Client settings using OMA Device Management. For that purpose, a MEM Management Object is defined in [MEM MO].

9. Content Transcoding

9.1 Generic Unicode for Header Fields

In order to guarantee the readability of the information, it is recommended that the MEM Client should request conversion of the email to generic Unicode (See [UNICODE]) on the MEM Server before it is downloaded to the MEM Client. All MEM Server implementations SHALL support this function. However, considering the overhead for such conversion, and that conversion to generic Unicode would break digital signatures and therefore not preserve the integrity of the information, users MAY set preferences on their MEM clients, requesting the MEM Server to transcode or not transcode email header (or/and body) .

The following standard operations SHALL be performed in the MEM server when the MEM Client requests to download the transcoded headers of an email.

- MEM Server transcodes the email headers from original character to generic Unicode;
- MEM Server sends the transcoded headers to the MEM Client;
- MEM Client displays the transcoded headers.

(Nevertheless whether the MEM Client displays the right content depends on the capability of the mobile device.)

9.2 Required Conversions

In addition to character set conversions for email headers defined in the previous section, the MEM Server SHALL support conversions specified in the following subsections.

9.2.1 Character Set Conversions

The MEM Server SHALL support character set conversions from iso-8859-1, iso-8859-2, iso-8859-3, iso-8859-4, iso-8859-5, iso-8859-6, iso-8859-7, iso-8859-8 and iso-8859-15 to utf-8 for text/plain, text/html and text/csv MIME body parts.

9.2.2 Image Conversions

The MEM Server SHALL support image conversions among the MIME types image/gif, image/jpeg and image/png Image conversions MUST support the following transcoding parameters:

- Size limit on the converted image
- Changes to the image dimensions (e.g., in pixels), with option to maintain aspect ratio when resizing.
- Color depth

9.3 Optional Conversions

9.3.1 Audio Conversions

The MEM Server MAY support conversion of audio/*.

If conversions among different audio/* MIME types are supported, the following transcoding parameters MUST be supported:

- Size limit on the resulting attachment
- Change of codec

- Change of bit rate, encoding method (variable bit rate/fixed bit rate), min-bit-rate (for variable bit rate encodings)

9.3.2 Video Conversions

The MEM Server MAY support conversion of video/*.

If conversions among different video/* MIME types are supported, the following transcoding parameters MUST be supported:

- Size limit on the resulting attachment
- Change of video dimensions
- Change of codec
- Change of bit rate, encoding method (variable bit rate/fixed bit rate), min-bit-rate (for variable bit rate encodings)
- Change of frame rate when converting to video

9.3.3 Text Conversions

The MEM Servers MAY support text/* conversions from non iso-8859-* character sets to utf-8, as well as conversions to other character sets

9.3.4 Image-to-Text Conversions

The MEM Servers MAY support extraction of information about image/* in textual format (conversion to text/plain), for example extracting information from image/jpeg about when and where the image was taken, etc.

The MEM Servers MAY support extraction of textual information from image/*, for example, Optical Character Recognition (OCR).

10.Bindings for MEM Protocol

According to the [MEM AD], in order to facilitate a mobile email service, the MEM-1 and MEM-2 interfaces MUST be provided for the MEM Protocol. This specification offers service providers two alternatives for the MEM Protocol. Service providers should take into account that neither of these two technologies meets all the requirements defined in the [MEM RD].

10.1 IETF Lemonade

The Lemonade working group in IETF has defined Lemonade Profile bis [Lemonade Profile bis]. [Lemonade Profile bis] describes a profile that uses a large number of SMTP, IMAP, and SIEVE extensions to provide Internet Mail enhancements supporting realizations of the majority of OMA MEM requirements defined in the [MEM RD]. Realization of some OMA MEM requirements will require reference to additional IETF RFCs on Internet Mail, beyond those contained in [Lemonade Profile bis]. [Lemonade Profile bis] also allows a wide range of deployment models – see Appendix C in [MEM AD]. Regardless of the employed deployment model however, The MEM Protocol, the MEM-1 and MEM-2 interfaces MUST be always in place.

10.1.1 MEM Protocol

The MEM Protocol MUST be based on IMAP, SMTP, and SIEVE, as defined in [Lemonade Profile bis]. [Lemonade Profile bis] offers some flexibility for the MEM Client and the MEM Server – meaning that some extensions are optional. The following optional extensions in [Lemonade Profile bis] are REQUIRED by the OMA MEM Enabler:

- [IMAP-FILTERS], which provides a way to persist definition of virtual mailboxes on the MEM Server. This allows:
 - User preferences/filters/settings to follow the user (i.e., to be shared) across devices when used sequentially or simultaneously
 - The user to change filtering rules on his MEM Client (see also [MANAGESIEVE] below)
- [METADATA] and [SIEVE-METADATA], which provide a way of storing user-defined data associated with a user account. This enables the user to:
 - Select the transport method for notifying the MEM Client about new emails based on the capabilities of the MEM Client and network (e.g., SMS)
 - Select if, when and how events are accessed by the MEM Client
- [MANAGESIEVE], a protocol for remotely managing Sieve scripts. This, in conjunction with [IMAP-FILTERS] and IMAP extensions in [Lemonade Profile bis], allow the user to change filtering rules on his/her MEM Client.

In addition to the SMTP, IMAP, and SIEVE extensions, plus other pertinent IETF RFCs defined in [Lemonade Profile bis], the following IETF RFCs are REQUIRED to meet the requirements of the OMA MEM Enabler:

- [RFC3685], supporting spam protection and virus protection for the MEM Enabler.

Furthermore, [RFC2633] and [RFC2015] MAY be needed to support end-to-end confidentiality between two MEM Clients.

In order to declare support for and to use the OMA MEM features, the MEM Client and the MEM Server MUST advertise and agree to use the following capability: X-OMA-MEM-V10 (the capability string is not case sensitive) whenever a new IMAP or SMTP connection is established. This capability agreement differentiates an OMA MEM Server from a pure Lemonade Profile-based server, and it ensures full compliance with the behaviour described in this specification.

The MEM Client MUST establish new IMAP and SMTP connections with the MEM Server as defined in [Lemonade Profile bis] over the MEM-2 interface. It is the MEM Client's responsibility to manage the established IMAP and SMTP connections. The MEM Server MUST communicate with the MEM Client using IMAP and SMTP as defined in [Lemonade Profile bis] over the MEM-1 interface.

10.1.2 MEM-1 Interface

The MEM Client MUST implement the MEM-1 interface. The MEM-1 interface is used by the MEM Server to communicate with the MEM Client while the MEM Client is connected to the MEM Server; therefore the MEM-1 interface is only usable while this connection exists.

While the MEM Client is not connected to the MEM Server, the MEM Server MAY send out-band notifications to the MEM Client provided that the MEM Client set up such preferences – see [MEM AD].

10.1.3 MEM-2 Interface

The MEM Server MUST implement the MEM-2 interface. The MEM-2 interface is used by the MEM Client to communicate with the MEM Server while the MEM Client is connected to the MEM Server; therefore the MEM-2 interface is only usable while this connection exists.

10.1.4 Content Transcoding

Lemonade provides content transcoding using IMAP CONVERT extension [RFC5259]. Header fields conversion to generic Unicode is supported by [RFC5259] using the BODY[...HEADER] data item to the CONVERT/UID CONVERT command, for example:

```
C: D002 CONVERT 2 (NIL ("CHARSET" "utf-8")) BODY[HEADER]
S: * 2 CONVERTED (TAG "d002") (BODY[HEADER] {157})
S: Date: Mon, 20 Apr 2007 20:05:43 +0200
S: From: Peter <peter@siroe.example.com>
S: To: Alexey <alexey@siroe.example.com>
S: Subject: =?UTF-8?Q?why encode this?=?
S:
S: )
S: D002 OK
```

In this example header fields of the message number 2 are converted to UTF-8 character set. The CONVERT/UID CONVERT command leaves the original messages unaffected and accessible by other clients. Furthermore, instead of specifying the exact target MIME media type the client wants to convert to, the client uses a special marker NIL (also known as "default conversion") to request the server to pick a suitable target media type, in accordance with procedures in [RFC 5259].

Other types of conversions described in section 9 can be also performed by the CONVERT/UID CONVERT command by specifying the target MIME type together with conversion parameters as the second parameter to such commands and using the BINARY[...] convert data item. For example:

```
C: b001 CONVERT 2 ("text/plain" ("charset" "utf-8")) BINARY[3]
S: * 2 CONVERTED (TAG "b001") (BINARY[3] {2135})
<the document in text/plain format with utf-8
```

```
charset>
```

```
)
```

```
S: b001 OK CONVERT COMPLETED
```

In this example “charset” is a conversion parameter that controls conversion of the 3rd body part of the 2nd message to text/plain format.

BINARY.SIZE[...] convert data item can be used to determine size of the message/attachment after conversion without the need to download the converted message/attachment first. For example:

```
C: c000 CONVERT 2 ("TEXT/PLAIN" ("CHARSET" "utf-8")) BINARY.SIZE[3]
```

```
S: * 2 CONVERTED (TAG "c000") (BINARY.SIZE[3] 2135)
```

```
S: c000 OK CONVERT COMPLETED
```

A more advanced MEM client MAY take advantage of the AVAILABLECONVERSIONS[...] convert data item to learn about possible target conversions for a given message or attachment before requesting a conversion.

The following table summarizes conversions described in section 9 and lists corresponding CONVERT/UID CONVERT conversion parameters that must be used to perform the necessary conversion:

Description of a transcoding request

Description of a transcoding request	CONVERT/UID CONVERT parameter to be used
Charset conversion to utf-8 for text/* MIME body parts	“CHARSET” “utf-8”
Restricting size of the converted message/attachment (for any MIME type)	“sizeLimit” “<bytes>”
Changing image/* or video/* dimensions	To specify exact dimensions, specify both PIX-X (width in pixels) and PIX-Y (height in pixels) conversion parameters in the CONVERT/UID CONVERT command. To maintain the aspect ratio when resizing, specify one of PIX-X/PIX-Y parameters.
Changing color depth of image/*	Specify “color-levels” and possibly “color” transcoding parameters (see [RFC 2531] for more details)
Changing audio/* or video/* codec	None The target codec is specified using the target MIME type.
Change of bit rate, encoding method (variable bit rate/fixed bit rate), minimum bit rate (for variable bit rate encodings only) for audio/* and video/*	“bitrate” “<value>” (where <value> is a non-negative integer specifying bits per second) “encoding-method” “<value>” (where <value> is one of FBR (fixed bit rate) or VBR (variable bit rate)) “min-bit-rate” “<value>” (where <value> is a non-negative integer specifying the minimum bits per second for a variable bit rate encoding)
Change of frame rate when converting to video/*	“frameRate” “<value>” (The maximum frame rate in frames per second, expressed as a float number)
Extraction of information about image/* in textual format (conversion to text/plain)	“ExifData” “true”

The following table summarizes how CONVERT parameters map to OMA STI parameters:

IMAP CONVERT parameter	OMA STI parameter
CHARSET	“charset” content type parameter ([OMA STI] section 5.2.4.2)
sizeLimit	sizeLimit
PIX-X	width
PIX-Y	height
color (one of “binary”, “grey”, “limited”, “mapped” or “full”)	<colorScheme>scheme (color=binary and color=grey are mapped to scheme=PaletteGrey, color=limited and color=mapped are mapped to scheme=PaletteColor,

	color=full corresponds to scheme=True)
color-levels	<colorScheme>depth
bitrate	bitrate
encoding-method	encoding-method
min-bit-rate	min-bit-rate
frameRate	frameRate

A simple MEM client that wishes to convert a message/attachment to a format it understands MAY use the “default conversion” specified in [RFC5259]. (The “default conversion” is signaled to the MEM server by specifying NIL as the target MIME type). The client wishing to use the default conversion MUST use the “device-id” conversion parameter to identify the type of device whose characteristics are known to the MEM server, which then converts the message/attachment to a format that the device is likely to support and display/play successfully.

10.1.5 Out-band Notifications

For SMS out-band notifications, OMA EMN SHALL be used in its text form. As such, the MEM Server SHALL send the mail information over SMS as a MAILAT URI or an IMAP URI.

OMA MEM SHALL therefore support two payloads:

- a) If the new message is delivered into the INBOX, then the MAILAT URI, as specified in Section 5.3 of OMA-TS-Push_EMN-V1-20071019-A.PDF, SHALL be used. Support for this default notification is mandatory.

An example of the contents of the SMS message is:

```
mailto:user@wapforum.org
```

- b) If the new message is delivered into a folder other than the INBOX then the IMAP URI SHALL be used. Support for this is optional.

An example of the contents of the SMS message to indicate that messages are in folder NEW is:

```
imap://wapforum.org/NEW
```

Upon receipt of either URI, the mobile device SHALL start the email program, including inband notifications.

10.2 OMA-DS

The DS working group in OMA has defined the OMA Data Synchronization Enabler [DS Enabler]. The OMA DS Enabler provides transport and data format agnostic synchronization. Since the OMA DS Enabler provides synchronization for any DS-enabled data store, it is an obvious choice to use it to synchronize a data store containing email objects as well. This section describes how the OMA DS Enabler can be utilized to deploy an email service that cover the OMA MEM requirements defined in the [MEM RD].

10.2.1 Component Mapping

[DS Enabler] requires the following components: OMA DS Server, OMA DS Client, and OMA DS Protocol. These can be mapped almost directly to OMA MEM:

OMA DS Enabler	Maps to	OMA MEM Enabler
OMA DS Server	↔	MEM Server
OMA DS Client	↔	MEM Client
OMA DS Protocol	↔	MEM Protocol
Sync interface for DS Server	↔	MEM-2
Sync interface for DS Client	↔	MEM-1

Table 1 – OMA DS Enabler component mapping

The MEM Protocol, the MEM-1 and MEM-2 interfaces – and the appropriate DS components – MUST be always in place.

10.2.2 MEM Protocol

The OMA Data Synchronization Enabler defines a data synchronization protocol.

The MEM Client MUST establish DS sessions with the MEM Server over the MEM-2 interface using the DS Protocol. The MEM Server MUST communicate with the MEM Client using the DS Protocol.

10.2.3 MEM-1 Interface

The MEM Client MUST implement the MEM-1 interface. The MEM-1 interface is used by the MEM Server to communicate with the MEM Client while the MEM Client is connected to the MEM Server; therefore the MEM-1 interface is only usable while this connection exists.

While the MEM Client is not connected to the MEM Server, the MEM Server MAY send Out-band Notifications to the MEM Client provided the MEM Client set up such preferences – see [6].

10.2.4 MEM-2 Interface

The MEM Server MUST implement the MEM-2 interface. The MEM-2 interface is used by the MEM Client to communicate with the MEM Server while the MEM Client is connected to the MEM Server; therefore the MEM-2 interface is only usable while this connection exists.

10.2.5 Out-band Notifications

For SMS outband-notification, the MEM Server can alert the MEM client to perform synchronization by using the Server Alerted Sync (SAS) mechanism that is defined in [OMA-DSPROTO].

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-Mobile_Email-V1_0-20110802-A	02 Aug 2011	Status changed to Approved by TP: OMA-TP-2011-0273-INP_Mobile_Email_V1_0_ERP_for_final_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for MEM Client

Item	Function	Reference	Status	Requirement
MEM-C-001	Support of MEM Protocol	Section 5 Section 10	M	MEM-C-008 OR MEM-C-009
MEM-C-002	Support of Out-band notifications mechanism	Section 6.3 Section 6.4.3	O	
MEM-C-003	Support of notification preferences	Section 6.3 Section 6.4	M	
MEM-C-004	Support Client Identification (Client-ID)	Section 6.4.1	M	
MEM-C-005	Support of publishing capability for both device and MEM client information	Section 7	O	
MEM-C-006	Support provisioning profile using OMA Device Management	Section 8.1 (MEM MO)	M	
MEM-C-007	Support of requesting conversion of the email	Section 9	O	MEM-C-008 OR MEM-C-009
MEM-C-008	Support of IETF Lemonade Profile in data channel	Section 10.1	O	
MEM-C-009	Support for OMA DS Profile binding in data channel	Section 10.2	O	

B.2 SCR for MEM Server

Item	Function	Reference	Status	Requirement
MEM-S-001	Support of MEM Protocol	Section 5 Section 10	M	MEM-S-010 OR MEM-S-011
MEM-S-002	Support of storing and filtering notification preferences of MEM client	Section 6	M	
MEM-S-003	Support of sending Out-band notification to offline MEM Client	Section 6	O	MEM-S-010 OR MEM-S-011
MEM-S-004	Support of SMS notification	Section 6.3.1	M	MEM-S-003 MEM-S-010 OR MEM-S-011
MEM-S-005	Support of storing device and MEM client capability information	Section 7	O	
MEM-S-006	Support for MEM Client provisioning profile using OMA Device Management	Section 8.1 (MEM MO)	M	

Item	Function	Reference	Status	Requirement
MEM-S-007	Support of content transcoding	Section 9.1	M	MEM-S-010
MEM-S-008	Support for character set and image conversions	Section 9.2	M	
MEM-S-009	Support for audio, video, and text conversions	Section 9.3	O	
MEM-S-010	Support of IETF Lemonade Profile in data channel	Section 10.1	O	
MEM-S-011	Support for OMA DS Profile binding in data channel	Section 10.2	O	

Appendix C. ABNF definitions

The specification contains ABNF notations to describe the syntax of information elements. This appendix collects all ABNF definitions into two sections – one for rules and another one for terminal values -, sorted alphabetically.

C.1 ABNF rules

```
Client-ID = 1*(ALPHA / DIGIT)
EventList = EventType 1*("+ " EventType)
EventType = 1*ALPHA
           ; Defined in [Store-Events]
PriorityIndicator = 1*DIGIT
SubscribedEventTypeList = nil / EventType / EventList
```

C.2 ABNF terminal values

This section contains terminal values defined in this document only. Please refer to [RFC5234] for the rest of the terminal values.

```
Boolean = "true" / "false"
nil = "nil"
```