



vObject OMA Minimum Interoperability Profile Requirements

Candidate Version 1.0 – 18 Jan 2005

Open Mobile Alliance
OMA-RD-vObjectOMAPProfile-V1_0-20050118-C

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

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

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

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

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

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

© 2005 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 (INFORMATIVE)	4
2. REFERENCES	5
2.1 NORMATIVE REFERENCES	5
2.2 INFORMATIVE REFERENCES	5
3. TERMINOLOGY AND CONVENTIONS	6
3.1 CONVENTIONS	6
3.2 DEFINITIONS	6
3.3 ABBREVIATIONS	6
4. INTRODUCTION (INFORMATIVE)	7
5. USE CASES (INFORMATIVE)	8
5.1 SENDING vOBJECT VIA LOCAL INTERFACE	8
5.1.1 Short Description	8
5.1.2 Actors	8
5.1.3 Pre-conditions	8
5.1.4 Post-conditions	8
5.1.5 Normal Flow	8
5.1.6 Operational and Quality of Experience Requirements	9
5.2 vOBJECT AS AN EMAIL ATTACHMENT	9
5.2.1 Short Description	9
5.2.2 Actors	9
5.2.3 Pre-conditions	9
5.2.4 Post-conditions	9
5.2.5 Normal Flow	9
5.3 DOWNLOADING vOBJECT VIA BROWSER	10
5.3.1 Short Description	10
5.3.2 Actors	10
5.3.3 Pre-conditions	10
5.3.4 Post-conditions	10
5.3.5 Normal Flow	10
5.3.6 Alternative Flow	11
5.4 MMS POSTCARD	11
5.5 INTEROPERABILITY CONSIDERATIONS	11
6. REQUIREMENTS (NORMATIVE)	12
6.1 GENERAL	12
6.2 HIGH-LEVEL FUNCTIONAL REQUIREMENTS	12
6.2.1 Features	12
6.2.2 Security	12
6.2.3 Charging	12
6.2.4 Administration and Configuration	12
6.2.5 Usability	13
6.2.6 Interoperability	13
6.2.7 Privacy	13
APPENDIX A. CHANGE HISTORY (INFORMATIVE)	14
A.1 APPROVED VERSION HISTORY	14
A.2 DRAFT/CANDIDATE VERSION <CURRENT VERSION> HISTORY	14

1. Scope

(Informative)

This document presents the use cases and requirements for developing a vObject minimum interoperability profile based on the vCard 2.1, vCalendar 1.0 and vBookmark1.0 specifications along with guidelines to ensure a consistent, non-ambiguous interpretation of the relevant base specifications.

It is not the purpose of this document to address any OMA enabler requirements which might result in proposing extensions or modifications to any of the specifications mentioned above. It is recognized that existing and new applications will introduce such vObject requirements. It is expected that these requirements will be addressed by a complementary OMA activity focusing on a longer term interoperability solution.

Interoperability of vObjects covers a range of issues. It is the intention of the vObject minimum interoperability profile to address the issues of physical data formatting for these objects as well as specifying a standard presentation for commonly used data properties. These issues have been identified as a subset of the larger set of interoperability; it does not attempt to address all the interoperability issues known to exist with the specifications.

2. References

2.1 Normative References

- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997,
URL:<http://www.ietf.org/rfc/rfc2119.txt>

2.2 Informative References

- [vCard2.1] “vCard The Electronic Business Card Version 2.1”, A versit Consortium Specification, September 18, 1996
URL: <http://www.imc.org/pdi/vcard-21.doc>
- [vCal] “vCalendar The Electronic Calendaring and Scheduling Exchange Format Version 1.0”, A versit Consortium Specification, September 18, 1996
URL: <http://www.imc.org/pdi/vcal-10.doc>
- [RFC2425] “A MIME Content-Type for Directory Information”, RFC2425.
URL: <http://www.faqs.org/rfcs/rfc2425.html>
- [RFC2426] “vCard MIME Directory Profile”, RFC2426.
URL: <http://www.faqs.org/rfcs/rfc2426.html>
- [RFC2445] “Internet Calendaring and Scheduling Core Object Specification (iCalendar)”, RFC2445
URL: <http://www.faqs.org/rfcs/rfc2445.html>
- [RFC2446] “iCalendar Transport-Independent Interoperability Protocol (iTIP) Scheduling Events, BusyTime, To-dos and Journal Entries”, RFC2446
URL: <http://www.faqs.org/rfcs/rfc2446.html>
- [RFC2447] “iCalendar Message-Based Interoperability Protocol (iMIP)”, RFC2447
URL: <http://www.faqs.org/rfcs/rfc2447.html>
- [vBookmark] “IrDA Infrared Mobile Communications v1.1”
URL: <http://www.irda.org/>
- [SyncML-IOP-Whitepaper] “SyncML Common Object Interoperability White Paper”, Open Mobile Alliance™
OMA-DS-2004-0052-Interoperability-White-Paper on OMA DS web portal,
URL:<http://www.openmobilealliance.org/>
- [WAE20] “Wireless Application Environment Specification – version 2.0”, WAP Forum™. WAP-236-WAESpec. URL:<http://www.openmobilealliance.org/>
- [MMS_V12] “OMA Conformance Document Version 1.2”, Open Mobile Alliance™, OMA-MMS-CONF-v1_2, URL:<http://www.openmobilealliance.org/>

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

This is an informative document, which is not intended to provide testable requirements to implementations.

3.2 Definitions

3.3 Abbreviations

MIME	Multipurpose Internet Mail Extensions
IMC	Internet Mail Consortium
IrDA	Infrared Data Association

4. Introduction

(Informative)

This document presents the use cases and requirements for a vObject minimum interoperability profile based on the vCard 2.1[vCard2.1], vCalendar 1.0[vCal] and vBookmark1.0[vBookmark] specifications along with guidelines to ensure a consistent, non-ambiguous interpretation of the relevant base specifications. It is recognized that existing and new applications will introduce additional vObject requirements, which will be addressed separately by a complementary OMA activity focusing on a longer term interoperability solution.

There are a number of activities regarding vObject that have been carried-out in different organizations. The Versit consortium developed vCard and vCalendar (the most recent versions of the specification are vCard2.1 [vCard2.1], vCalendar1.0 [vCal]). In 1996, the IMC assumed the work from Versit to further develop and promote these two important technologies.

Other open standards include vCard3.0 and iCalendar. The vCard3.0 specification is consists of two parts:

- MIME Content-Type for directory information [RFC2425]
- vCard MIME Directory profile [RFC2426].

The iCalendar specification consists of three RFCs:

- Internet Calendaring and Scheduling Core Object Specification [RFC2445]
- iCalendar Transport-Independent Interoperability Protocol (iTIP) [RFC2446]
- iCalendar Message-based Interoperability Protocol (iMIP) [RFC2447].

vBookmark1.0 [vBookmark] was developed by the IrDA which was established to promote infrared standards to allow convenient cordless connectivity and foster application interoperability over a broad range of platforms and devices.

OMA activities related to vObject include (1) the development of SyncML common object interoperability white paper[SyncML-IOP-Whitepaper] (2) the development of MMS Conformance Document[MMS_V12], and (3) MAE specification [WAE20]. The latter provides implementation guidelines for vCard and vCalendar, however, a vObject minimum interoperability profile was not developed within those activities.

5. Use Cases (Informative)

The following are a collection of vObject use cases that are considered to represent a good basis for the derivation of requirements

5.1 Sending vObject via local interface

5.1.1 Short Description

This use case describes a situation where Alice sends a vCard (the same use case as sending vCalendar or vBookmark) to Ken via an infrared interface. Ken is able to receive the vCard from Alice and save it to his addressbook on his terminal. While this use case illustrates the exchange of vCard via infrared interface, this use case is also intended to illustrate the exchange of vCalendar or vBookmark data, as well as, illustrate exchanges via other local interfaces as well (e.g. Bluetooth, etc).

5.1.2 Actors

- Alice– the sender of the vCard
- Ken – the recipient of the vCard for saving to his addressbook

5.1.2.1 Actor Specific Issues

- Alice would like to send her contact information stored on her terminal to Ken in a convenient manner.
- Ken would like to receive and store Alice’s contact information in the addressbook of his terminal.

5.1.2.2 Actor Specific Benefits

- Alice is able to conveniently send her contact information to Ken over a local interface.
- Ken is able to conveniently receive Alice’s contact information and store it in his addressbook without having to manually input her contact information.

5.1.3 Pre-conditions

- Alice has a terminal capable of sending and receiving vObject data via infrared interface.
- Ken has a terminal capable of sending and receiving vObject data via infrared interface.
- Alice is able to browse her addressbook and choose a vCard to send via local interface (e.g. IrDA, Bluetooth).
- Ken is able to receive the vCard data via local interface and save it to his addressbook.

5.1.4 Post-conditions

Ken receives vCard data and stores it to his addressbook.

5.1.5 Normal Flow

- ① Alice browses her addressbook and selects a vCard entry
- ② From her addressbook application, Alice chooses to send the selected vCard to Ken via infrared interface
- ③ Alice sends the vCard to Ken
- ④ After Ken’s handset receives the data via infrared interface, the handset determines that the data is a vCard

- ⑤ A confirmation message is shown to Ken asking if he would like to save the vCard to the addressbook
- ⑥ Ken chooses to save the vCard to his addressbook

5.1.6 Operational and Quality of Experience Requirements

None.

5.2 vObject as an Email Attachment

5.2.1 Short Description

This use case describes the situation where Alice is composing an email, Alice chooses to attach a vCard from her addressbook and then sends it to Ken. While this use case illustrates the exchange of vCard via email attachment, this use case is also intended to illustrate the exchange of vCalendar or vBookmark data, as well as, illustrate exchanges via other messaging technologies (e.g. MMS)

5.2.2 Actors

- Alice—the composer of an email with a vCard attachment
- Ken—the recipient of the email with a vCard attachment

5.2.2.1 Actor Specific Issues

- Alice would like to send her contact information stored on her terminal to Ken in a convenient manner.
- Ken would like to receive and store Alice's contact information in the addressbook of his terminal.

5.2.2.2 Actor Specific Benefits

- Alice is able to conveniently send her contact information to Ken over
- Ken is able to conveniently receive Alice's contact information and store to his addressbook without having to manually input her contact information.

5.2.3 Pre-conditions

- Alice has a terminal with an email client capable of sending and receiving vObject data as an email attachment.
- Ken has a terminal with an email client capable of sending and receiving vObject data as an email attachment
- Alice is able to browse her addressbook and choose a vCard to attach to an email created on her terminal
- Ken's terminal has an email client capable of saving vCard email attachments to his addressbook.

5.2.4 Post-conditions

- Ken receives vCard data and stores it to his addressbook

5.2.5 Normal Flow

- ① After Alice finishes composing an email on her terminal, she chooses to add an attachment to the email. From email composer interface, Alice selects to add attachment from here addressbook
- ② Alice opens her addressbook, chooses a specific vCard, and attaches it to the email
- ③ Alice sends the email to Ken

- ④ Ken receives the email and chooses to save the received vCard attachment to his addressbook
- ⑤ A confirmation message is shown to Ken asking if he would like to save the vCard to the addressbook
- ⑥ Ken selects [Yes] and the vCard is saved in his addressbook.

5.3 Downloading vObject via browser

5.3.1 Short Description

This use case describes a situation where Alice is browsing the mobile content provider ABC's web site and wishes to download an event schedule to her own scheduler. ABC has provided the necessary mechanism on the "download event schedule" page on their web site. While this use case illustrates the download of vCalendar via browser, this use case also intended to illustrate the download of vCard or vBookmark data.

5.3.2 Actors

- Alice – user who browses ABC's web page in order to download and ABC's event schedule information to her terminal
- ABC – a content provider that provides event schedule information on its homepage

5.3.2.1 Actor Specific Issues

- Alice would like to acquire ABC's event schedule information and save it to her terminal for later reference.
- ABC would like to allow users to its web site to obtain ABC event schedule information

5.3.2.2 Actor Specific Benefits

- Alice is able to download and add event schedule information to the scheduler application on her terminal automatically without having to manually input the information
- ABC is able to offer its users a convenient way to access and store ABC's event schedule information in the scheduling application located on the handset.

5.3.3 Pre-conditions

- ABC has created a 'download event schedule' page available for its users. Alice is browsing ABC's web page and wishes to down their event schedule to her scheduler application.

5.3.4 Post-conditions

- Alice successfully downloads the desired event schedule from ABC's web page and saves the information to the schedule application on her handset.

5.3.5 Normal Flow

- ① ABC's web site contains a "download our event schedule" hyperlink which links to a vCalendar object.
- ② Alice browses ABC's web site and clicks on the 'download our event schedule' hyperlink.
- ③ ABC's event schedule information is downloaded to Alice's handset. A confirmation message is shown to Alice to confirm if she would like to add the downloaded schedule to her scheduler application.
- ④ Alice confirms to add the event schedule to her scheduler application
- ⑤ Alice opens her scheduler application and can view the event schedule downloaded from ABC's web site.

5.3.6 Alternative Flow

None

5.4 MMS PostCard

This use case refers to Use Case TPSP, MMS Postcard in MMS requirement document.

5.5 Interoperability considerations

Ambiguities in the vCard2.1[vCard2.1], vCalendar1.0[vCal] and vBookmark1.0[vBookmark] specifications coupled with the lack of well-known implementation guidelines has led to various commercial implementations that may result in interoperability problems for the uses cases described in this specification. The following list presents possible examples of interoperability problems that may occur. This list provides examples and is not intended to present an exhaustive list of issues.

- Ken receives a vCard sent from Alice and finds that the last name information field has been lost. This situation could arise if the implementation on Alice's terminal uses semi-column to separate the first name and last name fields of the N property and the implementation on Ken's terminal is unable to process it and treats the semi-column as an end tag for the vCard information.
- Ken receives a vCard sent from Alice and finds that he lost two telephone entries. This situation could arise if the implementation of Alice's terminal supports up to five telephone entries and the implementation on Ken's terminal only supports up to three entries for the same telephone type.
- Ken's terminal is unable to process the vCard which attached to the email from Alice and returns an error. This situation could arise if the implementation on Ken's terminal can only identify text/x-vCard as the content-type for a vCard object and the implementation on Alice's terminal supports others content-types defined for the vCard object.
- Ken loses the time information of the meeting sent from Alice. This situation could arise if Ken's terminal supports only the UTC time format and Alice's terminal supports ISO 8601 standard for the representation of dates and times.
- Ken downloads a task from the web and within his terminals task application the task is shown as medium priority even though it was created with a low priority. VTOD support priority levels of 1 through9 but how these map to low, medium, and high differs from implementation to implementation.
- Alice creates a recurring meeting that is every 2nd Tuesday of the month for 6 months and she emails it to Ken. Ken only sees the first instance in his terminal's calendar. Implementations have not properly implemented the full set of RRULES.
- Ken beams Alice an day event. On Alice's terminal it shows up as a 24hour meeting starting at midnight and not as one of the terminals supported all day events. There is no proper way of specifying an all day event.
- Alice beams Ken an untimed event. Ken's terminal shows nothing. Similar to all day events there is no proper way of specifying an untimed event.
- Alice downloads the schedule for June for the local sports team. The titles of all the events show up in her terminal with garbage characters. A different default char set has been assumed.
- Ken receives an assigned task through email. He can't figure out what he is supposed to do however as the description has weird entries at the end. The multi-line task description has been improperly folded.
- Ken receives a vCard sent from Alice containing a big image data, and finds his terminal is not able to process it.

6. Requirements (Normative)

6.1 General

1. The definition of vObject OMA minimum interoperability profile MUST leverage existing work from other relevant standards bodies (e.g., IMC, IrDA).
2. The vObject OMA minimum interoperability profile datasets defined in OMA MUST be based on existing standards vCard2.1[vCard2.1], vCalendar1.0[vCal] and vBookmark1.0[vBookmark] without extensions or modifications to any of the standards, but syntactic and semantic ambiguities are to be addressed using vCard 3.0 and/or other sources for possible fixes.
3. The vObject implementation guideline defined in OMA MUST aligned with OMA MMS Conformance Document [MMS_V12], MAE Specifications[WAE20] and SyncML common object interoperability whitepaper [SyncML-IOP-Whitepaper].

6.2 High-Level Functional Requirements

This section presents the high-level functional requirements that are derived from the use cases.

6.2.1 Features

1. The vObject profile MUST support the following content-types
 - a. vCard2.1
 - b. vCalendar1.0
 - c. vBookmark1.0
2. The user SHALL be able to send or receive vObject data over local transmission interfaces (e.g., IrDA, Bluetooth)
3. The recipient of vObject data SHALL be able to store such data to a local application residing on the terminal.
4. The user SHALL be able to download vObject data via the browser interface. The user MAY store the downloaded data to a local application residing on the terminal.
5. The user SHALL be able to send or receive vObject data as an attachment to email for those terminals that have email clients capable of attachments.

6.2.2 Security

This work is restricted to define vObject data formatting and security is out of scope of this work.

6.2.3 Charging

No requirements have been identified.

6.2.4 Administration and Configuration

No requirements have been identified.

6.2.5 Usability

No requirements have been identified.

6.2.6 Interoperability

1. Content type, file extension **MUST** be standardized to ensure interoperability.
2. Encoding and usage of character set **MUST** be standardized to ensure interoperability.
3. Folding method **MUST** be specified.
4. Minimum field sizes for all supported fields **MUST** be specified
5. Each of the vCard, vCalendar and vBookmark objects **MUST** support a minimum size for the object types, and the terminal **MUST** be able to process any one of these objects
6. Minimum set of attributes and supported orders for data type (e.g. TEL, EMAIL, ADR, etc) **MUST** be specified to ensure interoperability.
7. Format of data values (e.g. time) **MUST** be specified to ensure interoperability.

6.2.7 Privacy

This work is restricted to define vObject data formatting and privacy is out of scope of this work.

Appendix A. Change History

(Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-RD_vObjectOMAPProfile-V1_0	18 May 2004	whole document	Initial draft created
	20 June 2004	3.2, 3.3, 6.2.2, 6.2.7 1,4	According to comments received from BAC-MAE and in interim review with REQ
	23 June 2004	1, 5.1.2, 5.1.2.2, 5.2.1, 5.2.5, 5.4, 5.5, 6.1, 6.2.6	According to comments received in formal review with REQ
Candidate Versions OMA-RD_vObjectOMAPProfile-V1_0	13 July 2004		Candidate version approved by OMA TP
	15 Dec 2004	2.2, 6.1	According to OMA-MAE-2004-0183-OMA-RD-vObject-ChangeRequest-20041216-1
OMA-RD-vObjectOMAPProfile-V1_0	18 Jan 2005	n/a	OMA TP ref# OMA-TP-2005-0014-vObject-V1_0-for-Candidate-approval.