OMA DS Standards Change History

Candidate – 02 Oct 2007

Open Mobile Alliance

OMA-WP-SyncML_ChangeHistory-20071002-C

Contents

1. SCOPE 4

2. REFERENCES 5

3. TERMINOLOGY AND CONVENTIONS 6

3.1 CONVENTIONS 6

3.2 DEFINITIONS 6

3.3 ABBREVIATIONS 6

4. INTRODUCTION 7

5. A BRIEF HISTORY OF SYNCML, SYNCML INITIATIVE AND OMA 8

5.1 SYNCML INITIATIVE 8

5.2 OMA AND SYNCML INITIATIVE 8

6. SYNCML/OMA DS SPECIFICATIONS EVOLUTION 9

6.1 CHANGES OVERVIEW 9

6.1.1 SyncML 1.0.1 9

6.1.2 OMA DS 1.1.2 10

6.1.3 OMA DS 1.2 11

6.2 OMA DS FEATURE CHANGES FROM 1.1.2 TO 1.2 13

6.3 OMA DS TAG CHANGES FROM 1.1.2 TO 1.2 14

6.4 OMA DS CONFORMANCE CHANGES FROM 1.1.2 TO 1.2 15

6.4.1 Client Conformance Changes 15

6.4.2 Server Conformance Changes 15

6.5 OMA DS NEW CODES IN 1.2 16

6.5.1 New Alert Codes 16

6.5.2 New Status Codes 16

APPENDIX A. CHANGE HISTORY (INFORMATIVE) 17

1. Scope

This document is intended to help newcomers to understand OMA Data Sync (also known as SyncML DS) protocol and to select the appropriate release enabler they would like to implement. This document might also help developers to migrate from release 1.1.2 to 1.2, in that it will outline the feature differences between various releases.

This document is informative, and is not intended to add requirements to current OMA DS and OMA SyncML Common specifications.

2. References

[DSPRO_V1_1_2] “SyncML Synchronization Protocol”, Open Mobile Alliance™,
OMA-SyncML-DataSyncProtocol-V1_1_2-20030612-A,
http://www.openmobilealliance.org/release_program/ds_v112.html

[DSPRO_V1_2] “SyncML Synchronization Protocol”, Open Mobile Alliance™,
OMA-SyncML-DataSyncProtocol-V1_2-20040601-C,
http://www.openmobilealliance.org/release_program/ds_v12.html

[DSREPU_V1_1_2] “SyncML Representation Protocol, Data Synchronization Usage”, Open Mobile Alliance™,
OMA-SyncML-DataSyncRep-V1_1_2-20030612-A,
http://www.openmobilealliance.org/release_program/ds_v112.html

[DSREPU_V1_2] “SyncML Representation Protocol, Data Synchronization Usage”, Open Mobile Alliance™,
OMA-SyncML-DataSyncRep-V1_2-20040601-C,
http://www.openmobilealliance.org/release_program/ds_v12.html

[REPPRO_V1_1_2] “SyncML Representation Protocol”, Open Mobile Alliance™,
OMA-SyncML-RepPro-V1_1_2-20040711-A,
http://www.openmobilealliance.org/release_program/SyncML_v112.html

[REPPRO_V1_2] “SyncML Representation Protocol”, Open Mobile Alliance™,
OMA-TS-SyncML_RepPro-V1_2-20050509-C,
http://www.openmobilealliance.org/release_program/SyncML_v12.html

[METAINF_V1_1_2] “SyncML Meta Information”, Open Mobile Alliance™,
OMA-SyncML-MetaInfo-V1_1_2-20030612-A,
http://www.openmobilealliance.org/release_program/SyncML_v112.html

[METAINF_V1_2] “SyncML Meta Information”, Open Mobile Alliance™,
OMA-TS-SyncML_MetaInfo-V1_2-20050509-C,
http://www.openmobilealliance.org/release_program/SyncML_v12.html

[DEVINFO_V1_1_2] “SyncML Device Information”, Open Mobile Alliance™,
OMA-SyncML-DevInfo-V1_1_2-20040721-A,
http://www.openmobilealliance.org/release_program/ds_v112.html

[DEVINFO_V1_2] “SyncML Device Information”, Open Mobile Alliance™,
OMA-SyncML-DevInfo-V1_2-20040601-C,
http://www.openmobilealliance.org/release_program/ds_v12.html

[SAN] “SyncML Server Alerted Notification”, Open Mobile Alliance™,
OMA-TS-SyncML_SAN-V1_2-20050509-C,
http://www.openmobilealliance.org/release_program/SyncML_v12.html

[EMAILOBJ] “Email Data Object Specification”, Open Mobile Alliance™,
OMA-SyncML-DataObjEmail-V1_2-20040601-C,
http://www.openmobilealliance.org/release_program/ds_v12.html

[FILEOBJ] “File Data Object Specification”, Open Mobile Alliance™,
OMA-SyncML-DataObjFile-V1_2-20040601-C,
http://www.openmobilealliance.org/release_program/ds_v12.html

[FOLDEROBJ] “Folder Data Object Specification”, Open Mobile Alliance™,
OMA-SyncML-DataObjFolder-V1_2-20040601-C,
http://www.openmobilealliance.org/release_program/ds_v12.html

[PRIMER] "A Primer to SyncML/OMA DS", Open Mobile Alliance™,
OMA-WP-SyncML_Primer-20071002-C,
http://www.openmobilealliance.com/tech/publicmaterial.html

3. Terminology and Conventions

3.1 Conventions

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

3.2 Definitions

Client A SyncML Client refers to the protocol role when the application issues SyncML "request" messages. For example in data synchronization, the Sync SyncML Command in a SyncML Message.

Data A unit of information exchange, encoded for transmission over a network.

Data collection A data element which acts as a container of other data elements, (e.g., {c {{i1, data1}, ... {in, datan}}}). In SyncML, data collections are synchronized with each other. See data element.

Data element A piece of data and an associated identifier for the data, (e.g., {i, data}).

Data synchronization The act of establishing an equivalence between two data collections, where each data element in one item maps to a data item in the other, and their data is equivalent (but not necessarily equal).

Data synchronization protocol The well-defined specification of the "handshaking" or workflow REQUIRED to accomplish synchronization of data elements on an originator and recipient data collection. The SyncML specification forms the basis for specifying an open data synchronization protocol.

Data type The schema used to represent a data object (e.g., text/calendar MIME content type for an iCalendar representation of calendar information or text/directory MIME content type for a vCard representation of contact information).

Enabler Release A collection of specifications that combined together form an enabler for a service area, e.g. a download enabler, a browsing enabler, a messaging enabler, a location enabler, etc. The specifications that are forming an enabler should combined fulfill a number of related market requirements.

Message A SyncML Message is the primary contents of a SyncML Package. It contains the SyncML Commands, as well as the related data and meta-information. The SyncML Message is an XML document.

Representation protocol A well-defined format for exchanging a particular form of information. SyncML is a representation protocol for conveying data synchronization and device management operations.

Server A SyncML Server refers to the protocol role when an application issues SyncML "response" messages. For example in the case of data synchronization, a Results Command in a SyncML Message.

Synchronization Anchor A string representing a synchronization event. The format of the string will typically be either a sequence number or an ISO 8601-formatted extended representation, basic format date/time stamp.

3.3 Abbreviations

DM	Device Management
DS	Data Synchronization
HTTP	HyperText Transfer Protocol
OBEX	OBject EXchange protocol
OMA	Open Mobile Alliance
WSP	Wireless Session Protocol

4. Introduction

OMA Data Sync protocol (also known as SyncML DS protocol) is a fairly powerful and flexible synchronization protocol that offers the following characteristics:

- Operates effectively over wireless and wireline networks
- Supports a variety of transport protocols, including HTTP, WSP (Wireless Session Protocol), OBEX (Bluetooth, IrDA), SMTP, pure TCP/IP networks, and proprietary communication protocols
- Supports arbitrary networked data and common personal data formats such as vCard, vCalendar, iCalendar, and e-mail, among others
- Addresses the resource limitations of the device
- Is built upon existing Internet and Web technologies

The protocol's minimal function delivers the most commonly required synchronization capability across the entire range of devices (such as small-memory-footprint mobile devices).

In this document, we provide some guidance to SyncML developers, who would like to implement this protocol on a client or a server platform. In addition with [PRIMER], this document can be viewed as guiding map, in which the reader will find brief description of specific part or releases of the protocol, with links to additional information in the specification documents.

[PRIMER] proposes a short introduction to data synchronization challenges, as well as to SyncML protocol. Furthermore, the reader will find in that paper a description of the different specification documents provided in the Enabler Release Package, and some information about the OMA DS test process.

On its side, the current document is a snapshot of the evolution of SyncML through the different releases. Changes between 1.1.2 and 1.2 releases are more detailed, as 1.2 release introduces interesting new features or powerful mechanisms, like folders/emails synchronization or search/filtering capabilities.

5. A brief history of SyncML, SyncML Initiative and OMA

5.1 SyncML Initiative

The **SyncML Initiative, Ltd.** was a not-for-profit corporation formed in February 2000 by a group of companies who co-operated to produce an open specification for data synchronization and device management that can be used industry-wide.

Prior to SyncML, data synchronization and device management had been based on a set of different, proprietary protocols, each functioning only with a very limited number of devices, systems and data types. These non-interoperable technologies have complicated the tasks of users, manufacturers, service providers, and developers. Further, a proliferation of different, proprietary data synchronization and device management protocols has placed barriers to the extended use of mobile devices, has restricted data access and delivery and limited the mobility of the users.

At the beginning, The SyncML Initiative was sponsored by mobile technology industry leaders, such as Ericsson, IBM, Lotus, Matsushita, Motorola, Nokia, Palm Inc., Psion, Starfish Software. In few months, a record number of technology companies joined SyncML as industry supporters, to pioneer the development, introduction and adoption of universal data synchronization. New members that had joined the initiative included industry leaders such as Cisco Systems, Samsung, Bell Labs Lucent Technologies, America Online, Intel, Siemens, Sun Microsystems, Yahoo! and other companies all around the world. The rapid growth of members was evidence to the significance of the future of data synchronization.

The first SyncML 1.0 specification was published in December 2000 and the first products passed the SyncML conformance and interoperability testing processes in April 2001.

5.2 OMA and SyncML Initiative

The **Open Mobile alliance (OMA)** was formed in June 2002 by nearly 200 companies including the world's leading mobile operators, device and network suppliers, information technology companies and content and service providers. The fact that the whole value chain is represented in OMA marks a change in the way specifications for mobile services are done. Rather than keeping the traditional approach of organizing activities around "technology silos", with different standards and specifications bodies representing different mobile technologies, working independently, OMA is aiming to consolidate into one organization all specification activities in the service enabler space.

The **SyncML Initiative** merged with the **Open Mobile Alliance** in November 2002. The SyncML legacy specifications were converted to the OMA format with the 1.1.2 versions of OMA SyncML Common, OMA Data Synchronization and OMA Device Management in May 2002. The relationship between these documents which had been created during the SyncML Initiative has been preserved and is detailed in [PRIMER].

6. SyncML/OMA DS Specifications Evolution

6.1 Changes Overview

As SyncML Initiative joined OMA in 2002, post-2002 SyncML releases has been renamed OMA Data Sync. Therefore you will find two sets of versions: SyncML 1.0.0, 1.0.1, 1.1.0, 1.1.1, prior to OMA integration, and OMA DS 1.1.2 and 1.2, since OMA integration. SyncML 1.1.1 and OMA DS 1.1.2 versions are nearly the same. The major changes relate to evolutions between SyncML 1.0.1 and SyncML 1.1.1, and between OMA DS 1.1.2 and OMA DS 1.2. The next sections outline the main changes made to the protocol since 1.0.1 release.

6.1.1 SyncML 1.0.1

The SyncML protocol v1.0.1 main features are:

- **Different Sync Types**

Two-way Sync	A normal sync type in which the client and the server exchange information about modified data in their devices. The client sends the modification first.
Slow sync	A form of two-way sync in which the client and the server exchange all the data from their database. This type of sync is used for the first sync, or after a synchronization failure.
One way sync from client only	A sync type in which the client sends its modifications to the server, but the server does not send its modifications back to the client.
Refresh sync from client only	A sync type in which the client sends all its data from a database to the server (i.e. backup). The server is expected to replace all data in the target database with the data sent by the client.
One way sync from server only	A sync type in which the client gets all modifications from the server but the client does not send its modification to the server.
Refresh sync from server only	A sync type in which the server sends all its data from a database to the client (i.e. restore). The client is expected to replace all data in the target database with the data sent by the server.
Server Alerted Sync	A sync type in which the server alerts the client to perform sync. That is, the server informs the client to start a specific type of sync with the server.

- **Usage of anchors**

The protocol uses sync anchors (see Definitions) to enable sanity checks of synchronization. Each device manages two anchors: the Last anchor, which describes the last synchronization event (e.g. time), and the Next anchor which describe the current event of sync. The client and server must send their own anchors to each other. Each device must also store the Next anchor of the other until the next sync event. By comparing the matching values of Last anchors sent and Next anchors stored, one device is able to conclude that no failures have happened to the other since last sync.

- **Mapping of ids**

The protocol is based on the principle that the client and the server can have their own ID's for data items in their database. Because the ID's can be different, the server must maintain a ID mapping table for items, in order to know which client ID (LUID) and which server ID (GUID) corresponds to the same data item. Then client is requested to send mapping information when new items, sent by the server, are added to its database.

- **Conflict resolution**

Conflicts happen when modifications have been made on the same items on the server and the client databases. The server is in general assumed to include the sync engine functionality that resolves the conflicts. But it can also return back to the client a notification to let the client resolve itself the conflicts. The protocol provides status codes for some common policies to resolve a conflict. These status codes are used to inform the other device of the conflict and how it has been resolved. The administration and configuration of the conflict resolution policy is beyond the scope of the SyncML protocol.

- **Authentication**

The protocol requires the support for the basic and the MD5 digest access authentication on the server layer. The authentication on the database layer is also enabled, if the device needs it. Both the sync client and the server can challenge for the authentication and the device receiving the authentication challenge must be able to send the authorization credentials back.

- **Exchange of device capabilities**

The protocol provides the functionality to exchange the device capabilities (identifier, datastore names, data format supported for each datastore, etc.) during the initialization. The client must send its device information to the server when the first synchronization is done, or when the static device information has been updated. It should also be able to receive the server device information.

- **Device memory management**

The protocol provides also possibility to specify the dynamic memory capabilities for databases of a device or for persistent storage on a device. It specifies how much memory is left for usage.

- **Multiple messages in packages**

As depicted in the [PRIMER] document, a sync session is composed with the exchange of different packages of information between the client and the server. The protocol provides the functionality to transfer one package in multiple SyncML messages, if the package is considered too large because of transport or device limitations.

For example, a client can send its modifications using three successive messages. For each message sent, the server responds by a message including status information for these modifications. In practice, client and server packages might be interlaced, because the receiver of a package may start to send its next package at the same time if it makes sense.

- **Sync without separate initialization**

The synchronization can be started without a separate initialization. This means that the initialization and data changes can be sent within the same message, in order to decrease the number of messages to be exchanged. The server must be able to handle that case.

- **Busy signaling**

If the server is able to receive the data from the client, but is not able to process the request(s) at a reasonable time after receiving the modifications from the client, the server must send information about that to the client. This happens by sending a busy status package back to the client. After that, the client may ask the sync results later or start the synchronization from the beginning.

6.1.2 OMA DS 1.1.2

The OMA Data Synchronization 1.1.2 (or SyncML 1.1.1) main changes are:

- **Large object handling**

This new feature enables synchronization of object whose size exceeds that which can be transmitted within one message. To achieve synchronization, that object is split into chunks that fit within the message and a new tag <MoreData/> is added to the message to signal to the recipient that the data item is incomplete and has further chunks to come. See [DSPRO_V1_1_2].

- **Busy signaling**

This feature is not mandatory anymore for the server. See [DSPRO_V1_1_2].

- **Authentication**

The protocol allows 4 additional authentication schemes: X.509 Certificate, SecurID, SafeWord, DigiPass. See [REPPRO_V1_1_2].

- **Number of changes**

The new tag <NumberOfChanges> enables the sender to indicate the total number of changes (number of Add, Replace and Delete commands) to be sent to the recipient during the session, so that the recipient may use this information to calculate progress information. See [REPPRO_V1_1_2].

- **MaxObjSize**

The new tag <MaxObjSize> specifies the maximum size of data object that the device is able to receive. See [METAINF_V1_1_2].

- **UTC**

The new tag <UTC> specifies that the device supports UTC based time. See [DEVINFO_V1_1_2].

6.1.3 OMA DS 1.2

The OMA Data Synchronization 1.2 main changes are:

- **Suspend and Resume**

This new feature provides a means to suspend intentionally and resume a synchronization. This kind of interruption occurs when user request's to pause the current session. The interruption of synchronization can be done at any phase of the session. If the server accepts the interruption, the session can be terminated by disconnecting the transport. But since it's an interruption, anchors must not be updated, and session can be resumed later. A session can be resumed whether it was interrupted intentionally or not. The client and server do not need to re-send data that were successfully exchanged before interruption. See [DSPRO_V1_2].

- **Server Alerted Notification**

This feature was completely reconsidered. It enhances the Server Alerted Sync synchronization type, hardly implementable. Many devices cannot (or do not wish) continuously listen for connections from a server, but most devices can receive unsolicited packages called “Notifications” (for examples handsets can receive SMS packages). Therefore, the server alerted sync is not anymore initiated by sending a sync alert within a SyncML message to the client, but a by sending a Server Alerted Notification package. The general structure of this new notification package is described in [SAN] and the notification body specific to Data Synchronization is described in [DSPRO_V1_2]. Actually, the notification Package comprises three parts: a notification header, notification body and a digest of the whole notification. The notification header contains two interesting informations which are:

ui-mode: specifies the server recommendations as to whether the server wants the session to be executed in the background or show an indicator to the user, or ask user for acceptance ;

initiator: specifies how the server has interpreted the initiation of the sending of the package, either because the end user requested it or because the server has actions to perform.

- **Data Sync Record and Field Level Filtering**

Data Sync Record and Field Level filtering is replacing Target Address Filtering, to enable the filtering of data items to synchronize. Target Address Filtering was specified with CGI scripting and restricted to the value of Target LocUri element type. To enhance this feature, new tags are added to SyncML Representation: a new Filter element including other new elements (Field, FilterType, Record) and more precisions on how to synchronize the data subsets resulting from filter, and how to handle with the data out of criteria. The filter criteria can apply to data subsets or to data field subsets. New tags are added to the Devinf 1.2 too (Filter-Rx and Filter-Cap) to indicate support for filtering.

See [DSREPU_V1_2] and [DEVINFO_V1_2].

- **Field Level Replace**

The Replace command may be partial or full, whereas in past version no precision was mentioned. The meta information “FieldLevel” is used to indicate that a Replace command is partial. If the FieldLevel flag is present in a Replace command, the receiver must not remove any fields of the item in the database that are not present in data element associated with the command sent. See [DSREPU_V1_2] and [METAINF_V1_2].

- **Synchronization of hierarchical data objects**

New tags concern synchronization of hierarchical data objects. The new “SourceParent” and “TargetParent” tags provide parent information of the child that is mentioned in the Source LocURI or Target LocURI of sync commands, if and only if the objects have hierarchical nature. The new “Move” command allows moving items (ex: files, folder, emails, vCards) from current location to a new location. See [DSREPU_V1_2].

- **Folder Data Objects**

A new document has been proposed within the OMA-DS Specification, which is Folder Data Objects Specification [FOLDEROBJ]. This document describes folder data objects that can be used to organize objects such as email, file, other folder objects, etc. in a hierarchical manner. It indicates also how data folder properties can be used to filter synchronization (as it is already done with vCard and vCal properties). Folder data objects can be added, replaced, deleted or moved as well as other type of objects.

- **Email Data Objects**

A new document has been proposed within the OMA-DS Specification, which is Email Data Objects Specification [EMAILOBJ]. That document describes email data objects that can be used to represent an interpersonal electronic mail object. It encapsulates the format of electronic messages defined in rfc822 and rfc2822, and specifies new fields such as “read”, “forwarded”, “replied”, “received” etc …The document explains also how it can be used to filter synchronization.

- **File Data Objects**

A new document has been proposed within the OMA-DS Specification, which is File Data Objects Specification [FILEOBJ]. That document describes folder data objects that can be used to represent file which can be stored in various file systems such as FAT, HFS, UFS, etc... The document explains also how it can be used to filter synchronization.

6.2 OMA DS feature Changes from 1.1.2 to 1.2

Theme	Reference	Description	Type	Device	M/O
Suspend & Resume	[DSPRO_V1_2] ch.6.12	Support of suspend/resume	New	Client Server	O M
Field Level Replace	[DSREPU_V1_2] ch.5.8.1 ch. 6.5.12 [METAINF_V1_2] ch.5.2.3 [DEVINFO_V1_2] ch.5.3.11	Support of Field Level Replace	New	Client Server	O O
Filtering	[DSREPU_V1_2] ch.5.13	Support of Data Sync Record and Field Level Filtering	New	Client Server	O O
Filtering	[DSREPU_V1_2] ch.6.1.7 to 6.1.9	Support of Field, Filter and FilterType elements	New	Client Server	O O
Filtering	[DEVINFO_V1_2] ch.5.3.12 to 5.3.14	Support of Filter-Rx, FilterCap and FilterKeyword elements	New	Client Server	O O
Devinfo	[DEVINFO_V1_2] ch.5.3.27, 5.3.29	Support of Property and PropParam elements	New	Client Server	M M
Devinfo	[DEVINFO_V1_2] ch.5.3.21, ch.5.3.24	Support of MaxOccur and NoTruncate elements	New	Client Server	O M
Devinfo	[DEVINFO_V1_2] ch.5.3.21, ch.5.3.22 and 5.3.24	Support of MaxSize element. This element replaces the "Size" element that was removed from devinfo DTD.	New	Client Server	O M
Hierarchical Data Sync	[DSREPU_V1_2] ch.6.1.25, ch.6.1.28 and ch.6.5.10 [DEVINFO_V1_2] ch.5.3.34	Support of Synchronization of hierarchical data objects : support of SourceParent, TargetParent elements and Move Command	New	Client Server	O O
Email Data Objects	[EMAILOBJ]	Support for x-email media type (application/vnd.omads-email)	New	Client Server	O O
File Data Objects	[FILEOBJ]	Support for x-file media type (application/vnd.omads-file)	New	Client Server	O O
Folder Data Objects	[FOLDEROBJ]	Support for x-folder media type (application/vnd.omads-folder)	New	Client Server	O O
Sync Alert	[SAN], [DSPRO_V1_2] ch.12	Support for Sever Alerted Notification	Improve	Client Server	O O
Sync Alert	[DSPRO_V1_2] Appendix A, SCR-DS-client014	Support WAP Push operation, for WAP capable devices that use HTTP or WSP transport for DS	New	Client	O

6.3 OMA DS tag Changes from 1.1.2 to 1.2

Theme	Reference	Description	Type
DataType	[DEVINFO_V1_2] ch.5.3.4	Changed restrictions of Datatype use	Improve
DevId	[DEVINFO_V1_2] ch.5.3.5	Changed use for Server Alerted Notification	Improve
DisplayName	[DEVINFO_V1_2] ch.5.3.8	Changed Parent Elements	Improve
Format	[METAINF_V1_2] ch.5.2.4	Restrictions added	Improve
PropName	[DEVINFO_V1_2] ch.5.3.28	Changed Parent Elements	Improve
Size	[METAINF_V1_2] ch.5.2.16	Restrictions added	Improve
Type	[METAINF_V1_2] ch.5.2.17	Restrictions added	Improve
ValEnum	[DEVINFO_V1_2] ch.5.3.43	Changed Parent Elements	Improve
VerCT	[DEVINFO_V1_2] ch.5.3.44	Changed Parent Elements	Improve
Version	[METAINF_V1_2] ch.5.2.18	Restrictions added	Improve

6.4 OMA DS conformance Changes from 1.1.2 to 1.2

6.4.1 Client Conformance Changes

	Reference V1.1.2	Conformance V1.1.2	Reference V1.2	Conformance V1.2
Device information	[DEVINFO_V1_1_2]		[DEVINFO_V1_2]	
Support of “FwV” element	Ch.5.3.11	O	Ch.5.3.15	M
Support of “HwV” element	Ch.5.3.12	O	Ch.5.3.16	M
Support of “Man” element	Ch.5.3.13	O	Ch.5.3.17	M
Support of “Mod” element	Ch.5.3.17	O	Ch.5.3.23	M
Support of “SwV” element	Ch.5.3.28	O	Ch.5.3.37	M
Meta information	[METAINF_V1_1_2]		[METAINF V1_2]	
Support of “MaxObjSize”	Ch.5.2.9	M	Ch.5.2.10	O
Synchronization Protocol	[DSPRO_V1_1_2]		[DSPRO_V1_2]	
Support of “Sync Alert”	Ch.13	O	Ch.12	O * / M **

(*) for non WAP clients, or WAP clients that only use OBEX transport for DS

(**) by WAP Push method, for WAP capable devices that use HTTP or WSP transport for DS

6.4.2 Server Conformance Changes

	Reference V1.1.2	Conformance V1.1.2	Reference V1.2	Conformance V1.2
Device information	[DEVINFO_V1_1_2]		[DEVINFO_V1_2]	
Support of “FwV” element	Ch.5.3.11	O	Ch.5.3.15	M
Support of “HwV” element	Ch.5.3.12	O	Ch.5.3.16	M
Support of “Man” element	Ch.5.3.13	O	Ch.5.3.17	M
Support of “Mod” element	Ch.5.3.17	O	Ch.5.3.23	M
Support of “SwV” element	Ch.5.3.28	O	Ch.5.3.37	M
Meta information	[METAINF_V1_1_2]		[METAINF _V1_2]	
-	-	-	-	-
Synchronization Protocol	[DSPRO_V1_1_2]		[DSPRO_V1_2]	

Support of “One-way sync from client only” sync type	Ch.11	O	Ch.10	M
Support of “Refresh sync from client only” sync type	Ch.11.3	O	Ch.10.3	M
Support of “One-way sync from server only” sync type	Ch.12	O	Ch.11	M
Support of “Refresh sync from server only” sync type	Ch.12.5	O	Ch.11.5	M

6.5 OMA DS new codes in 1.2

6.5.1 New Alert Codes

See [DSREPU_V1_2] section 7, or [DSPRO_V1_2] section 13.2.

Theme	Alert Code	Name	Description
Suspend & Resume	224	SUSPEND	Suspend Synchronization Session
Suspend & Resume	225	RESUME	Resume Synchronization Session

6.5.2 New Status Codes

See [REPPRO_V1_2] section 10.

Theme	Alert Code	Description
Fied Level Replace	426	Partial item not accepted. Receiver of status code MAY resend the whole item in next package.
Hierarchical Data Sync	427	Item not empty. Parent cannot be deleted since it contains children
Hierarchical Data Sync	428	Move Failed

Appendix A. Change History (Informative)

Document Identifier	Date	Sections	Description
OMA-WP-DataSyncChanges-20040418-D	18 Oct 2004	all	initial version of WP as permanent doc
OMA-WP-SyncML-Primer-20050113-D	13 Jan 2005	all	Modification/ to become a Primer.
OMA-WP-SyncML-Primer-20050712-D	12 July 2005	all	New edition of the Primer/ minor 1.1.2- 1.2 changes addition
OMA-WP-SyncML-ChangeHistory-20060209-D	2 Sep 2005	All	Document re-organization and minor edits (split from [PRIMER])
OMA-WP-SyncML-ChangeHistory-20060116-D	16 Jan 2006	All	Pre-final draft of the document
OMA-WP-SyncML-ChangeHistory-20060209-D	9 feb 2006	All	Implementation of CR 2006-27R01 - minor edits
OMA-WP-SyncML-ChangeHistory-20060316-D	16 Mar 2006	Footer	Year corrected in the footer and copyright clause.
OMA-WP-SyncML-ChangeHistory-20070718-D	18 Jul 2007	All	Ediorial corrections
OMA- WP-SyncML_ChangeHistory-20071002-C	02 Oct 2007	n/a	Status changed to candidate, TP R&A from 2007-09-19 to 2007-10-02. TP doc # OMA-TP-2007-0344-INP_SyncML_Primer_V1_0_RRP_for_Candidate_Approval