



A Primer to SyncML/OMA DS

Candidate – 02 Oct 2007

Open Mobile Alliance
OMA-WP-SyncML_Primer-20071002-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.

© 2007 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	4
2. REFERENCES	5
3. TERMINOLOGY AND CONVENTIONS	7
3.1 CONVENTIONS	7
3.2 DEFINITIONS.....	7
3.3 ABBREVIATIONS	7
4. INTRODUCTION	8
5. SYNCML OVERVIEW	9
5.1 MOBILE COMPUTING AND DATA SYNCHRONIZATION CHALLENGES.....	9
5.2 SYNCML PROTOCOL SNAPSHOT.....	9
5.2.1 SyncML protocol specifications	9
5.2.2 SyncML key aspects	10
5.2.3 Other SyncML features	13
5.2.4 SyncML types of synchronization	14
5.3 OMA DS DELIVERABLES	15
5.3.1 Documentation	15
6. THE BENEFITS OF OMA DS	17
APPENDIX A. CHANGE HISTORY (INFORMATIVE).....	18
APPENDIX B. A COMPLETE EXAMPLE OF A SYNCML SESSION (INFORMATIVE).....	19

1. Scope

This document is designed to introduce OMA Data Synchronization standard, also known as SyncML or OMA DS, and define its capabilities. The reader will also find in this document a brief history of the standard and an outline of its technical and commercial benefits. For further information about OMA DS standard change history, please refer to [HISTORY]. For more in-depth information on OMA DS, the reader will find references of books and specification documents in the next section.

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

2. References

- [DEVINF] “SyncML Device Information”, Open Mobile Alliance™, OMA-SyncML-DevInfo-V1_2-20040601-C, http://www.openmobilealliance.org/release_program/ds_v12.html
- [DSREPU] “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
- [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
- [HISTORY] “OMA DS Standards Change History”, Open Mobile Alliance™, OMA-WP-SyncML_ChangeHistory-20071002-C, <http://www.openmobilealliance.com/tech/publicmaterial.html>
- [META] “SyncML Meta Information”, Open Mobile Alliance™, OMA-TS-SyncML_MetaInfo-V1_2-20050509-C, http://www.openmobilealliance.org/release_program/SyncML_v12.html
- [METADTD] “SyncML Meta Information, Document Type Definition”, Open Mobile Alliance™, OMA-TS-SyncML_MetaInfo_DTD-V1_2-20050303-C, http://www.openmobilealliance.org/release_program/SyncML_v12.html
- [REPPRO] “SyncML Representation Protocol”, Open Mobile Alliance™, OMA-TS-SyncML_RepPro-V1_2-20050509-C, http://www.openmobilealliance.org/release_program/SyncML_v12.html
- [REPPRODTD] “SyncML Representation Protocol, Document Type Definition”, Open Mobile Alliance™, OMA-TS-SyncML_RepPro_DTD-V1_2-20050303-C, http://www.openmobilealliance.org/release_program/SyncML_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
- [SYNCHTTP] “SyncML HTTP Binding”, Open Mobile Alliance™, OMA-TS-SyncML_HTTPBinding-V1_2-20050509-C, http://www.openmobilealliance.org/release_program/SyncML_v12.html
- [SyncMLBook] SyncML: Synchronizing and Managing Your Mobile Data
by U. Hansmann, R. Mettala, A. Purakayastha, P. Thompson, published by Pearson Education; (Sept 2002)
- [SYNCOBEX] “SyncML OBEX Binding”, Open Mobile Alliance™, OMA-TS-SyncML_OBEXBinding-V1_2-20050509-C,

http://www.openmobilealliance.org/release_program/SyncML_v12.html

[SYNCPRO]

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

[SYNCWSP]

“SyncML WSP Binding”, Open Mobile Alliance™,
OMA-TS-SyncML_WSPBinding-V1_2-20050509-C,
http://www.openmobilealliance.org/release_program/SyncML_v12.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 synchronization protocol that is designed to be data-type and transport agnostic in that it allows any transport (Bluetooth, Infrared, etc.) and enables the synchronization of any data type (contacts, calendar, files, etc.). OMA DS provides the following capabilities for any application developers to implement on client, server or end-to-end platforms:

- 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. 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.

Therefore, this document will propose a short introduction to data synchronization challenges, as well as to SyncML protocol. Furthermore, the reader will find a description of the different specification documents provided in the Enabler Release Package, and some information related to the OMA DS test process.

5. SyncML Overview

5.1 Mobile Computing and Data Synchronization challenges

The popularity of mobile computing and communications devices can be traced to their ability to deliver information to users when needed. Users want ubiquitous access to information and applications from the device at hand; in addition they want to access and update this information on the fly.

The ability to use applications and information on one mobile device, then to synchronize any updates with the applications and information back at the office, or on the network, is a key to the utility and popularity of this pervasive, disconnected way of computing.

Until year 2000, there was a proliferation of different, proprietary data synchronization protocols for mobile devices. Each of these protocols was -and still is - only available for selected transports, implemented on a selected subset of devices, and able to access only a small set of net-worked data.

Unfortunately, without a common and standardized synchronization protocol, we could not achieve these dual visions:

- Networked data that support synchronization with any mobile device
- Mobile devices that support synchronization with any networked data

The absence of a single synchronization standard would pose many problems for end users, device manufacturers, application developers, and service providers.

SyncML is an open synchronization protocol that was first released by SyncML Initiative Ltd. in 2000, to solve this situation. 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.

To simplify the comprehension of the document, we will use the term "SyncML" when dealing with the generic aspects of protocol, and use either "OMA DS" or "SyncML" when appropriate.

For more information on the history of SyncML, SyncML initiative and OMA please refer to the OMA site for [HISTORY] SyncML evolution paper.

5.2 SyncML protocol snapshot

5.2.1 SyncML protocol specifications

SyncML Data Synchronization specifies a common data synchronization framework and XML-based format for synchronizing data on networked devices. SyncML Data Synchronization is designed for use between mobile devices that are intermittently connected to the network and network services that are continuously available on the network, but can also be used for peer-to-peer data synchronization.

It uses XML documents to exchange information about the changes made to data, as well as for the changed data itself. The data representation specifies an XML DTD that allows the representation of all the information required to perform synchronization, including data, metadata and commands (add, replace, delete, move ...).

The synchronization protocol is based on a client/server architecture and specifies how SyncML messages conforming to the DTD are exchanged in order to allow a SyncML client and server to exchange additions, deletions, updates and other status information. Those documents are transferred in a sequence defined by the synchronisation protocol.

SyncML Data Synchronization contains the following main specifications:

- Data Synchronization protocol [SYNCPRO], which defines the actions between a SyncML client and a SyncML server. *This is the first document to read in order to have a more detailed view of SyncML*
- Data Synchronization representation protocol ([DSREPU], [REPPRO], and [REPPRODTD]), which defines the representation format of SyncML messages (in XML) and details the inner workings of the SyncML framework
- Content-specific aspects of synchronization for different data types ([EMAILOBJ] for Email, [FILEOBJ] for File, and [FOLDEROBJ] for Folders)
- Transport bindings for the protocols ([SYNCHTTP] for HTTP, [SYNCOBEX] for OBEX, and [SYNCWSP] for WSP...)

Some of those specifications are part of OMA Common Specifications (which are common to Data Synchronization and Device Management).

In order to build a SyncML-compliant product, you must adhere to the requirements of both **OMA Common Specifications** and **OMA Data Sync Specifications**. See section 5.3 for more details on OMA DS deliverables.

5.2.2 SyncML key aspects

This section introduces the main SyncML Data Synchronization principles. The first subsection emphasizes two fundamental aspects of SyncML: data type independence of the protocol and the concept of internal mechanisms delegation. Then the second subsection presents the different steps of the synchronization as well as the main features of SyncML Data Synchronization which result from this.

5.2.2.1 Independence and delegation

Data Types Independence

OMA DS is both data type and data store independent. OMA DS can carry any data type which can be represented as a MIME object. To promote interoperability between different implementations of this protocol, the specification includes the representation formats used for common PIM data, like contacts or calendar, and since SyncML v1.2: emails, files and folders. It's nevertheless possible to synchronize any other kind of data, using any representation format, by the time the format is MIME registered.

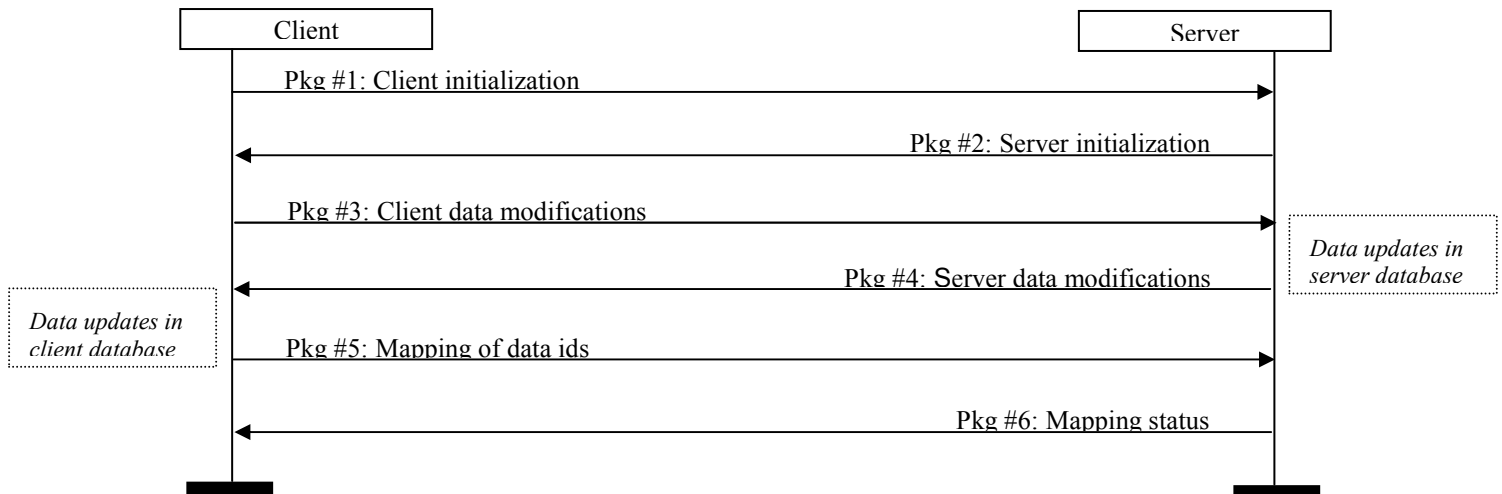
Internal Mechanisms Delegation

OMA DS objective is to define a framework for data synchronization. As a consequence, the protocol does not re-define already existing mechanisms for side features, like authentication. OMA DS clients or servers can delegate those specific functionalities to external modules.

5.2.2.2 Synchronization session

Packages

A synchronization session is composed with the exchange of different packages of information between the client and the server. For example, a two-way synchronization session (see 5.2.4 for further explanations) can be schematized in a sequence of different message packages (a complete example is given in Appendix B):



An **initialization package** generally contains the following information:

- Device capabilities
- Requested databases access, type of sync desired
- Authentication information
- Sync anchors

A **modification package** generally contains the following information:

- Synchronization commands: add, replace, delete, move ...
- Data ids and contents associated with these commands (except for delete)

A **mapping package** (only sent by client) generally contains the following information:

- Mapping information (a couple of LUID and GUID for each new item added from server)

Device capabilities

The protocol provides the functionality to exchange the device capabilities (identifier, sync types supported, datastore names, data format supported for each datastore ...) 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. The client must also be able to transmit its device information upon the server's request. Likewise, the server must be able to send its device information upon the client's request. A SyncML server must be enabled to receive and process device information, regardless of whether it has been sent by the client or requested by the server itself. If the capabilities of the client and server match up, they can proceed into a synchronization session.

Two types of information are exchanged between device and server, as follows:

- Device information includes device type, model number, and manufacturer
- Datastore information represents the capabilities of the data objects supported by the client or server. If the client supports vCard version 2.1 for slow sync and two-way sync, the server must also have this capability; otherwise the sync session cannot proceed for the vCard data object.

The capabilities of the mobile devices and computers over the network may be very different. It is not unusual, due to the limitations of the mobile device, for its applications to contain less data fields per item than similar applications on computers. SyncML will address resource limitations of the devices and make the data as

identical as possible on it, by taking advantages of device information capacity knowledge. Moreover, data exchanged using the protocol may be encoded in a binary format (such as WBXML) to reduce memory requirements.

Authentication

SyncML is intended to provide a framework for secure data synchronization. SyncML itself does not define any new security schemes. The standard is open to delegate this functionality by integrating any external security mechanism. Instead, it provides the framework to challenge authentication and authorization at different layers of the network. SyncML defines a user authentication mechanism for three different levels of the protocol: the server level, the database level, and the object level. SyncML mandates only that its authentication mechanism be supported at the server level. Authentication at the database and object levels is optional.

To be compliant with the SyncML protocol, a SyncML client or server must support both basic and MD5 authentication. Both the 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.

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), from the sending device point of view;
- the Next anchor which describe the current event of synchronization from the sending device point of view.

The client and server must always send their own anchors to each other when a synchronization session is initiated. Each device must also store the Next anchor of the other until the next sync event. By comparing the matching values of Last anchor sent and Next anchor stored, one device is able to conclude that no failures have happened to the other since last sync. If they don't match, the device can request an appropriate action from the other device, such as a slow sync. The stored sync anchor is updated only when the session has ended successfully.

Map operation

The protocol is based on the principle that the client and the server can have their own unique identifiers ID's for data items in their database. The client ID is known as a locally unique identifier (LUID); the server ID is known as a globally unique identifier (GUID). These IDs may or may not be congruent between client and server. Because the ID's can be different, the server must maintain a ID mapping table for all items exchanged between itself and the client, in order to know which client LUID and which server GUID corresponds to the same data item. Then client is requested to send mapping information (the LUIDs of new items sent by the server that are added to its database). Following the Map operation, the server will update its mapping table with the client LUID.

Status Codes

Most of the information sent to a device requires that the recipient sends an acknowledgement for this information back. Acknowledgement is made via "**Status**" command, in order to inform the sender that the information was received. A status code indicates how the information has been processed by recipient. So each package may contain status information.

Multiple SyncML messages

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. Client and server packages might be interlaced: the receiver of a package may start to send its next package at the same time if it makes sense.

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. Here are some of the status codes representing conflict resolution policies:

- 207: Conflict resolved with data merge
- 208: Conflict resolved with client "win"
- 209: Conflict resolved with data duplication

The administration and configuration of the conflict resolution policy is beyond the scope of the SyncML protocol.

5.2.3 Other SyncML features

This section deals with the other features contained in the SyncML standard:

Suspend and Resume

This feature provides a means to suspend intentionally and resume synchronization. This kind of interruption occurs when user requests to pause the current session. The interruption 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.

Large Object Handling

The protocol provides a means to synchronize an object whose size exceeds that which can be transmitted within one message. This is achieved by splitting the object into chunks that will fit within the message and using a special marker to signal to the recipient that the data item is incomplete and has further chunks to come.

Sync without Separate Init

Synchronization can be started without a separate initialization. This means that the initialization is done simultaneously with sync. This can be done to decrease the number of SyncML messages to be sent over the air.

Device Memory Management

This feature provides possibility to specify the dynamic memory capabilities for databases of a device or for persistent storage on a device. The capabilities specify how much memory there is left for usage. The dynamic capabilities can be specified every time when the synchronization is done. The static memory capabilities are exchanged when the sync initialization is done.

Data Sync Record and Field Level Filtering

SyncML provides two means to filter the data that has to be synchronized.

- The **Record level filtering** enables to synchronize with a subset of the data that resides in a datastore. For example, a client can wish to synchronize only contact items that belong to the "Family" category on the server side. The server is expected not to send any new data that doesn't belong to this category.
- The **Field level filtering** enables to synchronize with a subset of the fields of all the data that resides in a datastore. For example, a client can wish not to synchronize the commentary and picture fields of contact items on the server side. The server is expected not to send these two fields within the data during synchronization session.

Synchronization of hierarchical data objects

SyncML enables the synchronization of hierarchical data objects, such as file or email folders.

5.2.4 SyncML types of synchronization

SyncML specifies different synchronization types. Those are meant to evolve in order to add new possibilities of interaction between a client and a server. At the present time, the seven different synchronization types are defined as follows:

Two-way sync (Incremental sync)

Two-way sync is a normal synchronization type in which client and server exchange information about any modifications to the data in their device database.

Each client and server is connected to a database, which contains analogous data that need to be synchronized. The client always initiates this exchange and sends its data modifications to the server first. The server processes these modifications in its own database. The server might have to resolve conflicts, when these changes occur on a data that has been also modified on the server side. Then, the server sends its own data modifications to the client device, which updates its database accordingly. Finally, the client sends back the data id mapping information (for all data items additions asked by the server) necessary to maintain the correspondence between client and server synchronized data.

Slow sync (Full sync)

The slow sync is a form of the two-way synchronization in which all the items in the client database are compared with all the items in the server database on a field-by-field basis. A slow sync is requested by the client or server when the very first synchronization session occurs, or when it has detected that a problem occurred since last synchronization session. In practice, the slow sync means that the client sends all its data to the server, which compares and unifies its own data with that sent by the client. Then, the server returns all the modification information to the client. Finally, the client returns the mapping information.

One-way sync from client only (Export sync)

This is a synchronization type in which the client sends its modifications to the server, but the server does not send its modifications back to the client.

One-way sync from server only (Import sync)

This is a synchronization 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 client only (Backup sync)

This is a synchronization type in which the client sends all its data from a database to the server (i.e., exports). The server is expected to replace all data in the target database with the data sent by the client. This sync type could be used for backup/restore application (in backup mode).

Refresh sync from server only (Restore sync)

This is a synchronization type in which the server sends all its data from a database to the client (i.e., exports). The client is expected to replace all data in the target database with the data sent by the server. This sync type could be used for backup/restore application (in restore mode).

Server-alerted sync

This is a synchronization 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. This sync type allows on-demand/on-event synchronization from networked services. As an example, a user synchronizing his handset address book with a networked address book could launch a synchronization session from the networked address book interface.

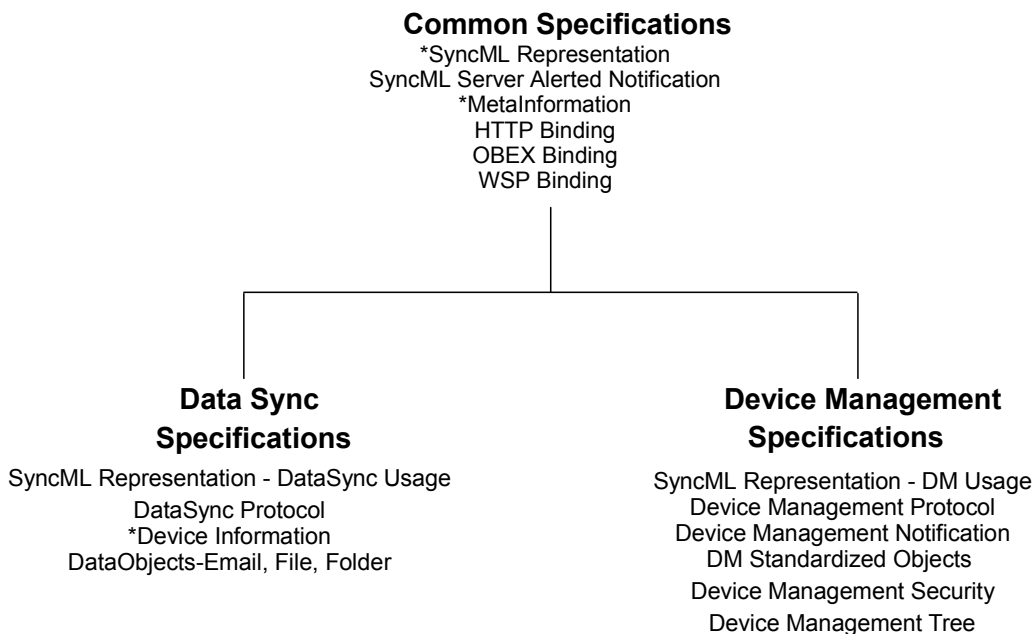
The following table summarizes those different synchronization types:

	ONE-WAY	TWO-WAY
FULL		Full sync
INCREMENTAL	Export sync – Import sync	Incremental sync
REFRESH	Backup sync – Restore sync	-

5.3 OMA DS deliverables

5.3.1 Documentation

The relationship between the documents which had been created during the SyncML Initiative has been preserved and is depicted in



The OMA Data Synchronization specifications are based on the SyncML Initiative’s Data Synchronization specifications and make use of the OMA SyncML Common specifications.

The OMA SyncML Common Specifications Enabler Release includes the following documents:

- SyncML Representation [REPPRO]: The XML-based representation protocol which specifies the common XML syntax and semantics used by all SyncML protocols. This defines the superset of the DS and DM representation protocols. (* includes DTD [REPPRODTD]).
- The transport bindings: HTTP [SYNCHTTP], OBEX [SYNCOBEX], WSP [SYNCWSP]. These specify the features REQUIRED for each transport to send and receive DS and DM protocol messages.

- The Meta Information [META] associated with a SyncML command or data item or collection used by either DS or DM (* includes DTD [METADTD])
- SyncML Server Alerted Notification [SAN]: The logical structure and format of the notification messages used by all SyncML server alerted notifications, for both DS and DM.

The OMA Data Synchronization Specifications Enabler Release includes the following documents:

- SyncML Representation DataSync Usage [DSREPU]: The subset of the Common Specifications SyncML Representation Specification necessary to define the Data Synchronization commands and protocol, with examples and commentary specific to DS.
- DataSyncProtocol [SYNCPRO]: Specifies how SyncML Common messages conforming to the DTD are exchanged in order to allow an OMA DS client and server to exchange additions, deletions, updates and other status information.
- Device Information [DEVINF]: Used to exchange device specific information, including hardware, firmware, software levels, available memory, and local databases supported. (* Includes DTD)
- Data Objects: Email [EMAILOBJ], File [FILEOBJ], Folder [FOLDEROBJ]: Each object is identified by a unique MIME media type (e.g. application/vnd.omads-email). The objects are either represented by or encapsulated in a mark-up language defined by XML. Meta or state data is included in the representation (e.g. Read/Unread, Creation Date, and Last Modified Date).

6. The benefits of OMA DS

OMA DS is designed to be data-type and network agnostic. Beyond this scalable modular purpose, OMA DS brings a number of benefits to application developers. To enumerate a few:

- OMA DS allows synchronization of any data type. Current industry implementations have been deployed for the following data types:
 - Java credit card records
 - Medical, doctor, patient records
 - PIM data: Contacts, Calendar, Tasks / To Dos, Memos / Notes
 - Email with any attachment type
 - Rich Media: SMS messages, MMS message, pictures, videos, music, bookmarks
- OMA DS is designed to operate on any mobile network and is radio frequency agnostic. It has been deployed on the following: GSM, CDMA, CDMA 2000, GPRS, EDGE, FOMA, 3G, TDMA, W-CDMA, UMTS, etc.
- Based on XML-messaging exchanges, OMA DS is designed to be transport agnostic. It has been tested with the following transports: HTTP, WSP (Wireless Session Protocol), OBEX (Bluetooth, IrDA), SMTP, pure TCP/IP networks, and proprietary communication protocols.
- OMA DS is a several years old mature protocol (first release in 2001) and has been adopted by hundreds of device manufacturers, application developers and network operators.
- OTA OMA DS services are offered by mobile operators in every continent as evidenced by global announcements of service availability.

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 Jul 2005	all	New edition of the Primer/ minor 1.1.2- 1.2 changes addition
OMA-WP-SyncML_Primer-20051011-D	11 Oct 2005	all	Split on this paper into 2 papers and addition of key OMA DS benefits also updates throughout the document
OMA-WP-SyncML_Primer-20060116-D	16 Jan2006	all	renaming (changing the date) the doc to upload it on the permanent section
OMA-WP-SyncML_Primer-20060209-D	09 Feb 2006	all	Implementation of CR OMA-DS-2006-0026R01 - Minor (typo) corrections, link to SyncML books added
OMA-WP-SyncML_Primer-20060316-D	16 Mar 2006	Copyright clause	Year corrected to 2006
OMA-WP-SyncML_Primer-20060612-D	12 Jun 2006	5.4, 6, 5.2.4	revision after 1 st R&A : implementation of OMA-DS-PrimerWP-2006-0001-CR_Primer
OMA-WP-SyncML_Primer-20060911-D	11 Sep 2006	Appendix B	Updated according to agreed CR : OMA-DS-PrimerWP-2006-0003
OMA-WP-SyncML_Primer-20070718-D	18 Jul 2007	All	Editorial updates
OMA-WP-SyncML_Primer-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

Appendix B. A complete example of a SyncML session (Informative)

Here is a complete example of a SyncML session, in the case of a two-way synchronization. All packages examples are extracted from [SYNCPRO]. The numbering of the packages refers to the chart depicted in the section 5.2.2.

Package 1: Sync Initialization Package from Client

```
<SyncML>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>SyncML/1.2</VerProto>
    <SessionID>4</SessionID>
    <MsgID>1</MsgID>
    <Target><LocURI>http://www.syncml.org/sync-server</LocURI></Target>
    <Source><LocURI>IMEI:493005100592800</LocURI></Source>
    <Cred> <!--The authentication is optional. -->
      <Meta><Type xmlns='syncml:metinf'>syncml:auth-basic</Type></Meta>
      <Data>QnJ1Y2UyOk9oQmVoYXZI</Data> <!--base64 formatting of "userid:password"-->
    </Cred>
    <Meta> <!--The Meta is now used to indicate the maximum SyncML message size, which client can receive.-->
      <MaxMsgSize xmlns='syncml:metinf'>5000</MaxMsgSize>
    </Meta>
  </SyncHdr>
  <SyncBody>
    <Alert>
      <CmdID>1</CmdID>
      <Data>200</Data> <!-- 200 = TWO_WAY_ALERT -->
      <Item>
        <Target><LocURI>./contacts/james_bond</LocURI></Target>
        <Source><LocURI>./dev-contacts</LocURI></Source>
        <Meta>
          <Anchor xmlns='syncml:metinf'>
            <Last>234</Last>
            <Next>276</Next>
          </Anchor>
        </Meta>
      </Item>
    </Alert>
    <Put>
      <CmdID>2</CmdID>
      <Meta><Type xmlns='syncml:metinf'>application/vnd.syncml-devinf+xml</Type></Meta>
      <Item>
        <Source><LocURI>./devinf12</LocURI></Source>
        <Data>
          <DevInf xmlns='syncml:devinf'>
            <Man>Big Factory, Ltd.</Man>
            <Mod>4119</Mod>
            <OEM>Jane's phones</OEM>
            <FwV>2.0e</FwV>
            <SwV>2.0</SwV>
            <HwV>1.22I</HwV>
            <DevId>1218182THD000001-2</DevId>
            <DevTyp>phone</DevTyp>
            <UTC/>
            <SupportLargeObjs/>
            <SupportNumberOfChanges/>
            <DataStore>
              <SourceRef>./contacts</SourceRef>
              <DisplayName>Phonebook</DisplayName>
              <MaxGUIDSize>32</MaxGUIDSize>
              <Rx-Pref>

```

```

        <CTType>text/x-vcard</CTType>
        <VerCT>2.1</VerCT>
    </Rx-Pref>
    <Tx-Pref>
        <CTType>text/x-vcard</CTType>
        <VerCT>2.1</VerCT>
    </Tx-Pref>
    <CTCap>
        <CTType>text/x-vcard</CTType>
        <VerCT>2.1</VerCT>
        <Property>
            <PropName>BEGIN</PropName>
            <ValEnum>VCARD</ValEnum>
        </Property>
        <Property>
            <PropName>END</PropName>
            <ValEnum>VCARD</ValEnum>
        </Property>
        <Property>
            <PropName>VERSION</PropName>
            <ValEnum>2.1</ValEnum>
        </Property>
        <Property>
            <PropName>N</PropName>
        </Property>
        <Property>
            <PropName>TEL</PropName>
            <MaxOccur>5</MaxOccur>
            <PropParam>
                <ParamName>TYPE</ParamName>
                <ValEnum>VOICE, CELL</ValEnum>
                <ValEnum>VOICE, HOME</ValEnum>
            </PropParam>
        </Property>
        <Property>
            <PropName>NOTE</PropName>
            <MaxOccur>1</MaxOccur>
            <MaxSize>255</MaxSize>
            <NoTruncate/>
        </Property>
        <Property>
            <PropName>PHOTO</PropName>
            <MaxOccur>1</MaxOccur>
            <PropParam>
                <ParamName>TYPE</ParamName>
                <ValEnum>JPEG</ValEnum>
            </PropParam>
        </Property>
    </CTCap>
    <SyncCap>
        <SyncType>01</SyncType>
        <SyncType>02</SyncType>
        <SyncType>07</SyncType>
    </SyncCap>
</DataStore>
</DevInf>
</Data>
</Item>
</Put>
<Get>
    <CmdID>3</CmdID>
    <Meta><Type xmlns='syncml:metinf'>application/vnd.syncml-devinf+xml</Type></Meta>
</Item>

```

```

                <Target><LocURI>./devinf12</LocURI></Target>
            </Item>
        </Get>
    </Final/>
</SyncBody>
</SyncML>

```

Package 2: Sync Initialization Package from Server

```

<SyncML>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>SyncML/1.2</VerProto>
    <SessionID>4</SessionID>
    <MsgID>1</MsgID>
    <Target><LocURI>IMEI:493005100592800</LocURI></Target>
    <Source><LocURI>http://www.syncml.org/sync-server</LocURI></Source>
    <Cred> <!--The authentication is optional.-->
      <Meta><Type xmlns='syncml:metinf'>syncml:auth-basic</Type></Meta>
      <Data>QnJ1Y2UyOk9oQmVoYXZl</Data> <!--base64 formatting of "userid:password"-->
    </Cred>
  </SyncHdr>
  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>1</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
      <TargetRef>http://www.syncml.org/sync-server</TargetRef>
      <SourceRef>IMEI:493005100592800</SourceRef>
      <Data>212</Data> <!--Statuscode for OK, authenticated for session-->
    </Status>
    <Status>
      <CmdID>2</CmdID>
      <MsgRef>1</MsgRef><CmdRef>1</CmdRef><Cmd>Alert</Cmd>
      <TargetRef>./contacts/james_bond</TargetRef>
      <SourceRef>./dev-contacts</SourceRef>
      <Data>200</Data> <!--Statuscode for OK-->
      <Item>
        <Data><Anchor xmlns='syncml:metinf'><Next>276</Next></Anchor></Data>
      </Item>
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>1</MsgRef><CmdRef>2</CmdRef><Cmd>Put</Cmd>
      <SourceRef>./devinf12</SourceRef>
      <Data>200</Data> <!--Statuscode for OK-->
    </Status>
    <Results>
      <CmdID>4</CmdID>
      <MsgRef>1</MsgRef><CmdRef>3</CmdRef>
      <Meta><Type xmlns='syncml:metinf'>application/vnd.syncml-devinf+xml</Type></Meta>
      <Item>
        <Source><LocURI>./devinf12</LocURI></Source>
        <Data>
          <DevInf xmlns='syncml:devinf'>
            <Man>Small Factory, Ltd.</Man>
            <Mod>Tiny Server</Mod>
            <OEM>Tiny Shop</OEM>
            <DevId>485749KR</DevId>
            <DevTyp>Server</DevTyp>
            <UTC/>
            <SupportLargeObjs/>
            <SupportNumberOfChanges/>
            <DataStore>

```

```

<SourceRef>./contacts</SourceRef>
<DisplayName>Addressbook</DisplayName>
<Rx-Pref>
  <CTType>text/x-vcard</CTType>
  <VerCT>2.1</VerCT>
</Rx-Pref>
<Rx>
  <CTType>text/vcard </CTType>
  <VerCT>3.0</VerCT>
</Rx>
<Tx-Pref>
  <CTType>text/x-vcard</CTType>
  <VerCT>2.1</VerCT>
</Tx-Pref>
<Tx>
  <CTType>text/vcard</CTType>
  <VerCT>3.0</VerCT>
</Tx>
<CTCap>
  <CTType>text/x-vcard</CTType>
  <VerCT>2.1</VerCT>
  <Property>
    <PropName>BEGIN</PropName>
    <ValEnum>VCARD</ValEnum>
  </Property>
  <Property>
    <PropName>END</PropName>
    <ValEnum>VCARD</ValEnum>
  </Property>
  <Property>
    <PropName>VERSION</PropName>
    <ValEnum>2.1</ValEnum>
  </Property>
  <Property>
    <PropName>N</PropName>
  </Property>
  <Property>
    <PropName>TEL</PropName>
    <MaxOccur>8</MaxOccur>
    <PropParam>
      <ParamName>TYPE</ParamName>
      <ValEnum>VOICE, CELL</ValEnum>
      <ValEnum>VOICE, HOME</ValEnum>
      <ValEnum>FAX, HOME</ValEnum>
    </PropParam>
  </Property>
  <Property>
    <PropName>NOTE</PropName>
    <MaxOccur>1</MaxOccur>
    <MaxSize>1024</MaxSize>
    <NoTruncate/>
  </Property>
  <Property>
    <PropName>PHOTO</PropName>
    <MaxOccur>1</MaxOccur>
    <PropParam>
      <ParamName>TYPE</ParamName>
      <ValEnum>JPEG</ValEnum>
      <ValEnum>GIF</ValEnum>
    </PropParam>
  </Property>
</CTCap>
<CTCap>

```

```

<CTType>text/vcard</CTType>
<VerCT>3.0</VerCT>
<Property>
  <PropName>BEGIN</PropName>
  <ValEnum>VCARD</ValEnum>
</Property>
<Property>
  <PropName>END</PropName>
  <ValEnum>VCARD</ValEnum>
</Property>
<Property>
  <PropName>VERSION</PropName>
  <ValEnum>3.0</ValEnum>
</Property>
<Property>
  <PropName>N</PropName>
</Property>
<Property>
  <PropName>TEL</PropName>
  <MaxOccur>8</MaxOccur>
  <PropParam>
    <ParamName>TYPE</ParamName>
    <ValEnum>VOICE, CELL</ValEnum>
    <ValEnum>VOICE, HOME</ValEnum>
    <ValEnum>FAX, HOME</ValEnum>
  </PropParam>
</Property>
<Property>
  <PropName>NOTE</PropName>
  <MaxOccur>1</MaxOccur>
  <MaxSize>1024</MaxSize>
  <NoTruncate/>
</Property>
<Property>
  <PropName>PHOTO</PropName>
  <MaxOccur>1</MaxOccur>
  <PropParam>
    <ParamName>TYPE</ParamName>
    <ValEnum>JPEG</ValEnum>
    <ValEnum>GIF</ValEnum>
  </PropParam>
</Property>
</CTCap>
<SyncCap>
  <SyncType>01</SyncType>
  <SyncType>02</SyncType>
  <SyncType>03</SyncType>
  <SyncType>04</SyncType>
  <SyncType>05</SyncType>
  <SyncType>06</SyncType>
  <SyncType>07</SyncType>
</SyncCap>
</DataStore>
</DevInf>
</Data>
</Item>
</Results>
<Alert>
  <CmdID>5</CmdID>
  <Data>200</Data> <!-- 200 = TWO_WAY_ALERT -->
</Item>
  <Target><LocURI>./dev-contacts</LocURI></Target>
  <Source><LocURI>./contacts/james_bond</LocURI></Source>

```

```

        <Meta>
          <Anchor xmlns='syncml:metinf'>
            <Last>20040119T081812Z </Last>
            <Next>20040120T093223Z </Next>
          </Anchor>
        </Meta>
      </Item>
    </Alert>
  </Final/>
</SyncBody>
</SyncML>

```

Package 3: Client Sending Data Modifications to Server

```

<SyncML>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>SyncML/1.2</VerProto>
    <SessionID>4</SessionID>
    <MsgID>2</MsgID>
    <Target><LocURI>http://www.syncml.org/sync-server</LocURI></Target>
    <Source><LocURI>IMEI:493005100592800</LocURI></Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>1</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
      <TargetRef>IMEI:493005100592800</TargetRef>
      <SourceRef> http://www.syncml.org/sync-server</SourceRef>
      <Data>212</Data> <!--Statuscode for OK, authenticated for session-->
    </Status>
    <Status>
      <CmdID>2</CmdID>
      <MsgRef>1</MsgRef><CmdRef>5</CmdRef><Cmd>Alert</Cmd>
      <TargetRef>./dev-contacts</TargetRef>
      <SourceRef>./contacts/james_bond</SourceRef>
      <Data>200</Data> <!--Statuscode for Success-->
    </Status>
    <Item>
      <Data>
        <Anchor xmlns='syncml:metinf'><Next>20040120T093223Z </Next></Anchor>
      </Data>
    </Item>
  </Status>
  <Sync>
    <CmdID>3</CmdID>
    <Target><LocURI>./contacts/james_bond</LocURI></Target>
    <Source><LocURI>./dev-contacts</LocURI></Source>
    <Meta>
      <Mem xmlns='syncml:metinf'>
        <FreeMem>8100</FreeMem><!--Free memory (bytes) in Calendar database on a device -
->
        <FreeId>81</FreeId><!--Number of free records in Calendar database-->
      </Mem>
    </Meta>
    <NumberOfChanges>1</NumberOfChanges>
    <Replace>
      <CmdID>4</CmdID>
      <Meta><Type xmlns='syncml:metinf'>text/x-vcard</Type></Meta>
      <Item>
        <Source><LocURI>1012</LocURI></Source>
        <Data>
          <!-- ... -->
          <!-- ... -->
        </Data>
      </Item>
    </Replace>
  </Sync>
</SyncBody>
</SyncML>

```



```

                                <!--The vCard data would be placed here.-->
                                <!-- ... -->
                                <!-- ... -->
                            </Data>
                        </Item>
                    </Replace>
                </Sync>
            </Final/>
        </SyncBody>
    </SyncML>

```

Package 4: Server Sending Data Modifications to Client

```

<SyncML>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>SyncML/1.2</VerProto>
    <SessionID>4</SessionID>
    <MsgID>2</MsgID>
    <Target><LocURI>IMEI:493005100592800</LocURI></Target>
    <Source><LocURI>http://www.syncml.org/sync-server</LocURI></Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>2</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
      <TargetRef>http://www.syncml.org/sync-server</TargetRef>
      <SourceRef>IMEI:493005100592800</SourceRef>
      <Data>200</Data>
    </Status>
    <Status><!--This is a status for the client modifications to the server.-->
      <CmdID>2</CmdID>
      <MsgRef>2</MsgRef><CmdRef>3</CmdRef><Cmd>Sync</Cmd>
      <TargetRef>./contacts/james_bond</TargetRef>
      <SourceRef>./dev-contacts</SourceRef>
      <Data>200</Data> <!--Statuscode for Success-->
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>2</MsgRef><CmdRef>4</CmdRef><Cmd>Replace</Cmd>
      <SourceRef>1012</SourceRef>
      <Data>200</Data> <!--Statuscode for Success-->
    </Status>
    <Sync>
      <CmdID>4</CmdID>
      <Target><LocURI>./dev-contacts</LocURI></Target>
      <Source><LocURI>./contacts/james_bond</LocURI></Source>
      <NumberOfChanges>2</NumberOfChanges>
      <Replace>
        <CmdID>5</CmdID>
        <Meta><Type xmlns='syncml:metinf'>text/x-vcard</Type></Meta>
        <Item>
          <Target><LocURI>1023</LocURI></Target>
          <Data>
            <!-- ... -->
            <!-- ... -->
            <!--The vCard data would be placed here.-->
            <!-- ... -->
            <!-- ... -->
          </Data>
        </Item>
      </Replace>
    </Add>
  </SyncBody>
</SyncML>

```

```

        <CmdID>6</CmdID>
        <Meta><Type xmlns='syncml:metinf'>text/x-vcard</Type></Meta>
        <Item>
            <Source><LocURI>10536681</LocURI></Source>
            <Data>
                <!-- ... -->
                <!-- ... -->
                <!--The vCard data would be placed here.-->
                <!-- ... -->
                <!-- ... -->
            </Data>
        </Item>
    </Add>
</Sync>
<Final/>
</SyncBody>
</SyncML>

```

Package 5: Client Sending Data Update Status to Server (mapping of data ids)

```

<SyncML>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>SyncML/1.2</VerProto>
    <SessionID>4</SessionID>
    <MsgID>3</MsgID>
    <Target><LocURI>http://www.syncml.org/sync-server</LocURI></Target>
    <Source><LocURI>IMEI:493005100592800</LocURI></Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>2</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
      <TargetRef>IMEI:493005100592800</TargetRef>
      <SourceRef> http://www.syncml.org/sync-server</SourceRef>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>2</CmdID>
      <MsgRef>2</MsgRef><CmdRef>4</CmdRef><Cmd>Sync</Cmd>
      <TargetRef> ./dev-contacts</TargetRef>
      <SourceRef> ./contacts/james_bond</SourceRef>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>2</MsgRef><CmdRef>5</CmdRef><Cmd>Replace</Cmd>
      <TargetRef> 1023</TargetRef>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>4</CmdID>
      <MsgRef>2</MsgRef><CmdRef>6</CmdRef><Cmd>Add</Cmd>
      <SourceRef> 10536681</SourceRef>
      <Data>200</Data>
    </Status>
    <Map>
      <CmdID>5</CmdID>
      <Target><LocURI> ./contacts/james_bond</LocURI></Target>
      <Source><LocURI> ./dev-contacts</LocURI></Source>
      <MapItem>
        <Target><LocURI> 10536681</LocURI></Target>
        <Source><LocURI> 1024</LocURI></Source>
      </MapItem>
    </Map>
  </SyncBody>
</SyncML>

```

```

        </MapItem>
      </Map>
    </Final/>
  </SyncBody>
</SyncML>

```

Package 6: Server Mapping Acknowledgement to Client

```

<SyncML>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>SyncML/1.2</VerProto>
    <SessionID>4</SessionID>
    <MsgID>3</MsgID>
    <Target><LocURI>IMEI:493005100592800</LocURI></Target>
    <Source><LocURI>http://www.syncml.org/sync-server</LocURI></Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>3</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
      <TargetRef>http://www.syncml.org/sync-server</TargetRef>
      <SourceRef>IMEI:493005100592800</SourceRef>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>3</MsgRef><CmdRef>5</CmdRef><Cmd>Map</Cmd>
      <TargetRef>./contacts/james_bond </TargetRef>
      <SourceRef>./dev-contacts</SourceRef>
      <Data>200</Data>
    </Status>
  </SyncBody>
</SyncML>

```