DS 2.0 Requirements

Candidate Version 2.0 – 18 Apr 2007

Open Mobile Alliance

OMA-RD-DS-V2_0-20070418-C

Contents

Figures

Tables

1. Scope (Informative)

This document contains use-cases and high level requirements for improved data synchronization enabler which are needed to supply the core data synchronization service.

This document contains information applicable to Network Operators, terminal and network manufacturers, enterprises, independent software vendors, and service providers.

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

[XMLNS] “Namespaces in XML”, World Wide Web Consortium, January 14, 1999, http://www.w3.org/TR/REC-xml-names/

[XMLSCHM0] “XML Schema Part 0: Primer”, World Wide Web Consortium, October 28, 2004, http://www.w3.org/TR/xmlschema-0/

2.2 Informative References

[DSPRO] “DS Protocol”, Open Mobile Alliance™, OMA-TS-DS_Protocol-V1_2, URL:http://www.openmobilealliance.org/

[DEVINF] “DS Device Information”, Open Mobile Alliance™, OMA-TS-DS_DevInf-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

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

Data Store A logical storage of data elements. For example, client data store is used for store client-side data, such as vCard, vCalendar, etc.

Data Sync Client An entity 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 Sync Server An entity 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.

Device Equipment which is normally used by users for communications and related activities.

Implementer Manufacturer of the device, or a software company, producing data sync client and/or server.

Logical Session The logical session is a relationship between the client and server which continues while data is exchanged through multiple physical connections or sessions.

Message Atomic unit that contains the SyncML Commands, as well as the related data and meta-information.

Network Operator An entity providing network connectivity for a Device.

Package A conceptual set of commands that could be spread over multiple messages.

Server Alerted Sync The synchronization server sends a notification to the client, requesting that it initiate a synchronization with the server.

Service Provider An entity that combines content from various sources into a service or an application to be consumed on a mobile device by an end user.

User An entity which uses services. Example: a person using a data synchronization service.

3.3 Abbreviations

DS	Data Synchronization
DTD	Document Type Definition
EMS	Enhanced Messaging Service
GUID	Global Unique Identifier
IOP	Interoperability
LUID	Local Unique Identifier
MBCS	Multi Byte Character Set
MMS	Multimedia Messaging Service

OMA	Open Mobile Alliance
PIM	Personal Information Manager
SAS	Server Alerted Sync
SIP	Session Initiation Protocol
SMS	Short Message Service
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WBXML	WAP Binary XML
XML	Extensible Mark-up Language

4. Introduction (Informative)

The objective of this document is to collect the requirements for the next enabler release of OMA Data Synchronization from the whole industry perspective. The requirements for existing enabler releases of OMA DS are not covered in this document.

This document defines the requirements to enhance data synchronization in the following areas:

- Reducing Traffic (Compression, Reducing Transfer of information, Combining commands and packages, etc.)

  For solutions which attempt to implement quasi-real time (always up-to-date) views of data, the need to reduce the overhead of a sync session becomes key. As such reducing the number of round trips, reducing processing requirements, and reducing the size of the messages within each of these trips is the main goal of this work area.

- Improving Security (Binding level authentication and encryption, Protocol level encryption, etc.)

  As solutions that are DS based become more prevalent the need to satisfy the security concerns of all involved (users, IT departments, operators, etc…) also increases in prevalence. DS needs to mandate that implementations recognize this importance while providing the maximum number of options.

- Real-time Sync (Always on capabilities etc.)
  - Investigation into a new binding that provides always on abilities.
  - Possible merging with other notification techniques such as OMA Email Notification.
  - Ways to provide a continuous transparent sync experience.
- Adjustments to OMA DS based Email sync

  One of DS 1.2's key enhancements was the introduction of Email Sync capabilities. Since this was first envisioned within the SyncML Initiative several of the use cases have evolved however, which require enhancements in this area. After the initial DS WG Email Sync activities, the Mobile Email Requirements Subgroup was formed. In parallel with the continuing DS WG Email Sync activities, the Mobile Email Requirements Subgroup has been developing a Requirements Document. Many of these use cases are already addressed in the 1.2 OMA DS specifications and this work item will address others.

This work area should therefore consider…

  - Techniques for retrieving previously filtered objects.
  - Methods for intelligently forwarding and reply messages with attachments.
  - Small Enhancements to the Email data object as deemed required.

- Specification readability and interoperability improvements

- Syntax enhancements

5. Use Cases (Informative)

5.1 Consumer Short Message Synchronization

5.1.1 Short Description

As fashion youth, Tom will send and receive a large number of short messages. Short messages are one of his most important communication methods and some of them are much cherished by him. When his device memory is full, he would choose to save these short message to Data Sync Server owned by the Network Operator.

Tom can also reverse the sync operation and restore the messages from the Data Sync Server whenever needed.

5.1.2 Actors

- Device: a device which supports UI for short message synchronization.
- Data Sync Server: server which will may UI for subscriber to trigger backup reversely.
- User

5.1.3 Actor Specific Issues

N/A

5.1.4 Actor Specific Benefits

- User: Tom, user will enjoy the ability to preserve cherished short messages.
- Network Operator: The Network Operator can increase the revenue from providing a fashion service for the customer

5.1.5 Pre-conditions

GPRS session can be set up correctly.

The User has proper ID/password to access the Data Sync Server.

The Data Sync Client on the device has access to the short message store.

5.1.6 Post-conditions

Tom has successfully backed up his short messages on the Data Sync Server.

5.1.7 Normal Flow

1. Tom sets up GRPS connection.
2. He opens the UI on the device to choose short message sync on the received message folder.
3. The sync session is established between the Data Sync Client on the device and the Data Sync Server.
4. Short messages are transferred to the Data Sync Server in the background.
5. The sync operation is finished successfully.

5.1.8 Alternative Flow

1. Tom can choose short message synchronization from sending folder, draft folder, etc.

2. Tom can restore the short messages stored on the Data Sync Server to his device.

3. Tom can also receive a request from the Data Sync Server to transfer messages to the device from Data Sync Server. The request is initiated via the Internet.

5.1.9 Operational and Quality of Experience Requirements

N/A

5.2 Data store partial Synchronization

5.2.1 Short Description

As a company manager Tom stores many contacts of his colleagues, customers and friends. In order to manage them effectively, he creates the following folders to organize them: VIP, friends and colleagues. Then he divides the folder ‘colleagues’ into three subfolders: Software dept., Support dept., Market dept.

The contacts within the ‘colleagues’ change often. Sometimes Tom only wants to synchronize the ‘colleagues’, as well as its subfolders and items within them. For better understanding, a figure is provided:

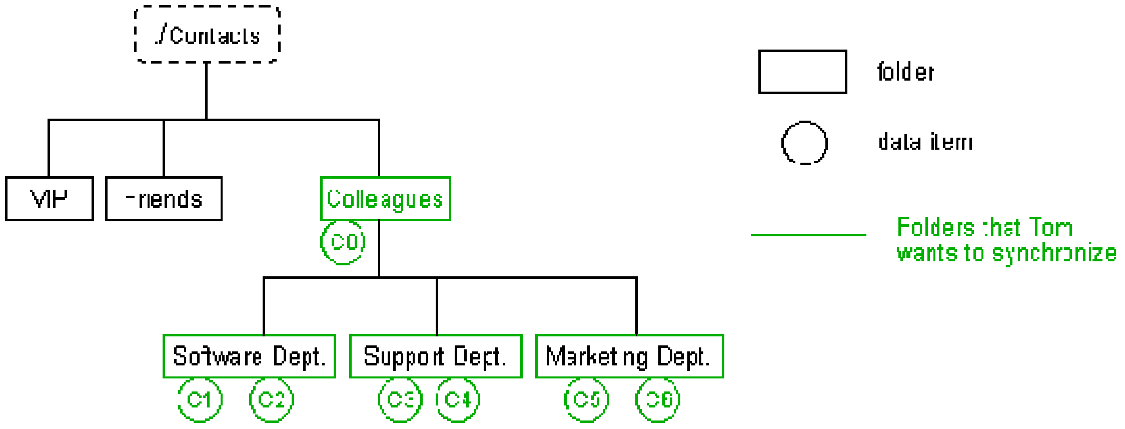

Figure 1: Hierarchical content example

5.2.2 Actors

- User: Tom

- Data Sync Client

- Data Sync Server

5.2.2.1 Actor Specific Issues

- Tom only needs to synchronize a part of his contacts’ hierarchy

5.2.2.2 Actor Specific Benefits

- Partial synchronization accords with user’s habit and is more flexible.

- It can reduce the traffic.

5.2.3 Pre-conditions

- Data Sync Client supports partial synchronization and hierarchical synchronization.

- Data Sync Server supports partial synchronization and hierarchical synchronization.

5.2.4 Post-conditions

- User selected folders are synchronized.

- Subfolders inside user selected folders and the items within them are synchronized based on user’s choice.

5.2.5 Normal Flow

1. Tom selects the folder ‘Colleague’ in his device and starts the synchronization.
2. The application in his device pops up a UI and asks Tom whether he wants to synchronize the subfolders and items within them.
3. Tom chooses ‘Yes’.
4. - Data Sync Client initiates the sync session and negotiates with the Data Sync Server that it only wants to synchronize folder ‘colleague’ recursively.
5. - Data Sync Server accepts the sync request and then the synchronization starts.
6. The synchronization session ends successfully

5.2.6 Alternative Flow1

Steps 1 - 2 are the same as in Normal Flow.

3. Tom chooses ‘No’ which indicates that he only wants to synchronize the ‘Colleagues’ folder and items within it.
4. Data Sync Client initiates the sync session and negotiates with the Data Sync Server that it only wants to synchronize folder ‘colleagues’ non-recursively.
5. Data Sync Server accepts the sync request and then the synchronization starts.
6. The synchronization session ends successfully.

5.2.7 Alternative Flow2

1. Tom selects the folder ‘VIP’ and ‘friends’ in his device at the same time and start the synchronization.
2. Data Sync Client initiates the sync session and negotiates with the Data Sync Server that it wants to synchronize two folders, but not the whole database.
3. Data Sync Server accepts the sync request and then the synchronization starts.
4. The synchronization session ends successfully

5.2.8 Operational and Quality of Experience Requirements

Data Sync Client MAY ask the user whether he/she wants to save the preference and automatically synchronize a folder recursively.

5.3 Putting and Getting partial device information

OMA DS device information (DevInfo) describes many characteristics of the device including version, microcode level, datastores supported, and many attributes of the data in each datastore, including whether it is possible to filter on the data item, or truncate it, and the maximum number of occurrences.

Over the course of several releases of DS the amount of data included in DevInfo has grown, yet it is still possible for some devices to support only small objects (ie. objects only as large as one buffer size) by indicating in DevInfo that they do not Support Large Objects. Large Object Support allows the sender to send multiple chunks of a large object, each chunk only the size of the recipient’s previously indicated maximum message size. It is a requirement that the DevInfo must be contained in a single message within the maximum message size to accommodate devices which are unable to support Large Objects

In the current spec there is no clear description of how to update and query partial device information. Without this method, syncing more datastores would require larger maximum message size on the device, and larger maximum message size could hit the upper bound of network capability and device sending buffer capability. Even when only static information of a single datastore has been changed, in the current version of the specification, the application would be required to send all the device information.

5.3.1 Short Description

Separating device information into messages by use of hierarchical update and query of device information allows reducing data traffic.

5.3.2 Actors

- Data Sync Client
- Data Sync Server
- User

5.3.3 Actor Specific Issues

- Both Data Sync Server and Data Sync Client support putting and getting partial device information of them by each other

5.3.4 Actor Specific Benefits

- For the User, shorter package size implies reduced connection time thus reduced connection cost and billing.
- For the User, reduced synchronization duration eliminates possible user feeling that synchronization is a slow process.
- For the User, enabling multiple datastores by putting devinfo in separate message expands more usability of services

5.3.5 Pre-conditions

- Data Sync Client syncs so many datastores at a time that the datastores information cannot be packed in a single message. Data Sync Client separates the information into messages. Data Sync Client also queries all the Data Sync Server side datastore information separately.

5.3.6 Post-conditions

- Sync is finished successfully
- Data Sync Client and Data Sync Server are in sync

5.3.7 Normal Flow

1. Data Sync Client initiates sync by putting a part of multiple datastore information and getting part of Data Sync Server side datastore information
2. Data Sync Server accepts the sync request and sends the requested Data Sync Server side datastore information.
3. Data Sync Client puts the rest of the datastore information and get rest of Data Sync Server side datastore information with final tag
4. Data Sync Server accepts the sync request and sends the requested Data Sync Server side datastore information with final tag.

5. Synchronization continues normally.

5.3.8 Alternative Flow

N/A

5.3.9 Operational and Quality of Experience Requirements

N/A

5.4 Using compression algorithms

OMA DS packages are in plain text (XML) or in WBXML (carried data are in clear). Using compression would reduce the size of the exchanged packages thus reducing traffic load.

5.4.1 Short Description

Thomas discovers that its on-line address book service offers now the possibility to backup all his data (emails, files, contacts, etc.) on his phone, which represent a significant amount of data to transfer. Since compression techniques are used for message transmission, synchronization is fast. As a result Thomas is satisfied and makes intensive use of the backup service.

5.4.2 Actors

- Data Sync Client
- Data Sync Server
- User : Thomas

5.4.2.1 Actor Specific Issues

- For the Data Sync Client, the compression algorithm implementation must be efficient enough in order that transmission gain compensates compression time cost.

5.4.2.2 Actor Specific Benefits

- For User, a fast synchronization implies reduced connection time thus reduced connection cost and billing.
- Fast synchronization can accelerate OMA DS protocol acceptance and usage.
- For User, reduced synchronization duration eliminates the feeling that synchronization is a slow process.

5.4.3 Pre-conditions

- Data Sync Client and Data Sync Server have declared which compression scheme they support.
- Both Data Sync Server and Data Sync Client must support the same compression/decompression scheme.

5.4.4 Post-conditions

- Sync is finished successfully
- Data Sync Client and Data Sync Server are in sync

5.4.5 Normal Flow

1. Data Sync Client initiates a sync requesting to use a specific compression technique (which has to be common to the Data Sync Server and Data Sync Client) for exchanged packages.

2. Data Sync Client and Data Sync Server continue the sync session exchanging compressed packages (using the chosen compression technique).

5.4.6 Alternative Flow

N/A

5.4.7 Operational and Quality of Experience Requirements

N/A

5.5 Simplified syntax

5.5.1 Short Description

Using a simplified syntax allows to reduce OMA DS package parser/generator complexity.

5.5.2 Actors

- Data Sync Client implementer
- Data Sync Server implementer
- User

5.5.2.1 Actor Specific Issues

N/A

5.5.2.2 Actor Specific Benefits

- For the Data Sync Client/Server implementer, a simplified syntax is less ambiguous and leads to simpler parsers/generators implementation.
- For the Data Sync Client/Server implementer, a less ambiguous syntax implies a better interoperability between different implementations.
- For the User: data synchronization will be more efficient

5.5.3 Pre-conditions

- Both Data Sync Server and Data Sync Client support simplified syntax – described by either DTD or XML Schemas.

5.5.4 Post-conditions

- Sync is finished successfully
- Data Sync Client and Data Sync Server are in sync

5.5.5 Normal Flow

1. Data Sync Client initiates sync with a package respecting the modified syntax.
2. Data Sync Server accepts the sync request.
3. Synchronization continues with packages respecting the simplified syntax.

5.5.6 Alternative Flow

N/A

5.5.7 Operational and Quality of Experience Requirements

N/A

5.6 Secure Data Synchronization

5.6.1 Short Description

The president of a company wants to keep her customer contact information on her device. In order to load the contact list, she chooses to synchronize the contact list that is on a Data Sync Server with her device.

After she sets up the proper connectivity and authentication information, she initiates the data synchronization session with the Data Sync Server. At the setup phase of the session, the device and the Data Sync Server ask for mutual authentication. If the device does not support transport layer encryption, then the Data Sync Client and Data Sync Server would encrypt the session data.

5.6.2 Actors

- Device
- Data Sync Server
- User

5.6.2.1 Actor Specific Issues

- Device should support application layer encryption if it does not support transport layer encryption.
- Server should support both transport layer and application layer encryption.

5.6.2.2 Actor Specific Benefits

- User would not worry about losing her information to unauthorized persons.

5.6.3 Pre-conditions

The Device supports transport layer encryption or application layer encryption.

The Data Sync Server supports both transport layer encryption and application layer encryption.

The User has right to access the Data Sync Server.

5.6.4 Post-conditions

The User has securely synchronized their contact list and calendar information, etc.

5.6.5 Normal Flow

- The Device initiates the session request to the Data Sync Server, which includes authentication information/credentials and the encrypted session request.
- The Data Sync Server and the Device successfully authenticate each other.
- The Data Sync Server and the Device agree upon the encryption to be used.
- The Data Sync Server and the Device encrypt the session data during a normal synchronization.
- After synchronization is done, the session ends normally, with the contact list securely synchronized.

5.6.6 Other issues to be considered

- Cryptographic functions to be supported
- Certificate support
- Recommendations on the chosen key lengths

5.6.7 Alternative Flow1 (Transport layer Security)

If the Device and the Data Sync Server support transport layer security (i.e. HTTPS), the Device and Data Sync Server will establish a mutually authenticated HTTPS connection prior to the start of the Data Sync session. Encryption is performed in transport layer and application layer encryption could be omitted.

5.6.8 Alternative Flow2

Besides specified against a Data Sync Server, the authentication and encryption challenges can be specified against a database. Furthermore, In the case of authentication challenges, they can be specified against an individual command on a database.

The main challenge regarding the real end to end security is the connection between data storage and Data Sync Server. In real life implementations data server (for example email server) and Data Sync Server are in different domains. The connection between the data storage and Data Sync Server is not specified by OMA DS. Hence OMA DS group can only *give recommendation* on the sufficient security solution between these two entities.

5.6.9 Alternative Flow3 (Integrity Protection)

The Device or Data Sync Server can request integrity protection in addition to encryption. Both the Device and the Data Sync Server MUST accept this request and provide a mutually acceptable mechanism for proof data is unchanged (e.g. a hash algorithm).

5.6.10 Operational and Quality of Experience Requirements

N/A

5.7 Sync Interruption & Continuation

OMA DS Protocol makes use of synchronization anchors to make sanity checks between Data Sync Client and Data Sync Server, in other words, to check whether Data Sync Client and Data Sync Server are on the same page in terms of synchronization. However, the usage of anchors is specified ambiguously, leaving many aspects of Data Sync Client and Data Sync Server behaviour open to question.

OMA DS 1.2 also contains a new feature, called Suspend/Resume for resuming the interrupted syncs without restarting them or initiating a slow sync. This feature allows to minimize the amount of synchronizations, time spent synchronizing and data sent over network in order to achieve synced state between Data Sync Client and Data Sync Server. However, the feature is also specified ambiguously, not clearly defining behaviour expected by sync participants in case of sync interruption.

Also, the usage of anchors mentioned above is inadequate for Suspend/Resume functionality, i.e. current anchors usage scheme does not allow to implement Suspend/Resume feature in a way that would fulfill Suspend/Resume’s goal for improving User experience.

Lack of IOP and inability to properly implement the resumption of sync using the current spec wording leads to user’s data corruption and /or loss, or high number of slow syncs (where client and server exchange the full content of their datasets) resulting in increasing charges and time to sync and in some cases, creating duplicate data items or unnecessarily deleting data items.

The nature of OMA DS 1.2 does not imply major changes to e.g. anchors mechanisms, also, the 1.2 version is now in a candidate state and needs to be released to gather more feedback from implementers, allowing OMA DS to solve the issues mentioned above. Thus, we're presenting requirements for OMA DS 2.0, that do not exactly introduce any new functionality, but cover aspects that were not covered by 1.2 when introducing certain features, e.g. IOP, ease of implementation etc.

The use cases presented as a base for requirements are similar to the use cases 1.2 was based on, however, since 1.2 was not able to cover these use cases properly, they are presented for OMA DS 2.0 as a base for major specification improvements.

5.7.1 Short Description

Alex usually performs syncs of his calendar on his way to the office to get an overview of the day ahead. Sometimes during this process he gets into places where there's no network. In spite of the interruptions Alex is able to sync his data successfully, quickly and keeping the charges low.

5.7.2 Actors

- User: Alex, initiator of the sync
- Data Sync Client: Data Sync Client implementation initiating the sync
- Data Sync Server: Data Sync Server implementation processing the sync request

5.7.2.1 Actor Specific Issues

- Data Sync Client is a 'thin client', i.e. implemented as simply as possible. Data Sync Client doesn't do any complex processing or conflict resolution.
- Data Sync Server is able to do complex processing and conflict resolution.

5.7.2.2 Actor Specific Benefits

- User is able to decrease the time spent synchronizing the data.
- User is able to decrease charges and thus save money.
- User does not have to worry about network coverage when performing a sync.

5.7.3 Pre-conditions

- Previous sync finished successfully

5.7.4 Post-conditions

- Sync is finished successfully
- No data corruption or loss occurred due to sync interruption
- Data Sync Client and Data Sync Server are in sync
- Next sync is a fast sync

5.7.5 Normal Flow

1. User makes changes on his mobile device and the portal
2. User initiates the sync using the user interface.
3. When the sync starts, User walks into an elevator. He sees the screen say 'Authorizing'.
4. Network fails and sync is interrupted.
5. User gets off the elevator, network is available again. User initiates the continuation of sync using the user interface.
6. Sync is performed in minimal time and finishes successfully.

5.7.6 Alternative Flow 1

Steps 1-2 are identical to Normal Flow.

3. When the sync starts, User walks into an elevator. He sees the screen say ‘Sending data’.

Steps 4-5 are identical to Normal Flow

Sync is performed in minimal time and finishes successfully.

5.7.7 Alternative Flow 2

Steps 1-2 are identical to Normal Flow.

3. When the sync starts, User walks into an elevator. He sees the screen say ‘Receiving data’.

Steps 4-5 are identical to Normal Flow

Sync is performed in minimal time and finishes successfully.

5.7.8 Operational and Quality of Experience Requirements

The overall time of synchronization should be minimal. It must not significantly exceed the time of analogous synchronization with no interruptions.

The overall cost of synchronization should be minimal. It must not significantly exceed the cost of analogous synchronization with no interruptions.

5.8 Always On Synchronization

Always-On is a new feature being proposed for the OMA DS specification. It will allow the data stored on Data Sync Clients and Data Sync Servers to always be synchronized. Whenever changes are detected on one side, without user intervention, the system will initiate a synchronization session. This feature allows the User’s data to always be synchronized, without the User noticing how and when all of the operations happened greatly enhancing the user experience.

5.8.1 Short Description

The User does not want to manually and explicitly (via device UI) initiate every synchronization with the Data Sync Server from his device. He just wants the device to automatically contain up-to-date information (eg. PIM, Email) whenever he views the data on his device, and he wants any updates that he makes on his device to be automatically propagated to the Data Sync Server.

5.8.2 Actors

- The User
- The Device
- The Data Sync server

5.8.2.1 Actor Specific Issues

N/A

5.8.2.2 Actor Specific Benefits

The user has his information all the time up-to-date on both client and server.

5.8.3 Pre-conditions

The device and server are configured to synchronize between each other.

5.8.4 Post-conditions

N/A

5.8.5 Normal Flow

1. Transport connection established between the client and the server.
2. Sync initialization takes place, the client and the server exchange full device information with each other; both devices indicate the synchronization types for all content to be synchronized. The logical synchronization session is now established between the client and the server, and both the client and the server are responsible of maintaining the session until the session is finalized.
3. In synchronization phase all data from the client is first sent to the server, after which the server compares the items received from the device to the items the server has, and sends the necessary items to the device. Client sends necessary status and map commands to the server.
4. Transport is disconnected.
5. Both the server and the client are in sync, and the logical session is maintained.

5.8.6 Alternative Flow 1 - End User Turns Always On functionality ON/OFF

The *user* activates the Always On feature. In this situation, the client should initiate an incremental sync to make sure that the data is correctly synchronized. At the end of the synchronization, the logical session remains open.

The user deactivates the Always On feature. The only action is for the client to close the logical session.

5.8.7 Alternative Flow 2 - End User performs a manual sync while Always On is activated

Although the User is normally satisfied with the automatic updates his device receives and propagates when using his default filters, he is at the airport, about to board a plane and wants to receive all of the text, including attachments, of the emails he has received from a customer in the last 2 weeks so he can review them during the plane ride to visit the customer. He is able to change his filters but also wants to be able to force a manual sync to retrieve the full set of emails from the customer.

5.8.8 Operational and Quality of Experience Requirements

N/A

5.9 Avoiding unnecessary notifications

5.9.1 Short Description

Lemon Telecom, a worldwide telecom company, wants to provide its customers with services based on OMA-DS protocol. Some of those services make use of the Data Sync Server Alerted Notification feature. Lemon Notification Server wants to receive acknowledgment for any sent notification in order to avoid resending unnecessary and unsolicited notifications.

5.9.2 Actors

- Service Provider
- Data Sync Client

- Data Sync Server
- Notification Server
- User: Ian

5.9.2.1 Actor Specific Issues

N/A

5.9.2.2 Actor Specific Benefits

- The User is not annoyed with unsolicited notifications
- The Notification Server has not to resend unnecessary notifications
- The Service Provider can improve its quality of service since acknowledgement allows finer analyze.

5.9.3 Pre-conditions

- Notification Server can send SAS packages.
- Notification Server knows how to reach the device (its phone number)
- Data Sync Client is configured to accept SAS.
- Data Sync Server is known by the Data Sync Client

5.9.4 Post-conditions

- The Notification Server is informed that the Data Sync Client has received the notification.

5.9.5 Normal Flow

1. During first sync between Data Sync Client and Data Sync Server, the Data Sync Client declares which transports it supports for SAS.
2. Notification Server sends a SAS package to the Data Sync Client using one of the above related transports
3. The Data Sync Client sends a SAS acknowledgment to the Notification Server
4. Following SAS specifications, synchronization is started either manually either automatically.

5.9.6 Alternative Flow

N/A

5.9.7 Operational and Quality of Experience Requirements

N/A

6. Requirements (Normative)

6.1 High-Level Functional Requirements

Label	Description	Enabler Release
HLF-1	The OMA DS Enabler SHALL contain mechanisms to support the functionality required by additional application use cases. Such as Mobile Email, Short Message Synchronization, DRM related content, etc.	DS 2.0
HLF-2	The OMA DS Enabler SHALL support improved mechanisms to allow Data Sync Clients and Data Sync Servers to identify the information subsets of interest.	DS 2.0
HLF-3	The OMA DS Enabler SHALL provide improved mechanisms for the declaration of Data Sync Clients and Data Sync Servers capabilities	DS 2.0
HLF-4	The OMA DS Enabler SHALL provide improved mechanisms for the negotiation of Data Sync Clients and Data Sync Servers preferences.	DS 2.0
HLF-5	The OMA DS Enabler SHALL NOT have dependencies on specific data object types.	DS 2.0
HLF-6	The OMA DS Enabler SHALL be based on precise data and language definition methodologies.	DS 2.0
HLF-7	The OMA DS Enabler SHALL provide additional and improved protocol level security mechanisms.	DS 2.0
HLF-8	The OMA DS Enabler SHALL define conformance requirement profiles for various classes of Data Sync Clients and Data Sync Servers.	DS 2.0
HLF-9	The OMA DS Enabler SHALL normalize specification documentation to reduce maintenance and ambiguities.	DS 2.0
HLF-10	The OMA DS Enabler SHALL provide additional and improved means to optimize bandwidth utilization and minimize latency.	DS 2.0
HLF-11	The OMA DS Enabler SHALL provide additional and improved means to reduce data loss and reduce duplication.	DS 2.0
HLF-12	The OMA DS Enabler SHOULD reduce the need for Data Sync Clients and Data Sync Servers to exchange the full content of their Data Sets to achieve synchronization.	DS.2.0
HLF-13	The OMA DS Enabler SHALL NOT define, for any protocol element, different syntax or functionality of that element between Data Sync Clients and Data Sync Servers.	DS 2.0
HLF-14	The OMA DS Enabler SHALL allow for multiple data exchanges without requiring a new sync session for each data exchange.	DS 2.0
HLF-15	The OMA DS Enabler SHOULD define transport bindings for SAS packages.	DS 2.0
HLF-16	The OMA DS Enabler SHOULD provide mechanisms for the acknowledgement of SAS packages	DS 2.0
HLF-17	The OMA DS Enabler SHOULD provide improved SAS package definition to enable richer content.	DS 2.0

Table 1: High-Level Functional Requirements

6.1.1 Security

Label	Description	Enabler Release
Security-01	The OMA DS Enabler SHALL declare a list of common protocol layer encryption/decryption techniques that all implementations SHALL support.	DS 2.0
Security-02	The OMA DS Enabler SHOULD declare a list of additional protocol layer encryption/decryption techniques that all implementations MAY support.	DS 2.0

Security-03	Data Sync Client and Data Sync Server SHALL be able to declare which encryption/decryption technique they support for package exchanges.	DS 2.0
Security-04	The OMA DS Enabler SHALL support transport layer encryption, such as TLS.	DS 2.0
Security-05	The OMA DS Enabler SHALL support a mechanism to do protocol layer integrity protection.	DS 2.0
Security-06	The OMA DS Enabler MAY support protocol layer certificate management to maintain the encryption keys.	DS 2.0
Security-07	The OMA DS Enabler SHALL support expiration of protocol layer authentication.	DS 2.0

Table 2: High-Level Functional Requirements – Security Items

6.1.1.1 Authentication

Label	Description	Enabler Release
AUTH-1	The OMA DS Enabler SHALL provide mechanism for the Data Sync Server to authenticate the Data Sync Client.	DS 2.0
AUTH-2	The OMA DS Enabler SHALL provide mechanism for the Data Sync Client to authenticate the Data Sync Server.	DS 2.0
AUTH-3	The OMA DS Enabler SHALL be able to provide replay protection to ensure confidence that a received message has not been recorded and played back.	DS 2.0

Table 3: High-Level Functional Requirements – Authentication Items

6.1.1.2 Authorization

N/A

6.1.1.3 Data Integrity

Label	Description	Enabler Release
DATA-1	The OMA DS Enabler SHALL provide mechanisms to ensure data integrity, protecting against accidental or intentional changes to the data, by ensuring that changes to the data are detectable.	DS 2.0

Table 4: High-Level Functional Requirements – Data Integrity Items

6.1.1.4 Confidentiality

Label	Description	Enabler Release
CONF-1	The OMA DS Enabler SHALL use/support data confidentiality that ensures that transmitted information is not made available or disclosed to unauthorised individuals, entities, or processes.	DS 2.0

Table 5: High-Level Functional Requirements – Confidentiality Items

6.1.2 Charging

N/A

6.1.3 Administration and Configuration

N/A

6.1.4 Usability

N/A

6.1.5 Interoperability

Label	Description	Enabler Release
IOP-1	Data Sync Clients and Data Sync Servers complying with OMA DS Enabler SHALL be interoperable and produce consistent sync results.	DS 2.0
IOP-2	The OMA DS Enabler SHALL be unambiguous and easy to implement.	DS 2.0
IOP-3	The OMA DS Enabler SHALL provide a mechanism for Data Sync Client and Data Sync Server to negotiate the current sync session parameters such as compression, encryption, data objects and authentication.	DS 2.0

Table 6: High-Level Functional Requirements – Interoperability Items

6.1.6 Privacy

N/A

6.2 Overall System Requirements

Label	Description	Enabler Release
SYSTEM-0	The Improved OMA DS Enabler SHOULD provide at least the equivalent functionality present in OMA DS 1.2 Enabler.	DS 2.0

Table 7: High-Level System Requirements

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 2.0 History

Document Identifier	Date	Sections	Description
OMA-DS-2005-0230R01-INP_Draft_DS_RD_Dec13_2005	13 Dec 2005	All sections	Incorporates verbal input from REQ after informal review
Draft versions: OMA-RD-DS-V2_0	09 Jan 2006	All sections	Applied new Template: OMA-Template-ReqDoc-20060207-I.doc
	18 Oct 2006	Sections 4, 5, 6, Appendix B, 5.3	Applied the following agreed CRs: OMA-DS-DS_2_0-2006-0060R01-CR_RD_Update OMA-DS-DS_2_0-2006-0001R01-CR_RD_REQInformalReview OMA-DS-DS_2_0-2006-0012-REQ_HLF_13_CR
	20 Feb 2007	Sections 3.2, 5.3, 5.6, 5.6, 6 and 6.1.	Applied editorial changes suggested in Formal R&A by Req and also applied editorial changes to address verbal suggestions made in an earlier Informal Review by Req These are recorded in the RDRR.
	05 Mar 2007	All	Final comments in the RDRR addressed
Candidate version OMA-RD-DS-V2_0	18 Apr 2007	n/a	Status changed to Candidate: OMA-TP-2007-0125R01-INP_RD_DS_V2_0_for_Candidate_Approval