OMA DS Concepts and Definitions

Approved Version 2.0 – 19 Jul 2011

Open Mobile Alliance

OMA-TS-DS_Concepts-V2_0-20110719-A

Contents

1. Scope

This document specifies the overall concepts, references, and definitions of OMA DS.

2. References

2.1 Normative References

[BTAN] "Bluetooth Assigned Numbers", Bluetooth SIG,
URL: http://www.bluetoothsig.org/assigned-numbers/

[BTGOEP] "Bluetooth V1.1 Profile Specifications" – PartK:10 Generic Object Exchange Profile, Bluetooth SIG,
URL:http://www.bluetooth.org/foundry/specification/document/Bluetooth_11_Profiles_Book/en/1/Bluetooth_11_Profiles_Book.pdf

[BTSDP] "Bluetooth V1.1 Core Specifications" – PartE: Service Discovery Protocol, Bluetooth SIG,
URL:http://www.bluetooth.org/foundry/specification/document/Bluetooth_V1.1_Core_Specifications/en/1/Bluetooth_V1.1_Core_Specifications.pdf

[BTSEP] "Bluetooth V1.1 Profile Specifications" – PartK:5 Serial Port Profile, Bluetooth SIG,
URL:http://www.bluetooth.org/foundry/specification/document/Bluetooth_11_Profiles_Book/en/1/Bluetooth_11_Profiles_Book.pdf

[DEIF] "Data elements and interchange formats – Information interchange – Representation of dates and times",
URL:http://www.iso.ch/iso/en/ISOOnline.frontpage

[DEVINF] "OMA DS Device Information ", Open Mobile Alliance™, OMA-TS-DS-DevInfo-V2_0,
URL:http://www.openmobilealliance.org

[DEVINF_XMLSCH] "OMA DS Device Information XML Schema", Open Mobile Alliance™, OMA-TS-DS-DevInfo_XML_Schema-V2_0,
URL:http://www.openmobilealliance.org

[DSHTTPBINDING] "SyncML HTTP Binding Specification", Open Mobile Alliance™,
OMA-TS-SyncML_HTTPBinding-V1_2_1,
URL:http://www.openmobilealliance.org/

[DSNOTIF] "Data Synchronization Notification", Open Mobile Alliance™, OMA-TS-DS_Notification-V2_0,
URL:http:www.openmobilealliance.org

[DSOBEXBINDING] "SyncML OBEX Binding Specification", Open Mobile Alliance™,
OMA-TS-SyncML_OBEXBinding-V1_2,
URL:http://www.openmobilealliance.org/

[DSPRO] "Data Synchronization Protocol", Open Mobile Alliance™, OMA-TS-DS_Protocol-V2_0,
URL:http:www.openmobilealliance.org

[DSPROVIS] "Data Synchronization Provisioning", Open Mobile Alliance™, OMA-TS-DS_Provisioning-V2_0,
URL:http:www.openmobilealliance.org

[DSSYNTAX] "Data Synchronization Syntax", Open Mobile Alliance™, OMA-TS-DS_Syntax-V2_0,
URL:http:www.openmobilealliance.org

[DSSYNTAX_XMLSCH] "Data Synchronization Syntax XML Schema", Open Mobile Alliance™, OMA-TS-DS_Syntax_XML_Schema-V2_0,
URL:http:www.openmobilealliance.org

[DSWSPBINDING] "SyncML WSP Binding Specification", Open Mobile Alliance™,
OMA-TS-SyncML_WSPBinding-V1_2,
URL:http://www.openmobilealliance.org/

[IMCVCAL] "vCalendar – The electronic calendaring and scheduling exchange format – Version 1.0",
URL:http://www.imc.org/pdi/vcal-10.doc

[IMCVCARD] "vCard – The electronic business card – Version 2.1",
URL:http://www.imc.org/pdi/vcard-21.doc

[IMEI] "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification" (3G TS 23.003 Version 3.4.0 Release 1999),
URL:http://webapp.etsi.org/action/PU/20000523/ts_123003v030400p.pdf

[IOPPROC] "OMA Interoperability Policy and Process", Open Mobile Alliance™, OMA-IOP-Process-V1_3, URL:http//www.openmobilealliance.org

[ISO8601] "Data elements and interchange formats – Information interchange – Representation of dates and times ISO 8601-2000", URL://www.iso.ch/iso/en/ISOOnline.openerpage

[OBEX] "IrDA Object Exchange Protocol (IrOBEX) Version 1.2", Infrared Data Association, URL: http://www.irda.org/standards/pubs/OBEX1p2_Plus.zip

[PUSH] "Push OTA Protocol Specification", WAP Forum, URL:http://www1.wapforum.org/tech/terms.asp?doc=WAP-235-PushOTA-20010425-a.pdf

[PUSHARCH] "Push Architectural Overview", WAP Forum, URL:http://www1.wapforum.org/tech/terms.asp?doc=WAP-250-PushArchOverview-200010703-a.pdf

[RFC1321] " The MD5 Message-Digest Algorithm ", R. Rivest, et al., April 1992, http://www.ietf.org/rfc/rfc1321.txt

[RFC1766] "Tags for the Identification of Languages", H. Alvestrand, March 1995, URL:http://www.ietf.org/rfc/rfc1766.txt

[RFC2045] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed & N. Borenstein, November 1996, URL:http://www.ietf.org/rfc/rfc2045.txt

[RFC2104] "HMAC: Keyed-Hashing for Message Authentication". H. Krawczyk, M. Bellare, R. Canetti February 1997. URL:http://www.ietf.org/rfc/rfc2104.txt

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

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF", D. Crocker, Ed., P. Overell, November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt

[RFC2279] "UTF-8, a transformation format of ISO 10646", F. Yergeau, January 1998, URL:http://www.ietf.org/rfc/rfc2279.txt

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, et al., August 1998, URL:http://www.ietf.org/rfc/rfc2396.txt

[RFC2425] "A MIME Content-Type for Directory Information" T. Howes, M. Smith, F. Dawson, September 1998, URL:http://www.rfc-editor.org/rfc/rfc2425.txt

[RFC2426] "vCard MIME Directory Profile", F. Dawson, T. Howes, September, 1998, URL:http://www.ietf.org/rfc/rfc2426.txt

[RFC2437] "RSA Cryptography Specifications Version 2.0". B. Kaliski, J. Staddon, October 1998. URL:http://www.ietf.org/rfc/rfc2437.txt

[RFC2445] "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", F. Dawson, D. Stenerson, November 1998, URL:http://www.ietf.org/rfc/rfc2445.txt

[RFC2616] "Hypertext Transfer Protocol – HTTP/1.1", R. Fielding, et al., June 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

[RFC2617] "HTTP Authentication: Basic and Digest Access Authentication", J. Franks, et al., June 1999, URL:http://www.ietf.org/rfc/rfc2617.txt

[RFC822] "Standard for the format of ARPA Internet text messages", David H. Crocker, August 1982, URL:http://www.ietf.org/rfc/rfc822.txt

[TS24008] "Technical Specification Group Core Network and Terminals; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3" 3GPP TS 24.008, URL:http://www.3gpp.org/

[VOBJMIP] "vObject Minimum Interoperability Profile", Version 1.0, Open Mobile Alliance™, OMA-TS-vObjectOMAProfile-V1_0-20071002-A,

URL:http:www.openmobilealliance.org

[WBXML] "WAP Binary XML Content Format Specification", WAP Forum™, WAP-240-WBXML, URL:http://www.openmobilealliance.org

[WDP] "Wireless Datagram Protocol Specification", WAP Forum, URL:http://www1.wapforum.org/tech/terms.asp?doc=WAP-259-WDP-20010614-a.pdf

[WSP] "Wireless Session Protocol specification", URL:http://www.wapforum.org/WAP-230-WSP-20010705-a.pdf

[WTP] "Wireless Transaction Protocol Specification", WAP Forum, URL:http://www1.wapforum.org/tech/terms.asp?doc=WAP-224-WTP-20010710-a.pdf

[XML] "Extensible Markup Language (XML) 1.0", World Wide Web Consortium Recommendation, URL:http://www.w3.org/TR/REC-xml

2.2 Informative References

[DSHISTORY] "OMA DS Standards Change History", Open Mobile Alliance™, OMA-WP-SyncML_ChangeHistory, URL:http:www.openmobilealliance.org

[DSPRIMER] "A Primer to SyncML/OMA DS", Open Mobile Alliance™, OMA-WP-SyncML_Primer, URL:http:www.openmobilealliance.org

[OMADICT] "Dictionary for OMA Specifications", Open Mobile Alliance™, OMA-ORG-Dictionary URL:http://www.openmobilealliance.org/

[RFC2045] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies". N. Borenstein, N. Freed, November 1996. URL:http://www.ietf.org/rfc/rfc2045.txt

[WAPARCH] "WAP Architecture". Open Mobile Alliance™. WAP-210-WAPArch. 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.

Any reference to components of the Data Synchronization XML Schema or XML snippets is specified in this `typeface`.

3.2 Definitions

Application	A SyncML application that supports the OMA DS protocol. The application can either be the originator or recipient of the SyncML protocol commands. The application can act as an OMA DS client or an OMA DS server.
Capabilities Exchange	The OMA DS capability that allows a client and server to exchange what device, user and application features they each support.
Client Modification	A modification of an item, which occurs in a client datastore before the modification is synchronized to the server database.
Command	A SyncML Command is a protocol primitive. Each SyncML Command specifies to a recipient an individual operation that is to be performed. Examples of SyncML Commands include Add, Delete, and Replace.
Content Format	The format used to represent a specific type of content (e.g. vCard 2.1 is a format to represent contact information).
Data	A unit of information exchange, encoded for transmission over a network.
Data element equivalence	When two data elements are synchronized. The exact semantics is defined by a given data synchronization model.
Data Exchange	The act of sending, requesting or receiving a set of data elements.
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 (DS 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 (DS 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.
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).
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.
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 OMA DS

	specification forms the basis for specifying an open data synchronization protocol.
Device	See [OMADICT].
Device Information	A document or object store (i.e., a database) on the source device that records information about the capabilities of the source device.
Enabler Release	See [OMA_DICT].
GUID (Global Unique Identifier)	An identifier assigned by the server to an object in a server data store. GUID values are expected to be unique, and should not be reused. Note that in practice, identifiers do not have to be unique forever, they MUST only be unique as long as they exist in some mapping table. Note that while GUIDs for particular objects are frequently expected to be the same from a particular server to all clients that it interacts with, this is not an explicit requirement. Note that if a truly globally unique or Universally unique identifier is required, it must be embedded within the particular data object – OMA DS GUIDs are opaque identifiers without a specified format, and must not be longer than the MaxGUIDSize specified by the client.
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.
LUID (Locally Unique Identifier)	An identifier assigned by the client to an object in a client data store. LUID values are expected to be locally unique, i.e., to a particular OMA DS client data store, but MAY be present on other OMA DS client data store. LUIDs are expected to be unique per device and per application, and should not be reused. Note that while LUIDs for particular objects are frequently expected to be the same from a particular client to all servers that it interacts with, this is not an explicit requirement.
Message	A grouping of OMA-DS Protocol elements in a valid XML document sent in one direction. E.g. Sync commands sent from the data sync client to the data sync server, or the responses back from the data sync server. Generally messages are grouped into packages in sessions.
Minimum Functionality Description	See [OMA_DICT].
Network Operator	An entity providing network connectivity for a Device.
Notification	A Command sent outside of a session. A logical set of operations sent in an asynchronous fashion, e.g. sent in an SMS message.
Notification Initiated Session	Device Management terminology for Server Alerted Notification.
Operation	A SyncML Operation refers to the conceptual transaction achieved by the SyncML Commands specified by a SyncML Package. For example in the case of data synchronization, "synchronize my personal address book with a public address book".Definition Needed that relates to individual operations, such as a single Add
Originator	The network device that creates a SyncML request.
Package	A SyncML Package is the complete set of commands and related data elements that are transferred between an originator and a recipient. The SyncML package can consist of one or more SyncML Messages. [ERP 1.2] OR A conceptual set of commands that could be spread over multiple messages. [AD&RD 2.0]. An identified logical set of related operations. E.g., the data sync client sending identifying information to the data sync server, or the data sync client sending all client modifications to the data sync server. A given package may take multiple messages, or certain packages may be combined into a single message.

Parser	Refers to an XML parser. An XML parser is not absolutely necessary to support SyncML. However, an OMA DS implementation that integrates an XML parser might be easier to enhance. This document assumes that the reader has some familiarity with XML syntax and terminology.
Recipient	The network device that receives a SyncML request, processes the request and sends any resultant SyncML response.
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.
Request	A message or a command sent from a device to another.
Server Alerted Notification	The general term for Server Alerter Synchronization
Server Alerted Sync	Data Synchronization usage of Server Alerted Notification.
Server modification	A modification of an item, which occurs in the server database before the modification is synchronized to the client database.
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.
Sync Type	Refers to behavior and direction associated with the synchronization session.
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.
Synchronization data	Refers to the data elements within a SyncML Command. In a general reference, can also refer to the sum of the data elements within a SyncML Message or SyncML Package.
SyncML request message	An initial SyncML Message that is sent by an originator to a recipient network device.
SyncML response message	A reply SyncML Message that is sent by a recipient of a SyncML Request back to the originator of the SyncML Request.
Temporary GUID	A temporary number assigned by the server to an object in a database (See also GUID.). Temporary GUID values are valid till the map operation for the items, with which the temporary GUIDs are associated, has been received from the client. After that the temporary GUID can be erased.
User	See [OMA_DICT].
Usage	Refers to the specific usage parts of the SyncML protocol, i.e. Data Synchronization or Device Management

3.3 Abbreviations

ABNF	Augmented Backus-Naur Form [RFC2234]
DTD	Document Type Definition
GUID	Global Unique Identifier
HTTP	HyperText Transfer Protocol [RFC 2616]
IMEI	International Mobile Equipment Identifier
LUID	Local Unique Identifier
MD5	Message Digest algorithm version 5 [RFC1321]
MIME	Multi-purpose Internet Mail Extensions
MSC	Message Sequence Chart
MSG	Message
OBEX	Object Exchange protocol |OBEX]

OMA	Open Mobile Alliance
PPG	Push Proxy Gateway [PUSHARCH]
URI	Uniform Resource Identifier [RFC2396]
URL	Uniform Resource Locator [RFC2396]
WAP	Wireless Application Protocol
WBXML	Wireless Binary XML Content Format
WSP	Wireless Session Protocol [WSP]
XML	Extensible Markup Language [XML]

4. Introduction

This document discusses the overall concepts involved in data synchronization, and the use of the OMA-DS protocol to achieve a successful synchronization. For a brief overview of OMA-DS, see [DSHISTORY]. For an introduction to OMA-DS, see [DSPRIMER].

4.1 Version 2.0

This is the first version of the Concepts specification. The reason for the specification version number 2.0 is to be consistent with the service release version number, that is, DS 2.0.

This specification describes the DS 2.0 concepts and the data synchronization usage.

5. OMA DS 2.0 Concepts

5.1 Data Sync based on Fingerprints

OMA-DS 2.0 has revised the standard Sync mechanism. This incorporates the DS 1.x functionality of Slow Sync, two-way sync, and so on. It revises the previous behavior of sync anchors, and suspend and resume behavior.

Sync now begins with the optional transferal of values known as fingerprints. This data is used to determine what data items, and potentially what portions of which data items need to be transferred during the sync session.

5.1.1 Fingerprints (FP)

Fingerprints are values associated with particular data item contents. They may be specified in a number of different methods. The method used may be negotiated, or specified by the data object format. Fingerprint size is typically 4 bytes.

Fingerprints are compared to detect if particular data items have changed, and may contain additional information, such as details of what has changed.

Fingerprints are transmitted as paired values – ID and fingerprint, or data item with ID and fingerprint. Some situations may transfer only the fingerprints for records that have changed.

5.1.2 Filtering support

When filtering is applied to records within data sets, the fingerprint of individual records will change as the set of fields within the filter change. It some situations, it may be computationally intensive to regenerate the fingerprint, so client generated fingerprint algorithms may choose to simplify this, as long as the fingerprint still changes when any data within the records, or when the active set of fields changes.

5.1.3 Algorithm Negotiation

It is recommended that each data object definition that does not wish to use client generated fingerprints define what fingerprint characteristics are allowed. For example, it may be reasonable for file objects to use mutually generated fingerprint, whereas data items that are stored with device specific restrictions such as truncation, or unsupported fields, may only make sense to use client generated algorithms.

The actual algorithm used may also be restricted by the device capabilities, as specified in the device info.

6. Data Synchronization Usage

DS Servers and DS Clients have a variety of possible usage scenarios. This section describes how to realize several of these possible use cases. Note that there is no requirement or expectation to present the use of the DS Protocol using any of the terms from the DS Protocol – frequently it is simpler to present the information about what is happening to the user in user-centric terms.

6.1 Bi-Directional Periodic Sync (Normal Sync)

One of the most common uses of OMA-DS is to keep the data of two information repositories, such as a mobile device and an Internet Portal synchronized. This may involve exchanging address book, calendar data, Email, or other types of data. When done on a periodic basis, disconnecting between each use, this would typically involve doing a Normal Sync, as discussed in [DSPRO] Section 7.2.1.

For this use case, the Sync Type Parameters ([DSPRO] Section 5.1) would be Direction:twoWay and Behavior:Preserve. Typically IDValidity and ChangeLogValidity would both be true, except for the first sync, and recovering from any error.

Things to consider during implementation include:

- In typical use, round trips may be saved by using a Sync without Separate Initialization, as described in [DSPRO] section 8.3. To be able to do this, the authentication method should be known, and available to use, such as when previous sessions have been successful with a particular authentication method, and the next nonce is known (if needed). Additionally, no special conditions such as losing change log information should apply. The use of Sync without Separate Initialization is recommended whenever possible.
- When regular polling is expected at moderately short intervals (e.g. minutes), and there is the ability to maintain a persistent session, Bi-Directional Continuous Sync (Section 6.2) should be used. Doing so will save on per session overhead, and reduce the latency for receiving changes.

6.2 Bi-Directional Continuous Sync

Bi-Directional Continuous Sync starts just as Bi-Direction Periodic Sync does, but maintains the connection after the initial data exchange through the use of a Session Maintenance command ([DSPRO] Section 6.2). This allows the savings of overall session overhead, such as authentication, as well as reducing the latency for receiving updates.

The Sync Type Parameters ([DSPRO] Section 5.1) would be Direction:twoWay and Behavior:Preserve. Typically IDValidity and ChangeLogValidity would both be true.

Things to consider during implementation include:

- Alert Poll ([DSPRO] Section 6.2.1) from the DS Client is best suited for situations where the majority of updates are expected to occur on the DS Client, or the timeliness of the propagation of DS Client changes is more significant than that of the DS Server changes. Alert Poll from the DS Client involves the DS Client waiting until additional data is available, and then sending another message – usually including information about what data has changed, and another Alert Poll. Implementations might also consider doing this for a short while after user input, based on a presumption that additional user input is likely.
- Alert Idle ([DSPRO] Section 6.2.2) from the DS Client is best suited for situations where the majority of updates are expected to occur on the DS Server, or from other DS Clients propagated through the DS Server, or where the timeliness of the DS Server changes is more significant than that of the DS Client changes. Alert Idle from the DS Client involves the DS Client sending the Alert Idle to the DS Server, with the response from the DS Server delayed until changes are available, or the specified timeout has occurred. This is best suited for devices that do not expect a lot of data to be changed (such as devices without a keyboard), or where no recent user input has occurred on the device.
- Note that the establishment of the timeout values for Session Maintenance depends upon the current environment, as well as user expectation. In general, there will be tradeoffs between the responsiveness of changes flowing in each direction versus battery use or connection overhead. In some environments it may be desirable to use logic to determine the most appropriate timeout values, such as by gradually increasing the timeout values until something in the environment causes the connection to be dropped, and then using a value below that in future sessions. Consideration need be given to any externally imposed limits on the timeout intervals.

6.3 Backup and Restore

6.3.1 Full Backup

To create a backup copy of a DS Client's data, a Sync may be performed with a Direction value of fromClient, and a Behavior value of Refresh. The DS Client need only send a New Sync Anchor, without a Last Sync Anchor. This new Sync Anchor will be a place marker for when the backup was performed.

Note that Servers may choose to support multiple sync anchors (for any operation, not just Full Backup). This allows for restoring the data to the values held at various points in time, such as for an Undo capability. Servers that support multiple anchors should indicate this in the StoredAnchors tree of their device information. Note that the MaxStoredAnchors element is a guideline, not an exact value, because the number of valid restore points may depend upon multiple factors, such as the amount of data, and the time the data is to be held.

6.3.2 Incremental Backup

To update an existing backup copy of a DS Client's data, a Sync may be performed with a Direction value of fromClient, and a Behavior value of Preserve. This may be done relative to previous Backups (Full or Incremental), or from previous twoWay syncs.

6.3.3 Continuous Incremental Backup

Continuous Incremental Backup begins with an initial data exchange that includes client data (Either Direction:twoWay or Direction:fromClient), and then maintains the connection after the initial data exchange through the use of a Session Maintenance command ([DSPRO] Section 6.2). This allows the savings of overall session overhead, such as authentication, as well as reducing the latency for receiving updates.

Since data need only be transferred from the DS Client to the DS Server for a Backup operation, Alert Poll ([DSPRO] Section 6.2.1) from the DS Client would be the preferred Session Maintenance command.

6.3.4 Full Restore

To restore the data on the DS Client, a Sync may be performed with a Direction value of fromServer, and a Behavior value of Refresh. The DS Client need only send a New Sync Anchor, without a Last Sync Anchor. If the DS Server supports multiple Sync Anchors, as indicated by its device information, then the specified New Sync Anchor may be a Sync Anchor known to the DS Server, in which case the data from that point in time would be restored. In all other cases, the data will be restored to the most currently known data available to the server. DS Clients that need to identify valid restore points (such as for an Undo Function), may read them from the ValidAnchor element(s) of the DS Server's Device Information.

6.3.5 Incremental Restore

To restore the data on the DS Client to a previous state, a Sync may be performed with a Direction value of fromServer, and a Behavior value of Preserve. The DS Client would then specify both a Last Sync Anchor, to indicate what data it currently has, and a Next Sync Anchor. This allows the DS Server to send only the changes required to update the data on the DS Client to the specified Next Sync Anchor. Note that failing to specify a known Last Sync Anchor would leave the DS Server without the information required to determine which changes to send to the DS Client, so a Recovery Sync would be required.

6.4 Subscription, Contribute, and Similar Cases

In certain situations, the user of a device might only be interested in the propagation of future changes. To address this, a Sync may be done with a Direction value of NoWay, which will effectively clear the change log, so that only future changes will be seen.

If the DS Client wishes to subscribe to changes from the DS Server, it may then do Syncs with a Direction value of fromServer, and a Behavior value of Preserve.

If the DS Client wishes only to Contribute its local changes to the DS Server, it may then do Syncs with a Direction value of fromClient and a behavior value of Preserve.

If the DS Client wishes to both subscribe to DS Server changes, and to Contribute its local changes to the DS Server, it may then do Syncs with a Direction value of twoWay and a behavior value of Preserve.

These syncs may be either periodic, or continuous, as desired.

Note that Email or File data (such as MP3s) is a typical use of these options – the user takes a device with them for travel, and is only interested in any new messages that come in. There is no need to delete any existing emails – they might be useful, but there is also no need to transfer existing emails from the DS Server down to the DS Client.

6.5 Client Controlled Conflict Resolution

In certain situations, the DS Client may need to perform some of the functions of the DS Server, such as conflict resolution. This might occur when one DS Server wishes to synchronize with another DS Server, such as when a new DS Server wishes to take some functions over from an existing DS Server. Since the protocol does not allow DS Server to DS Server communication, one of the devices must take on the role of DS Client.

If the new DS Client wishes to control the results of synchronization, one way to do so is to perform multiple Syncs. In the first Sync, which would be specified as a Direction of twoWay, the DS Client would refrain from actually sending any of its changes. The DS Server would send all of its changes as usual. The DS Client would then analyze all the changes received from the DS Server, and perform conflict resolution with its changes.

The DS Client would then perform another Sync with a Direction value of twoWay, sending the results of its conflict resolution as the DS Client changes. If this is performed shortly after the first Sync, there should be little or no new changes from the DS Server. If any new changes were received from the DS Server, conflict resolution would have to be performed on them, followed by another Sync to propagate the results (if they impacted the DS Server).

Note that Client Controlled Conflict Resolution should not be a normal situation, and should not be performed from firmware on a device, because of the risk of side effects. This should be reserved for situations such as the gradual migration of DS Server responsibility from one server to another. Note that if there is no need to run both DS Servers at the same time, a much simpler migration can be performed by just doing a Full Restore operation from the first DS Server, and then having the newer DS Server start working with that data.

7. Interoperability

7.1 Contacts Synchronization

DS Servers and DS Clients SHALL reduce the impacts of the interoperability issues as much as possible. The following is a non-exhaustive list of recommendations:

In general, the DS Server and DS Clients should be compliant with the vObject Minimum Interoperability Profile [VOBJMIP].

In particular, the following requirements should be applied to vCard 2.1 properties defined through [IMCVCARD]; as well as vCard 3.0 properties defined through [RFC2425], and [RFC2426]:

- Properties that do not have any value should not be included in the corresponding vCard object.
- The DS Client should support the BDAY property
- The DS Client should support the PHOTO property

If vCard 3.0 is supported (see [RFC2425] and [RFC2426]), the DS Client and DS Server should also support the following vCard Types:

- Geographical
- Nickname
- Sort String

In order to ensure interoperability among DS 2.0 implementations with regards to *dates* and *phone numbers*, at least the following formats must be understood by parsers in the DS Clients and DS Servers:

- For phone numbers (see [TS24008]) the Type of Number (TON) and Number Plan Identification (NPI) fields shall be understood.
- For dates, the following ISO formats (see [ISO8601]) shall be understood: “yyyymmddThhmmss”, and “yyyymmddTthtmZ”. Where *time* and *time zone* fields are optional and mutually exclusive:

Field	Value representation	Details
Date	yyyy	Year
	mm	Month
	dd	Day
Time	“T”	Fixed character indicating *time* data
	hh	Hours
	mm	Minutes
	ss	Seconds
Time Zone	“T”	Fixed character indicating *time* data
	th	Time Zone hours
	tm	Time Zone minutes
	“Z”	Fixed character indicating *time zone* data

Appendix A. Change History (Informative)

A.1 Approved Version 2.0 History

Reference	Date	Description
OMA-TS-DS_Concepts-V2_0-20110719-A	19 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0258-INP_DS_V2_0_ERP_for_final_Approval