Device Management OBEX Binding
Approved Version 1.3 – 24 May 2016

Open Mobile Alliance
OMA-TS-DM_OBEXBinding-V1_3-20160524-A
## Contents

1. SCOPE .......................................................................................................................................................... 5

2. REFERENCES ................................................................................................................................................. 6
   2.1 NORMATIVE REFERENCES ...................................................................................................................... 6
   2.2 INFORMATIVE REFERENCES .................................................................................................................. 6

3. TERMINOLOGY AND CONVENTIONS ....................................................................................................... 6
   3.1 CONVENTIONS ........................................................................................................................................ 7
   3.2 DEFINITIONS .......................................................................................................................................... 7
   3.3 ABBREVIATIONS .................................................................................................................................... 7

4. INTRODUCTION ............................................................................................................................................ 8

5. OBEX INTRODUCTION ................................................................................................................................. 9
   5.1 OBEX OVER IRDA .................................................................................................................................. 9
      5.1.1 IAS Entry ...................................................................................................................................... 10
   5.2 OBEX OVER BLUETOOTH .................................................................................................................... 11
      5.2.1 Bluetooth Service Discovery .......................................................................................................... 12
      5.2.2 Other Bluetooth Protocol Requirements .......................................................................................... 14

6. OBEX MAPPING TO DM MESSAGE ............................................................................................................. 15
   6.1 OBEX OPERATIONS ............................................................................................................................... 15
   6.2 OBEX CONNECTION OVERVIEW ....................................................................................................... 16
      6.2.1 Multiple Messages Per Package .................................................................................................... 17
      6.2.2 Type header requirement .............................................................................................................. 17
   6.3 OBEX CONNECTION ESTABLISHMENT ............................................................................................ 17
   6.4 EXCHANGING DM MESSAGES OVER THE OBEX CONNECTION ..................................................... 18
   6.5 OBEX DISCONNECTION ....................................................................................................................... 20
   6.6 OBEX ABORT ......................................................................................................................................... 21
   6.7 DM NOTIFICATION .............................................................................................................................. 21

7. EXAMPLES .................................................................................................................................................... 22
   7.1 OBEX CONNECT EXAMPLE ................................................................................................................... 22
   7.2 OBEX DISCONNECT EXAMPLE .......................................................................................................... 23
   7.3 OBEX ABORT EXAMPLE ...................................................................................................................... 23
   7.4 OBEX PUT EXAMPLE ........................................................................................................................... 23
   7.5 OBEX GET EXAMPLE .......................................................................................................................... 25

APPENDIX A. CHANGE HISTORY (INFORMATIVE) ...................................................................................... 26
   A.1 APPROVED VERSION HISTORY ............................................................................................................ 26

APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ................................................. 27
   B.1 SCR FOR CLIENT FEATURES ............................................................................................................... 27
      B.1.1 Common SCRs – OBEX Session Requirements ............................................................................... 27
      B.1.2 SCRs for Client Initiated Sessions ................................................................................................. 27
      B.1.3 SCRs for Server Initiated Sessions ............................................................................................... 27
   B.2 SCR FOR SERVER FEATURES ............................................................................................................ 27
      B.2.1 Common SCRs .................................................................................................................................. 27
      B.2.2 SCRs for Client Initiated Sessions ................................................................................................. 28
   B.3 SCRS FOR SERVER INITIATED SESSIONS .......................................................................................... 28

## Figures

Figure 1: OBEX over Bluetooth ...................................................................................................................... 11
Tables

Table 1: DM Server Service Records ................................................................. 12
Table 2: DM Client Service Records ................................................................. 13
Table 3: SDP PDUs ......................................................................................... 13
1. Scope

This document describes OBEX Binding for carrying DM Messages based on DM representation [DMREPRO]. The document uses the primitives and methods defined in the OBEX specification V1.2 as defined in [OBEX].
2. References

2.1 Normative References


2.2 Informative References

None.
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 DM DTD or XML snippets is specified in this typeface.

3.2 Definitions

Kindly consult [DMDICT] for all definitions used in this document.

3.3 Abbreviations

Kindly consult [DMDICT] for all abbreviations used in this document.
4. Introduction

This document describes the OBEX Binding for carrying DM Messages based on DM representation [DMREPRO]. The document uses the primitives and methods defined in the OBEX specification V1.2 [OBEX].
5. OBEX Introduction

OBEX [OBEX] is a protocol for exchanging objects. It was initially designed for infrared, but it has been adopted by Bluetooth, and is also used over RS232, USB and WAP.

OBEX is a session-oriented protocol, which allows multiple request/response exchanges in one session. An OBEX session is initiated by an OBEX CONNECT request, and is established when the other device returns a success response. The connection is terminated by sending a DISCONNECT request.

In this specification, the DM Client can work either as an OBEX client or as an OBEX server at the OBEX protocol layer. In consequence, the DM Server can work either as an OBEX client or as an OBEX server. The OBEX role depends on the fact which one, the DM Client or the DM Server, initiates the DM session.

When a session has been established, the data is transferred using the PUT request. The remote device acknowledges the data, by sending a response with a status code.

DM requires that an OBEX connection is established. Connectionless OBEX cannot be used with DM.

5.1 OBEX Over IrDA

The diagram below demonstrates the position of OBEX within the IrDA stack.

<table>
<thead>
<tr>
<th>SyncML Client</th>
<th>SyncML Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBEX</td>
<td>OBEX</td>
</tr>
<tr>
<td>Tiny TP</td>
<td>Tiny TP</td>
</tr>
<tr>
<td>IAS</td>
<td>IAS</td>
</tr>
<tr>
<td>IrLMP</td>
<td>IrLMP</td>
</tr>
<tr>
<td>IrLAP</td>
<td>IrLAP</td>
</tr>
<tr>
<td>IrDA Hardware</td>
<td>IrDA Hardware</td>
</tr>
</tbody>
</table>

e.g. Phone                  e.g. PC

IrLAP is the link level protocol.
IrLMP is a multiplexing layer.
Tiny TP provides flow control.
IAS is the Information Access Service.
OBEX includes both a session level protocol and an application framework.
5.1.1 IAS Entry

To enable an OBEX connection over IrDA, the OBEX protocol stack needs to provide IAS setting information to the IAS protocol stack. The DM Server and DM Client SHOULD use the following IAS entry settings for DM communication via OBEX over IrDA.

5.1.1.1 IAS Entry Settings for DM Server

DM Server SHOULD use the following IAS entry settings.

<table>
<thead>
<tr>
<th>Class</th>
<th>OBEX:DM-Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>IrDA:TinyTP:LsapSel</td>
</tr>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Description</td>
<td>IrLMP LSAP selector for DM over IrOBEX, legal values from 0x01 to 0x6F</td>
</tr>
</tbody>
</table>

5.1.1.2 IAS Entry Settings for DM Client

DM Client SHOULD use the following IAS entry settings.

<table>
<thead>
<tr>
<th>Class</th>
<th>OBEX:DM-Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>IrDA:TinyTP:LsapSel</td>
</tr>
<tr>
<td>Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Description</td>
<td>IrLMP LSAP selector for DM over IrOBEX, legal values from 0x01 to 0x6F</td>
</tr>
</tbody>
</table>
### 5.2 OBEX Over Bluetooth

The Bluetooth section is specified so that the DM Client MUST be able to function as either an OBEX client, or an OBEX server, or both. The DM Server MUST be able to function as both the OBEX server and client.

The figure below shows the protocols when DM and OBEX are run over the Bluetooth protocol stack.

![OBEX over Bluetooth](image)

**Figure 1: OBEX over Bluetooth**

The Baseband, LMP, and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10. SDP is the Bluetooth Service Discovery Protocol [BTSDP].

The DM Client layer shown in Figure 1 is the entity providing the sync client agent functionality. The DM Server is the SW providing the sync engine functionality.

In this specification, the DM Client can work either as an OBEX client or as an OBEX server at the OBEX protocol layer. In consequence, the DM Server can work either as an OBEX client or as an OBEX server. The OBEX role depends on the fact which one, the DM Client or the DM Server, initiates the DM session.
5.2.1 Bluetooth Service Discovery

To enable the OBEX connection over the Bluetooth protocol stack, the DM Server MUST advertise and the DM Client SHOULD advertise service records, which can be retrieved by a connecting device using the Bluetooth SDP [BTSDP].

In the case of the DM Server, the following information, i.e., service records MUST be put into the SDDB (Service Discovery DataBase).

<table>
<thead>
<tr>
<th>Item</th>
<th>Definition:</th>
<th>Type/Size:</th>
<th>Value:</th>
<th>AttrID:</th>
<th>Status:</th>
<th>Default Value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Class ID List</td>
<td></td>
<td>N/A</td>
<td>0x0001**</td>
<td>MUST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Class #0</td>
<td>DMServer</td>
<td>UUID</td>
<td>*</td>
<td>N/A</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Protocol Descriptor list</td>
<td></td>
<td>N/A</td>
<td>0x0004**</td>
<td>MUST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol ID #0</td>
<td>L2CAP</td>
<td>UUID</td>
<td>0x0100**</td>
<td>N/A</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Protocol ID #1</td>
<td>RFCOMM</td>
<td>UUID</td>
<td>0x0003**</td>
<td>N/A</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Param #0</td>
<td>CHANNEL</td>
<td>Uint8</td>
<td>Varies</td>
<td>N/A</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Protocol ID #2</td>
<td>OBEX</td>
<td>UUID</td>
<td>0x0008**</td>
<td>N/A</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Service name</td>
<td>Displayable Text name</td>
<td>String</td>
<td>Varies</td>
<td>0x0000+b***</td>
<td>MAY</td>
<td>&quot;DM Server&quot;</td>
</tr>
</tbody>
</table>

Table 1: DM Server Service Records

* The value 00000001-0000-1000-8000-0002EE000002 SHOULD be used in this place.
** The value or the attribute ID is specified in the Bluetooth Assigned Numbers specification [BTAN].
*** 'b' in this table represents a base offset as given by the LanguageBaseAttributeIDList attribute. For the principal language b MUST be equal to 0x0100 as described in the Bluetooth SDP specification [BTSDP].

The service records, which the DM Client SHOULD put into its SDDB, are listed below.
<table>
<thead>
<tr>
<th>Item</th>
<th>Definition:</th>
<th>Type/ Size:</th>
<th>Value:</th>
<th>AttrID:</th>
<th>Status:</th>
<th>Default Value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Class ID List</td>
<td></td>
<td></td>
<td>N/A</td>
<td>0x0001**</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Service Class #0</td>
<td>DMClient</td>
<td>UUID</td>
<td>*</td>
<td>N/A</td>
<td>MUST</td>
<td>N/A</td>
</tr>
<tr>
<td>Protocol Descriptor list</td>
<td></td>
<td></td>
<td>N/A</td>
<td>0x0004**</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td>Protocol ID #0</td>
<td>L2CAP</td>
<td>UUID</td>
<td>0x0100**</td>
<td>N/A</td>
<td>MUST</td>
<td>N/A</td>
</tr>
<tr>
<td>Protocol ID #1</td>
<td>RFCOMM</td>
<td>UUID</td>
<td>0x0003**</td>
<td>N/A</td>
<td>MUST</td>
<td>N/A</td>
</tr>
<tr>
<td>Param #0</td>
<td>CHANNEL</td>
<td>Uint8</td>
<td>Varies</td>
<td>N/A</td>
<td>MUST</td>
<td>N/A</td>
</tr>
<tr>
<td>Protocol ID #2</td>
<td>OBEX</td>
<td>UUID</td>
<td>0x0008**</td>
<td>N/A</td>
<td>MUST</td>
<td>N/A</td>
</tr>
<tr>
<td>Service name</td>
<td>Displayable</td>
<td>String</td>
<td>Varies</td>
<td>0x0000+b***</td>
<td>MAY</td>
<td>&quot;DM Client&quot;</td>
</tr>
</tbody>
</table>

Table 2: DM Client Service Records

* The value 00000002-0000-1000-8000-0002EE000002 SHOULD be used in this place.
** The value or the attribute ID is specified in the Bluetooth Assigned Numbers specification [BTA N].
*** 'b' in this table represents a base offset as given by the LanguageBaseAttributeIDList attribute. For the principal language b MUST be equal to 0x0100 as described in the Bluetooth SDP specification [BTSDP].

5.2.1.1 SDP Protocol Data Units

Table 3 shows the specified SDP PDUs (Protocol Data Units), which are REQUIRED.

<table>
<thead>
<tr>
<th>PDU no.</th>
<th>SDP PDU</th>
<th>Ability to Send</th>
<th>Ability to Retrieve</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DM Client</td>
<td>DM Server</td>
</tr>
<tr>
<td>1</td>
<td>SdpErrorResponse</td>
<td>MUST*</td>
<td>MUST</td>
</tr>
<tr>
<td>2</td>
<td>SdpServiceSearchAttribute-Request</td>
<td>MUST**</td>
<td>MUST</td>
</tr>
<tr>
<td>3</td>
<td>SdpServiceSearchAttribute-Response</td>
<td>MUST*</td>
<td>MUST</td>
</tr>
</tbody>
</table>

Table 3: SDP PDUs

* This is only applicable if the DM Client is able to function as the OBEX server.
** This is only applicable if the DM Client is able to function as the OBEX client.
5.2.2 Other Bluetooth Protocol Requirements

This specification partially requires compliance to the Bluetooth Serial Port (SeP) Profile [BTSEP] if Bluetooth is used as a physical medium for OBEX. These are:

- The compliance is REQUIRED to the RFCOMM requirements as defined in Chapter 4 in the SeP Profile.
- The compliance is REQUIRED to the L2CAP requirements as defined in Chapter 5 in the SeP Profile.
- The compliance is REQUIRED to the LM protocol requirements as defined in Chapter 7 in the SeP Profile.

The SDP requirements are defined by this specification and thus, any of the requirements defined in the SeP profile (Chapter 6 in the SeP profile) does not apply to this specification. The DM Server MUST comply with both the Device 'A' and Device 'B' requirements as defined in the SeP Profile. The DM Client MUST comply with either the Device 'A' requirements, or with the Device 'B' requirements, or both as defined in the SeP Profile.

The Bluetooth LC (Link Controller) capabilities and The Bluetooth Generic Access Profile (GAP) requirements for this specification are defined in Chapter 6.5 and Chapter 7 of the Bluetooth GOEP [BTGOEP], respectively. The DM Server MUST comply with both the client and server requirements as defined in Chapter 6.5 and Chapter 7 in the GOEP. The DM Client MUST comply with either the client requirements, or the server requirements, or both as defined in Chapter 6.5 and Chapter 7 in the GOEP.
6. OBEX Mapping to DM Message

The following sections define the requirements for the binding of DM Message to OBEX.

In client initiated sync, the DM Client initiates the OBEX link, so it is also the OBEX client. The DM Client can disconnect the OBEX link when it has received the last DM Message from the DM Server.

With server alerted sync, the DM Server initiates the OBEX link, so it is the OBEX client. The DM Server cannot disconnect the OBEX link before it has received the SyncML response message for the last DM Message.

6.1 OBEX Operations

The OBEX layer SHOULD be disconnected using the OBEX DISCONNECT operation. The OBEX specification also allows the link to be disconnected by disconnecting the underlying transport layer.

The OBEX connection can be authenticated as part of the OBEX CONNECT request/response messages, using the authenticate challenge and response headers.

The OBEX client MAY send an OBEX ABORT request, to terminate a multi-packet operation (such as PUT) before it would normally end.

The following OBEX operations are REQUIRED for Client Initiated sessions.

<table>
<thead>
<tr>
<th>OBEX Operation</th>
<th>DM Server / OBEX Server</th>
<th>DM Client / OBEX Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Disconnect</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Put</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Get</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Abort</td>
<td>MUST</td>
<td>MAY</td>
</tr>
</tbody>
</table>

The following OBEX operations are REQUIRED for Server Initiated sessions.

<table>
<thead>
<tr>
<th>OBEX Operation</th>
<th>DM Server / OBEX Client</th>
<th>DM Client / OBEX Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Disconnect</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Put</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Get</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Abort</td>
<td>MAY</td>
<td>MUST</td>
</tr>
</tbody>
</table>
6.2 OBEX Connection Overview

The OBEX connection is made at the start of the DM session, and remains open until the DM session has completed.

The following example shows the creation of an OBEX connection, the mapping of PUT and GET requests to the DM Message transfers, and the OBEX disconnection.

This example is not intended to show a complete a DM Session but merely illustrates the use of PUT and GET within a DM OBEX binding implementation.

<table>
<thead>
<tr>
<th>OBEX Client</th>
<th>OBEX Server</th>
<th>Message Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Success Response</td>
<td></td>
</tr>
<tr>
<td>PUT Request</td>
<td></td>
<td>DM Message from</td>
</tr>
<tr>
<td></td>
<td>Continue Response</td>
<td>OBEX Client to OBEX Server</td>
</tr>
<tr>
<td>PUT Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Continue Response</td>
<td></td>
</tr>
<tr>
<td>PUT Final Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Success Response</td>
<td></td>
</tr>
<tr>
<td>GET Request</td>
<td></td>
<td>DM Message from</td>
</tr>
<tr>
<td></td>
<td>Continue Response</td>
<td>OBEX Server to OBEX Client</td>
</tr>
<tr>
<td>GET Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Continue Response</td>
<td></td>
</tr>
<tr>
<td>GET Final Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Success Response</td>
<td></td>
</tr>
<tr>
<td>DISCONNECT Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Success Response</td>
<td></td>
</tr>
</tbody>
</table>
6.2.1 Multiple Messages Per Package

Each DM Message MUST be transferred as a DM MIME media type within the body of the OBEX request or response. However in order to transfer the message the OBEX / transport layer MAY split the message into many PUT requests, followed by a PUT Final Request. When there are multiple DM Messages per DM package to transfer, each message is transferred in a separate 'set' of PUT/GET commands; depending on whether it is a DM request or response.

The recipient of a DM Message can determine if there are more DM Messages in the package by the absence of the Final element in the last received DM Message. When the recipient receives a DM Message with the Final element, it is the final message within that SyncML package.

Similarly if the PUT is not a PUT final then the recipient knows it is not the final part of the DM Message, or if the response to the GET Final Request is not an OK/success then there is more data still to transfer.

6.2.2 Type header requirement

DM Clients and DM Servers MUST support this header with either the "application/vnd.syncml.dm+xml" or "application/vnd.syncml.dm+wbxml" MIME media type values.

In the case of DM Notification, the Type header of the OBEX packet transferring the notification message MUST include the MIME media type, "application/vnd.syncml.dm.notification".

6.3 OBEX Connection Establishment

The OBEX connection is established by the DM Client or DM Server generating a Connect Request, and the remote device indicates that the connection has been established, by returning a Connect Response. For each DM Session, a separate OBEX connection MUST be established.

The OBEX CONNECT request packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Opcode for CONNECT</td>
<td>0x80</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>OBEX Version Number</td>
<td>0x10</td>
<td>M</td>
<td>In [OBEX] it is stated that current version is 1.0</td>
</tr>
<tr>
<td>Field</td>
<td>Flags</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Max OBEX Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Target</td>
<td>Varies</td>
<td>M</td>
<td>The UUID to be used in device management is SYNCML-DM.</td>
</tr>
</tbody>
</table>

The OBEX CONNECT response packet contains the following fields and headers:
### Field/Header | Name | Value | M/O | Explanation |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Response code for CONNECT request</td>
<td>0x0A</td>
<td>M</td>
<td>0xA0 for success, otherwise fail</td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>OBEX Version Number</td>
<td>0x10</td>
<td>M</td>
<td>In [OBEX] it is stated that current version is 1.0</td>
</tr>
<tr>
<td>Field</td>
<td>Flags</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Max OBEX Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Connection ID</td>
<td>Varies</td>
<td>M</td>
<td>Connection ID is set by the Server during the OBEX Connect operation as a shorthand way for the client to direct the requests. <strong>This MUST be the first header.</strong></td>
</tr>
<tr>
<td>Header</td>
<td>Who</td>
<td>Varies</td>
<td>M</td>
<td>The UUID returned is the same UUID that was sent in the connect request target header</td>
</tr>
</tbody>
</table>

#### 6.4 Exchanging DM Messages over the OBEX Connection

Once an OBEX connection has been established, DM Messages can be transferred over the link.

The OBEX PUT request packet contains the following fields and headers:

### Field/Header | Name | Value | M/O | Explanation |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Opcode for PUT</td>
<td>0x02 or 0x82</td>
<td>M</td>
<td>0x02 is used for packets previous to the last put packet. 0x82 (which is 0x02 with the high bit set) is used for the last put packet.</td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Connection ID</td>
<td>Varies</td>
<td>M</td>
<td>Connection ID is set to the value returned by the Server during the OBEX Connect operation. This MUST be the first header. Sent only once in first packet in the request sequence.</td>
</tr>
<tr>
<td>Header</td>
<td>Type</td>
<td>Varies</td>
<td>M</td>
<td>The MIME type of the object. This MUST contain the DM MIME type declaration. Sent only once in first packet in the request sequence, MUST precede object Body headers.</td>
</tr>
<tr>
<td>Header</td>
<td>Length</td>
<td>Varies</td>
<td>O</td>
<td>Length of the object. This header is OPTIONAL but highly RECOMMENDED. Sent only once in the request sequence.</td>
</tr>
<tr>
<td>Header</td>
<td>Body/End of Body</td>
<td>Varies</td>
<td>M</td>
<td>End of Body identifies the last chunk of the object body. End of Body header (PUT final packet) SHOULD be sent with an empty body.</td>
</tr>
</tbody>
</table>
The OBEX PUT response packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/ Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Response code for PUT</td>
<td>0x90, 0xA0, 0xCD, 0xCF, ...</td>
<td>M</td>
<td>0x90 for continue 0xA0 for success 0xCD if the object is too large 0xCF if the object type is not supported</td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
</tbody>
</table>

Other headers, which can be optionally used, are found in [OBEX]

The OBEX GET request packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/ Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Opcode for GET</td>
<td>0x03 or 0x83</td>
<td>M</td>
<td>0x03 is used for packets previous to the last packet containing headers.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0x83 (which is 0x03 with the final bit set) is used for the last packet containing headers and for subsequent get packets in the current Get request sequence without headers. No headers can be sent in Get request packets once the 0c83 has been sent in a previous packet.</td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Connection ID</td>
<td>Varies</td>
<td>M</td>
<td>Connection ID is set to the value returned by the Server during the OBEX Connect operation. This MUST be the first header. Sent only once in first packet in the request sequence.</td>
</tr>
<tr>
<td>Header</td>
<td>Type</td>
<td>0x42, ...</td>
<td>M</td>
<td>The MIME type of the object. This MUST contain the DM MIME type declaration. Sent only once in first packet in the request sequence.</td>
</tr>
</tbody>
</table>
The OBEX GET response packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Response code for GET</td>
<td>0x90, 0xA0, 0xC0, 0xC3, ...</td>
<td>M</td>
<td>0x90 for continue 0xA0 for success 0xC0 bad request 0xC3 forbidden</td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Length</td>
<td>0xC3, ...</td>
<td>O</td>
<td>Length of the object. This header is OPTIONAL but highly RECOMMENDED. Sent only once in the request sequence.</td>
</tr>
<tr>
<td>Header</td>
<td>Body/End of Body</td>
<td>0x48/0x49, ...</td>
<td>M</td>
<td>End of Body identifies the last chunk of the object body.</td>
</tr>
</tbody>
</table>

Other headers, which can be optionally used, are found in [OBEX].

### 6.5 OBEX Disconnection

The OBEX connection is disconnected by the DM Client or DM Server, generating a Disconnect Request, and the remote device indicates that the connection has been terminated, by returning a success Response.

The OBEX DISCONNECT request packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Opcode for DISCONNECT</td>
<td>0x81</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Connection ID</td>
<td>Varies</td>
<td>M</td>
<td>Connection ID is set to the value returned by the Server during the OBEX Connect operation. This MUST be the first header.</td>
</tr>
</tbody>
</table>

Other headers (such as Description) which can be optionally used are found in [OBEX].

The OBEX DISCONNECT response packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Response code for DISCONNECT</td>
<td>0xA0</td>
<td>M</td>
<td>0xA0 for success, otherwise fail</td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
</tbody>
</table>
6.6  OBEX ABORT

The client can send an OBEX abort request to terminate a multi-packet operation (such as PUT) before it would normally end. The ABORT request and response always fit in one OBEX packet, and they always have the Final bit set.

The OBEX ABORT request packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Opcode for ABORT</td>
<td>0xFF</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Header</td>
<td>Connection ID</td>
<td>Varies</td>
<td>O</td>
<td>Connection ID is set to the value returned by the Server during the OBEX Connect operation. This MUST be the first header if it is included.</td>
</tr>
</tbody>
</table>

Other headers (such as Description) which can be optionally used are found in [OBEX].

The OBEX ABORT response packet contains the following fields and headers:

<table>
<thead>
<tr>
<th>Field/Header</th>
<th>Name</th>
<th>Value</th>
<th>M/O</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Response code for ABORT</td>
<td>0xA0</td>
<td>M</td>
<td>0xA0 for success, otherwise fail and the client SHOULD disconnect the OBEX connection.</td>
</tr>
<tr>
<td>Field</td>
<td>Packet Length</td>
<td>Varies</td>
<td>M</td>
<td></td>
</tr>
</tbody>
</table>

Other headers (such as Description) which can be optionally used are found in [OBEX].

6.7  DM Notification

The OBEX connection is established by the DM Server generating a CONNECT Request. The OBEX CONNECT request MUST have as the Target header a UUID of SYNCML-DM for Device Management packages. Once an OBEX connection has been established the DM Notification message data can be transferred over the link.
# 7. Examples

The following examples are formatted as the examples in [OBEX].

## 7.1 OBEX Connect Example

<table>
<thead>
<tr>
<th>Client Request:</th>
<th>Bytes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opcode</td>
<td>0x80</td>
<td>CONNECT, Final bit set</td>
</tr>
<tr>
<td></td>
<td>0x0015</td>
<td>packet length = 21</td>
</tr>
<tr>
<td></td>
<td>0x10</td>
<td>version 1.0 of OBEX</td>
</tr>
<tr>
<td></td>
<td>0x00</td>
<td>flags, all zero for this version of OBEX</td>
</tr>
<tr>
<td></td>
<td>0x0200</td>
<td>512 bytes is the max OBEX packet size client can accept</td>
</tr>
<tr>
<td></td>
<td>0x46</td>
<td>HI for Target header</td>
</tr>
<tr>
<td></td>
<td>0x000E</td>
<td>Length of Target header</td>
</tr>
<tr>
<td></td>
<td>0x53594E434D4C</td>
<td>UUID for SyncML DM (“SYNCML-DM”)</td>
</tr>
<tr>
<td></td>
<td>2D444D</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server Response:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>response code</td>
<td>0xA0</td>
</tr>
<tr>
<td></td>
<td>0x001A</td>
</tr>
<tr>
<td></td>
<td>0x10</td>
</tr>
<tr>
<td></td>
<td>0x00</td>
</tr>
<tr>
<td></td>
<td>0x0200</td>
</tr>
<tr>
<td></td>
<td>0xCB</td>
</tr>
<tr>
<td></td>
<td>0x00000001</td>
</tr>
<tr>
<td></td>
<td>0x4A</td>
</tr>
<tr>
<td></td>
<td>0x000E</td>
</tr>
<tr>
<td></td>
<td>0x53594E434D4C</td>
</tr>
<tr>
<td></td>
<td>2D444D</td>
</tr>
</tbody>
</table>
7.2 OBEX Disconnect Example

<table>
<thead>
<tr>
<th>Client Request:</th>
<th>bytes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>opcode</td>
<td>0x81</td>
<td>DISCONNECT, Final bit set</td>
</tr>
<tr>
<td></td>
<td>0x0008</td>
<td>packet length = 8</td>
</tr>
<tr>
<td></td>
<td>0xCB</td>
<td>HI for Connection Id header</td>
</tr>
<tr>
<td></td>
<td>0x00000001</td>
<td>ConnId = 1</td>
</tr>
</tbody>
</table>

Server Response:

<table>
<thead>
<tr>
<th>response code</th>
<th>0xA0</th>
<th>SUCCESS, Final bit set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0x0003</td>
<td>packet length = 3</td>
</tr>
</tbody>
</table>

7.3 OBEX Abort Example

<table>
<thead>
<tr>
<th>Client Request:</th>
<th>bytes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>opcode</td>
<td>0xFF</td>
<td>ABORT, Final bit set</td>
</tr>
<tr>
<td></td>
<td>0x0008</td>
<td>packet length = 8</td>
</tr>
<tr>
<td></td>
<td>0xCB</td>
<td>HI for Connection Id header</td>
</tr>
<tr>
<td></td>
<td>0x00000001</td>
<td>ConnId = 1</td>
</tr>
</tbody>
</table>

Server Response:

<table>
<thead>
<tr>
<th>response code</th>
<th>0xA0</th>
<th>SUCCESS, Final bit set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0x0003</td>
<td>packet length = 3</td>
</tr>
</tbody>
</table>

7.4 OBEX Put Example

<table>
<thead>
<tr>
<th>Client Request:</th>
<th>Bytes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>opcode</td>
<td>0x02</td>
<td>PUT, Final bit not set</td>
</tr>
<tr>
<td></td>
<td>0x0200</td>
<td>512 bytes is length of packet</td>
</tr>
<tr>
<td></td>
<td>0xCB</td>
<td>HI for Connection Id header</td>
</tr>
<tr>
<td></td>
<td>0x00000001</td>
<td>ConnId = 1</td>
</tr>
<tr>
<td></td>
<td>0x42</td>
<td>HI for Type header</td>
</tr>
<tr>
<td></td>
<td>0x0020</td>
<td>Length of Type header</td>
</tr>
<tr>
<td></td>
<td>0x6170706C6963</td>
<td>Type of object (application/vnd.syncml.dm+wbxml)</td>
</tr>
<tr>
<td></td>
<td>6174696F6E2F76</td>
<td>(null terminated ASCII text)</td>
</tr>
<tr>
<td></td>
<td>6E642E73796E63</td>
<td></td>
</tr>
<tr>
<td>6D6C2E646D2B776 2786D 6C00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xC3 HI for <strong>Length</strong> header</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x000004FE Length of object is 1278 bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x48 HI for Object <strong>Body</strong> chunk header</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x01D3 Length of <strong>Body</strong> header = 467. 464 plus HI and header length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x........ 464 bytes of body</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Server Response:**

<table>
<thead>
<tr>
<th>response code</th>
<th>0x90 CONTINUE, Final bit set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0003 length of response packet</td>
<td></td>
</tr>
</tbody>
</table>

**Client Request:**

<table>
<thead>
<tr>
<th>opcode</th>
<th>0x02 <strong>PUT</strong>, Final bit not set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0200 512 bytes is length of packet</td>
<td></td>
</tr>
<tr>
<td>0x48 HI for Object <strong>Body</strong> chunk</td>
<td></td>
</tr>
<tr>
<td>0x01FD Length of <strong>Body</strong> header = 509. 506 plus HI and header length</td>
<td></td>
</tr>
<tr>
<td>0x........ next 506 bytes of body</td>
<td></td>
</tr>
</tbody>
</table>

**Server Response:**

<table>
<thead>
<tr>
<th>response code</th>
<th>0x90 CONTINUE, Final bit set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0003 length of response packet</td>
<td></td>
</tr>
</tbody>
</table>

**Client Request:**

<table>
<thead>
<tr>
<th>opcode</th>
<th>0x82 <strong>PUT</strong>, Final bit set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x013A 314 bytes is length of packet</td>
<td></td>
</tr>
<tr>
<td>0x49 HI for Object <strong>End-of-Body</strong> chunk</td>
<td></td>
</tr>
<tr>
<td>0x0137 Length of header = 311. 308 plus HI and header length</td>
<td></td>
</tr>
<tr>
<td>0x........ last 308 bytes of body</td>
<td></td>
</tr>
</tbody>
</table>

**Server Response:**

<table>
<thead>
<tr>
<th>response code</th>
<th>0xA0 SUCCESS, Final bit set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0003 length of response packet</td>
<td></td>
</tr>
</tbody>
</table>
### 7.5 OBEX Get Example

<table>
<thead>
<tr>
<th>Client Request:</th>
<th>bytes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>opcode</strong></td>
<td>0x83</td>
<td>GET, Final bit set</td>
</tr>
<tr>
<td></td>
<td>0x0028</td>
<td>40 bytes is length of GET packet</td>
</tr>
<tr>
<td></td>
<td>0xCB</td>
<td>HI for Connection ID header</td>
</tr>
<tr>
<td></td>
<td>0x00000001</td>
<td>ConnId = 1</td>
</tr>
<tr>
<td></td>
<td>0x42</td>
<td>HI for Type header</td>
</tr>
<tr>
<td></td>
<td>0x0020</td>
<td>Length of Type header</td>
</tr>
<tr>
<td></td>
<td>0x6170706C6963</td>
<td>Type of object (application/vnd.syncml.dm+wbxml)</td>
</tr>
<tr>
<td></td>
<td>6174696F62F76</td>
<td>(null terminated ASCII text)</td>
</tr>
<tr>
<td></td>
<td>6E642E7396E63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6D6C2E646D</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2B7762786D6C00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server Response:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response code</strong></td>
<td>0x90</td>
</tr>
<tr>
<td></td>
<td>0x0200</td>
</tr>
<tr>
<td></td>
<td>0xC3</td>
</tr>
<tr>
<td></td>
<td>0x000002BE</td>
</tr>
<tr>
<td></td>
<td>0x48</td>
</tr>
<tr>
<td></td>
<td>0x01F8</td>
</tr>
<tr>
<td></td>
<td>0x..........</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Client Request:</th>
<th>bytes</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>opcode</strong></td>
<td>0x83</td>
<td>GET, Final bit set</td>
</tr>
<tr>
<td></td>
<td>0x0003</td>
<td>3 bytes is length of GET packet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server Response:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Response code</strong></td>
<td>0xA0</td>
</tr>
<tr>
<td></td>
<td>0x0038</td>
</tr>
<tr>
<td></td>
<td>0x49</td>
</tr>
<tr>
<td></td>
<td>0x00CC</td>
</tr>
<tr>
<td></td>
<td>0x..........</td>
</tr>
</tbody>
</table>
### Appendix A. Change History

#### A.1 Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA-TS-DM_OBEXBinding-V1_3-20160524-A</td>
<td>24 May 2016</td>
<td>Status changed to Approved by TP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TP Ref # OMA-TP-2016-0041R01-INP_DM_V1_3_ERP_for_final_Approval</td>
</tr>
</tbody>
</table>
Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES]. The static conformance requirements are specified for Client Initiated and Server Initiated Sessions.

B.1 SCR for Client Features

B.1.1 Common SCRs – OBEX Session Requirements

The following OBEX operations are REQUIRED for all sessions.

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSDM-OBEX-C-001-M</td>
<td>Support for OBEX Connect operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
<tr>
<td>DSDM-OBEX-C-002-M</td>
<td>Support for OBEX Disconnect operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
<tr>
<td>DSDM-OBEX-C-003-M</td>
<td>Support for OBEX Put operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
<tr>
<td>DSDM-OBEX-C-004-M</td>
<td>Support for OBEX Get operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
</tbody>
</table>

B.1.2 SCRs for Client Initiated Sessions

The following OBEX operations are REQUIRED for Client Initiated sessions.

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSDM-OBEX-C-005-O</td>
<td>DM Client/OBEX Client support for OBEX Abort operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
</tbody>
</table>

B.1.3 SCRs for Server Initiated Sessions

The following OBEX operations are REQUIRED for Server Initiated sessions.

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSDM-OBEX-C-006-M</td>
<td>DM Client/OBEX Server support for OBEX Abort operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
</tbody>
</table>

B.2 SCR for Server Features

B.2.1 Common SCRs

The following OBEX operations are REQUIRED for all sessions.

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSDM-OBEX-S-001-M</td>
<td>Support for OBEX Connect operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
<tr>
<td>DSDM-OBEX-S-002-M</td>
<td>Support for OBEX Disconnect operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
<tr>
<td>DSDM-OBEX-S-003-M</td>
<td>Support for OBEX Put operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
<tr>
<td>DSDM-OBEX-S-004-M</td>
<td>Support for OBEX Get operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
</tbody>
</table>
### B.2.2 SCRs for Client Initiated Sessions

The following OBEX operations are REQUIRED for Client Initiated sessions.

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSDM-OBEX-S-005-M</td>
<td>DM Server/OBEX Server support for OBEX Abort operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
</tbody>
</table>

### B.3 SCRs for Server Initiated Sessions

The following OBEX operations are REQUIRED for Server Initiated sessions.

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSDM-OBEX-S-006-O</td>
<td>DM Server/OBEX Client support for OBEX Abort operation</td>
<td>Section 6.1</td>
<td></td>
</tr>
</tbody>
</table>