Enabler Test Specification for Device Management

Candidate Version 1.2 – 19 Aug 2010

Open Mobile Alliance

OMA-ETS-DM-V1_2-20100819-C

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Use of this document is subject to all of the terms and conditions of the Use Agreement located at http://www.openmobilealliance.org/UseAgreement.html.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the "OMA IPR Declarations" list at http://www.openmobilealliance.org/ipr.html. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE "OMA IPR DECLARATIONS" LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. OMA-Template-EnablerTestSpec-20070101-]

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

1. Scope

This document describes in detail available test cases for Device Management 1.2 Enabler Release, http://www.openmobilealliance.org/.

The test cases are split in two categories, conformance and interoperability test cases.

The conformance test cases are aimed to verify the adherence to normative requirements described in the technical specifications.

The interoperability test cases are aimed to verify that implementations of the specifications work satisfactory.

If either conformance or interoperability tests do not exists at the creation of the test specification this part should be marked not available.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

2. References

2.1 Normative References

[ERELD] ”Enabler Release Definition for Device Management Version 1.2”, Open Mobile Alliance™, ERELD-DM-V1_2. URL:http:www.openmobilealliance.org

[DMPRO] “OMA Device Management Protocol, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-Protocol-V1_2. URL:http://www.openmobilealliance.org

[DMREPU] “OMA Device Management Representation Protocol, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-RepPro-V1_2. URL:http://www.openmobilealliance.org

[DMSEC] “OMA Device Management Security, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-Security-V1_2. URL:http://www.openmobilealliance.org

[DMTND] “OMA Device Management Tree and Description, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-TND-V1_2. URL:http://www.openmobilealliance.org

[DMSTDOBJ] “OMA Device Management Standardized Objects, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-StdObj-V1_2. URL:http://www.openmobilealliance.org

[DMBOOT] “OMA Device Management Bootstrap, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-Bootstrap-V1_2. URL:http://www.openmobilealliance.org

[DMNOTI] “OMA Device Management Notification Initiated Session, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-Notification-V1_2. URL:http://www.openmobilealliance.org

[DMTNDS] “OMA Device Management Tree and Description Serialization, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM-TNDS-V1_2. URL:http://www.openmobilealliance.org

[ELREDSC] “Enabler Release Definition for SyncML Common Specifications, Version 1.2”. Open Mobile Alliance™. OMA-ERELD-SyncML-Common-V1_2. URL:http://www.openmobilealliance.org

[PROVSC] “Provisioning Smartcard, Version 1.1”, Open Mobile Alliance™, OMA-WAP-TS-ProvSC-V1_1, URL:http://www.openmobilealliance.org

[REPPRO] “SyncML Representation Protocol Version 1.2”, Open Mobile Alliance™, OMA-SyncML-RepPro-V1_2, URL:http://www.openmobilealliance.org

[SAN] “SyncML Server Alerted Notification Version 1.2”, Open Mobile Alliance™, OMA-SyncML-SAN-V1_2, URL:http://www.openmobilealliance.org

[SYNCHTTP] “SyncML HTTP Binding Specification Version 1.2”, Open Mobile AllianceTM, OMA-SyncML-HTTPBinding- V1_2, URL:http://www.openmobilealliance.org

[SYNCMETA] “SyncML Meta Information, Version 1.2”. Open Mobile Alliance™. OMA-SyncML-MetaInfo-V1_2, URL:http://www.openmobilealliance.org

[SYNCOBEX] “SyncML OBEX Binding Specification Version 1.2”, Open Mobile AllianceTM, OMA-SyncML-OBEXBinding- V1_2, URL:http://www.openmobilealliance.org

[SYNCWSP] “SyncML WSP Binding Specification Version 1.2”, Open Mobile AllianceTM , OMA-SyncML-WSPBinding- V1_2, URL:http://www.openmobilealliance.org

[IOPPROC] “OMA Interoperability Policy and Process Version 1.6”, Open Mobile Alliance™, OMA-IOP-Process-V1_6, URL:http//www.openmobilealliance.org

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

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

2.2 Informative References

[OMADICT] "Dictionary for OMA specifications Version 2.6". Open Mobile Alliance™. OMA-ORG-Dictionary- V2_6. URL: http://www.openmobilealliance.org/

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

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”, are normative, unless they are explicitly indicated to be informative.

The following numbering scheme is used:

xxx-y.z-con-number where:

xxx	Name of enabler, e.g. MMS or Browsing
y.z	Version of enabler release, e.g. 1.2 or 1.2.1
’con’	Indicating this test is a conformance test case
number	Leap number for the test case

Or

xxx-y.z-int-number where:

xxx	Name of enabler, e.g. MMS or Browsing
y.z	Version of enabler release, e.g. 1.2 or 1.2.1
’int’	Indicating this test is a interoperability test case
number	Leap number for the test case

3.2 Definitions

SCTS	SyncML Conformance Test Suite.
Test Object	The implementation under test is refered to as the Test Object. In this document, the Client.
Test Case	A Test Case is an individual test used to verify the conformance of the Test Object to a particular mandatory feature of the protocol. A 4-digit number identifies Test Cases where the first two digits denote the Test Group ID.
Test Group	A Test Group is a collection of Test Cases, which are executed, in a single SyncML session in SCTS conformance test tool.
<Node>	Path from the root to the interior node that is configured to the SCTS before the testing is done (e.g.. ‘./SyncML/DMAcc’ or ‘./DevDetail’). Test case is driven to this configured interior node. The <Node> can be different between different Test Cases.
<Leaf> or <Leaf#n>	Leaf node(s) that is configured to the SCTS before the testing is done (e.g.. ‘SwV’ and/or ‘Name’). Test case is driven to this configured interior node. The <Leaf> can be different between different Test Cases.

3.3 Abbreviations

OMA	Open Mobile Alliance
SCTS	SyncML Conformance Test Suite
DM	Device Management

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

4. Introduction

This document describes in detail available test cases for Device Management 1.2 Enabler Release, http://www.openmobilealliance.org/.

The test cases are split in two categories, conformance and interoperability test cases.

The conformance test cases are aimed to verify the adherence to normative requirements described in the technical specifications.

The interoperability test cases are aimed to verify that implementations of the specifications work satisfactory.

If either conformance or interoperability tests do not exists at the creation of the test specification this part should be marked not available.

If an implementation states in their ICS that an optional feature is supported. Then the tests for the optional feature are mandatory for that implementation.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5. Device Management Client Conformance Test Cases

5.1 Device Management Client Conformance Test Group #1

5.1.1 DeviceManagement-v1.2-client-con-0102

Test Case Id	DeviceManagement-v1.2-client-con-0102
Test Object	Client device
Test Case Description	To check if the Test Object sent a valid Alert command.
Specification Reference	[DMREPU] Chapter 6.6.2 [DMREPU] Chapter 7 (Alert Codes) [DMPRO] Chapter 8.3
SCR Reference	DMREPPRO-PCE-C-001 Support for sending 'Alert'
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The Client is triggered to initiate a request with the server 2. The client sends a Setup-Request. 3. Test Tool sends OK response to the client to close Session
Pass-Criteria	The Test Object MUST send valid Client Initiated Alert. Step 2: The Test Object MUST send a request with a valid client initiated alert. Valid implies: 1. The Alert tag must have as sub elements a CmdID tag and a Data tag The value of the data tag must be 1201 showing that this is a Client initiated session.

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The client is triggered to initiate communication with the server.
2	→		Setup-Request	The client sends a Setup-Request.
3	←		Setup-Response	Test Tool sends OK response to the client to close Session

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.1.2 DeviceManagement-v1.2-client-con-0103

Test Case Id	DeviceManagement-v1.2-client-con-0103
Test Object	Client device
Test Case Description	To check if the Test Object sends Device Information
Specification Reference	[DMREPU] Chapter 6.6.11 [DMPRO] Chapter 8.3
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace'
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The client is triggered to initiate communication with the server. 2. The client sends a Setup-Request. 3. Test Tool sends OK response to the client to close Session
Pass-Criteria	The Test Object MUST send its Device Information in a Replace command Step 2: The Test Object MUST send its Device Information in a Replace command. This implies that: The setup request shall contain a Replace tag which contains a CmdID tag and elements from the ./DevInfo node. The latter represent Device Information.

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The client is triggered to initiate communication with the server.
2	→		Setup-Request	The client sends a Setup-Request.
3	←		Setup-Response	Test Tool sends OK response to the client to close Session

5.1.3 DeviceManagement-v1.2-client-con-0104

Test Case Id	DeviceManagement-v1.2-client-con-0104
Test Object	Client device
Test Case Description	To check if the client's Source LocURI is same as the value in ./DevInfo/DevId

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Specification Reference	[DMREPU] Chapter 6.1.10 [DMREPU] Chapter 6.6.11 [DMPRO] Chapter 8.3
SCR Reference	DMREPPRO-CUE-C-008 Support for 'LocURI'
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The client is triggered to initiate communication with the server. 2. The client sends a Setup-Request. 3. Test Tool sends OK response to the client to close Session
Pass-Criteria	The value of Source LocURI in the SyncHdr sent by the client MUST be equal to the value sent in ./DevInfo/DevId Step 2: The client MUST send a setup request as follows: 1. Setup-Request shall contain a Replace tag, and this tag contains a CmdID tag and elements from the ./DevInfo node. These represent Device Information. 2. The Value in the SyncHdr/Source/LocURI tag should be equal to the value in the DevInfo/DevID tag, the latter being a subelement of the Replace tag.

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The client is triggered to initiate communication with the server.
2	→		Setup-Request	The client sends a Setup-Request.
3	←		Setup-Response	Test Tool sends OK response to the client to close Session

5.2 Device Management Client Conformance Test Group #2

5.2.1 DeviceManagement-v1.2-client-con-0201

Test Case Id	DeviceManagement-v1.2-client-con-0201
Test Object	Client device
Test Case Description	To check if the Test Object can switch the authentication scheme based on the challenge (MD5).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Specification Reference	[DMSEC] Chapter 5.3 [DMREPU] Chapter 6.1.6 [DMPRO] Chapter 9
SCR Reference	DM-SEC-C-001 Client must authenticate itself to a server DM-SEC-C-005 Send credentials to server DM-SEC-C-008 Support for OMA DM syncml:auth-md5 type authentication
Test Tool	DM 1.2 conformance test tool
Preconditions	The client must support md5 authentication
Test Procedure	1. The client is triggered to initiate communication with the server. 2. The client sends a Setup-Request. 3. Test Tool receives the request and responds to this by sending a Challenge (Chal). In the challenge the server specifies that it is expecting an md5 authentication in the next request and server also specifies Nonce to be used. 4. Client resends login response this time sending its credentials. 5. Test Tool sends OK response to the client with 212, telling the client that it is authenticated or otherwise if client credentials are not correct
Pass-Criteria	The Test Object MUST update its authentication scheme and send credentials using MD5 in the next session. Step 4: The Test Object MUST update its authentication scheme and send credentials using MD5 in the next session. This implies: 1. The test object sends its credentials as part of the <Cred> tag and using MD5 as the digest schema. 2. The credentials as sent by the test object must be the same as those saved on the server thus confirming that the test object has indeed carried out the md5 authentication correctly.

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The client is triggered to initiate communication with the server.
2	→		Setup-Request	The client sends a Setup-Request.
3	←		Setup-Response +	Test Tool receives the request and responds to this by sending

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
			Challenge	a Challenge (Chal). In the challenge the server specifies that it is expecting an md5 authentication in the next request and server also specifies Nonce to be used.
4	→		Authentication using MD5	Client resends login response this time sending its credentials.
5	←		Login response	Test Tool sends OK response to the client with 212, telling the client that it is authenticated or otherwise if client credentials are not correct

5.3 Device Management Client Conformance Test Group #3

5.3.1 DeviceManagement-v1.2-client-con-0301

Test Case Id	DeviceManagement-v1.2-client-con-0301
Test Object	Client device
Test Case Description	To check if the Test Object supports the MD5 Digest authentication scheme.
Specification Reference	[DMSEC] Chapter 5.3 [DMREPU] Chapter 6.1.6 [DMPRO] Chapter 9
SCR Reference	DM-SEC-C-001 Client must authenticate itself to a server DM-SEC-C-005 Send credentials to server DM-SEC-C-008 Support for OMA DM syncml:auth-md5 type authentication
Test Tool	DM 1.2 conformance test tool
Preconditions	1. The client sends a Setup-Request 2. Test Tool receives the request and responds to this by sending a Challenge (Chal). In the challenge the server specifies that it is expecting an md5 authentication in the next request and server also specifies Nonce to be used.
Test Procedure	1. The client is triggered to initiate communication with the server. 2. The client sends a Setup-Request. 3. Credentials are correct, the server sends back the response confirming that the client has been successfully authenticated.
Pass-Criteria	The Test Object MUST send valid credentials encoded using the MD5 Digest authentication scheme. Step 2: 1. The test object sends its credentials as part of the <Cred> tag and using MD5 as the digest schema. 2. The credentials as sent by the test object must be the same as those

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

		saved on the server thus confirming that the test object has indeed carried out the md5 authentication correctly.

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The client is triggered to initiate communication with the server.
2	→		Setup-Request	The client sends a Setup-Request.
3	←		Login response	Credentials are correct, the server sends back the response confirming that the client has been successfully authenticated

5.3.2 DeviceManagement-v1.2-client-con-0302

Test Case Id	DeviceManagement-v1.2-client-con-0302
Test Object	Client device
Test Case Description	To check if the Test Object responds with a Results for a Get on the Root node.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMREPU] Chapter 6.6.12
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-C-010 Support for sending 'Results'
Test Tool	DM 1.2 Conformance test tool
Preconditions	Test tool should have ACL access rights for Get on the Root node.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the Root node ('.') in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the Get with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST respond with a Results containing at least the following element: DevInfo, DevDetail. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b. a 'Data' tag set to "200"

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	the message's 'SyncBody' SHALL contain a 'Results' tag with a 'Data' tag containing at least the following node names separated with a / a. DevInfo b. DevDetail

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the Root node ('.') in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.3.3 DeviceManagement-v1.2-client-con-0303

Test Case Id	DeviceManagement-v1.2-client-con-0303
Test Object	Client device
Test Case Description	To check if the Test Object responds with a Results for a Get on a leaf node.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMREPU] Chapter 6.6.12.
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-C-010 Support for sending 'Results'
Test Tool	DM 1.2 Conformance test tool
Preconditions	Test tool should have ACL access rights for Get on the leaf node. Client is not involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on a leaf node (e.g. './DevInfo/Lang') in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the Get with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST respond with a Results. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b. a 'Data' tag set to "200" the message's 'SyncBody' SHALL contain a 'Results' tag with a 'Data' tag containing the content of the node

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on a leaf node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.3.4 DeviceManagement-v1.2-client-con-0304

Test Case Id	DeviceManagement-v1.2-client-con-0304
Test Object	Client device
Test Case Description	To check if the Test Object responds correctly for a Get on a non-existant node.
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get'
Test Tool	DM 1.2 Conformance test tool
Preconditions	Client is not involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on a Non Existant Node (e.g.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	'./NonExistanrNode') in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the Get with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 404 status code on the Get. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b. a 'Data' tag set to "404"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on a Non Existant Node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.4 Device Management Client Conformance Test Group #4

5.4.1 DeviceManagement-v1.2-client-con-0401

Test Case Id	DeviceManagement-v1.2-client-con-0401
Test Object	Client device
Test Case Description	To check if the Test Object uses HMAC scheme.
Specification Reference	[DMSEC] Chapter 5.4
SCR Reference	DM-SEC-C-010 Integrity checking using HMAC-MD5 DM-SEC-C-011 Inserting HMAC in transport

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	DM-SEC-C-012 Using HMAC for all subsequent messages
Test Tool	DM 1.2 Conformance test tool
Preconditions	The client should support HMAC and use an insecure transport. Client is not involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with "407" code and a 'Chal' to the client of type 'auth-MAC' in the SyncBody of the Setup-Response (step 3 of the macro) as well as a 'NextNonce' tag set to b64 of ixit_NextNonce. 2. Client responds to the challenge by sending the requested credentials in the transport header. 3. Test tool checks the response and compares the credentials to those saved on the server and sends a 212 (Authenticated) or 401 (Unauthorized) response to the client, together with a 'Get' command on a test node (eg './DevInfo') 4. Client responds to the 'Get' with a valid package #3 message (see C.4) with the credentials as before in the transport header. 5. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session
Pass-Criteria	The Test Object MUST send valid HMAC. Step 2: The client MUST send a valid package #3 (see C.4) as follows: a. The transport header must contain the following: x-syncml-hmac: algorithm=MD5, username="", mac= where: 1. Algorithm is set to MD5 2. Username is the client's username (ixit_UserName) 3. mac is the digest computed as defined in the spec. a. The credentials sent by the client must be the valid according to ixit_UserName, ixit_UserPass, ixit_NextNonce Step 4: The client MUST send a valid package #3 (see C.4) as follows: a. The transport header must contain the following: x-syncml-hmac: algorithm=MD5, username="", mac=

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	where: 1. Algorithm is set to MD5 2. Username is the client's username (ixit_UserName) 3. mac is the digest computed as defined in the spec. a. The credentials sent by the client must be the valid according to ixit_UserName, ixit_UserPass, ixit_NextNonce b. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b a 'Data' tag set to "200". c. the message's 'SyncBody' SHALL contain a 'Results' tag with: a 'Data' tag containing Data stored in the node

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with "407" code and a 'Chal' to the client of type 'auth-MAC' in the SyncBody of the Setup-Response (step 3 of the macro) as well as a 'NextNonce' tag set to b64 of ixit_NextNonce.
2	→		Client Response with HMAC Auth Header	Client responds to the challenge by sending the requested credentials in the transport header.
4	←		Server Management Operations Message	Test tool checks the response and compares the credentials to those saved on the server and sends a 212 (Authenticated) or 401 (Unauthorized) response to the client.

5.5 Device Management Client Conformance Test Group #5

5.5.1 DeviceManagement-v1.2-client-con-0501

Test Case Id	DeviceManagement-v1.2-client-con-0501
Test Object	Client device
Test Case Description	To check if a interior node can be Added to a client.
Specification Reference	[DMREPU] Chapter 6.6.1
SCR Reference	DMREPPRO-PCE-C-003 Support for receiving 'Add'

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Tool	DM 1.2 Confomance Test tool
Preconditions	The node MUST not exist on the device
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command for an interior test node in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the 'Add' with a valid package #3 message (see C.4). 4. Test tool checks the response and if it is a 200 (OK) response it sends a 'Get' command on the interior node just created. If the response is '405' (Command Not Allowed) it sends a valid package #4 message (see C.5) to close the session. 5. Client responds to the Get command with a valid package #3 message (see C.4). 6. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return either a 200 or 405 status code.If the status code is 200,the new interior node MUST exist. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Add' used b. a 'Data' tag set to "200" or "405" Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b. a 'Data' tag set to "200"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command for an interior test node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the with a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test tool checks the response and if it is a 200 (OK) response it sends a 'Get' command on the interior node just created. If the response is '405' (Command Not Allowed) it sends a valid package #4 message (see C.5) to close the session.
5	→		Client Response	Client responds to the Get command with a valid package #3 message (see C.4).
6	←		Server Response	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.5.2 DeviceManagement-v1.2-client-con-0502

Test Case Id	DeviceManagement-v1.2-client-con-0502
Test Object	Client device
Test Case Description	To check if a leaf node can be Added to a client.
Specification Reference	[DMREPU] Chapter 6.6.1
SCR Reference	DMREPPRO-PCE-C-003 Support for receiving 'Add'
Test Tool	DM 1.2 Conformance test-tool
Preconditions	Client is not involved in a session with the server Test tool has the requires ACL rights and Accesstype privileges to add interior and leaf node on the client
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on a test interior node (e.g. The one created in 501) in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the 'Add' with a valid package #3 message (see C.4). 4. Test tool checks the response.and responds with an 'Add' command on a leaf node under the interior node on which the previous 'Add' was carried out.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	5. Client responds to the 'Add' with a valid package #3 message (see C.4). 6. Test tool checks the response and responds with a 'Get' command on the newly added leaf node. 7. Client responds to the 'Get' with a valid package #3 message (see C.4). 8. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 200 status code and the new leaf node MUST exist. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Add' used b. a 'Data' tag set to "200" or "418" Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Add' used b. a 'Data' tag set to "200" Step 7: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain a 'Results' tag with a 'Data' tag containing the content of the node. This Data should equal the Data that was added using the Add command on the leaf node

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on a test interior node (e.g. The one created in 501) in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the Add with a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test tool checks the response.and responds with an 'Add' command on a leaf node under the interior node on which the previous 'Add' was carried out.
5	→		Client Response	Client responds to the Add with a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test tool checks the response and responds with a 'Get' command on the newly added leaf node
7	→		Client Response	Client responds to the 'Get' with a valid package #3 message (see C.4).
8	←		Server Response	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.5.3 DeviceManagement-v1.2-client-con-0503

Test Case Id	DeviceManagement-v1.2-client-con-0503
Test Object	Client device
Test Case Description	To check if the Test Object returns a status code of 418 (Already Exists) for a Add on a existing leaf node.
Specification Reference	[DMREPU] Chapter 6.6.1
SCR Reference	DMREPPRO-PCE-C-003 Support for receiving 'Add'
Test Tool	DM 1.2 Conformance test tool
Preconditions	Test case 502 should have passed with a 200 status code. Client is not involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro'

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	(see E.1) with an 'Add' command on the leaf node created by testcase 502 in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the Add with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 418 status code. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Add' used b. a 'Data' tag set to "418"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the leaf node created by testcase 502 in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the Add with a valid package #3 message (see C.4).
4	←		Server Response	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.6 Device Management Client Conformance Test Group #6

5.6.1 DeviceManagement-v1.2-client-con-0601

Test Case Id	DeviceManagement-v1.2-client-con-0601
Test Object	Client device
Test Case Description	To check if the Test Object handles a Replace.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Specification Reference	[DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace'
Test Tool	DM 1.2 Confomance Test tool
Preconditions	The client must not be involved in a session with the server The client must allow adding of nodes
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Replace' command for a test leaf node in the Setup-Response (step 3 of the macro) to change the data to some value x. 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the 'Replace' with a valid package #3 message (see C.4). 4. Test tool checks the response and if it is a 200 (OK) response go to Step 6. If the response is '425' (Permission Denied) then go to step 4a. Otherwise if the response code is 404 then go to Step 5. 4a. Test tool sends a Replace command on the ACL rights of the test leaf node so that 'Replace' is allowed (eg Replace=*) 4b. Client responds to the 'Replace' with a valid package #3 message (see C.4). 4c. Test tool checks the response and sends a 'Replace' command on the Data of the leaf node to change the data to some value x 4d. Client responds to the 'Replace' with a valid package #3 message (see C.4). Go to Step 6. 5. Test tool sends an 'Add' command to the client with the path of an interior test node. 5a. Client responds to the 'Add' with a valid package #3 message (see C.4). 5b. Test tool checks the response and sends an 'Add' command on a leaf node under the interior node created in Step 5. 5c. Client responds to the 'Add' with a valid package #3 message (see C.4). 5d. Test tool checks the response and sends a 'Replace' command on the ACL rights of the newly created leaf node such that Replace is enabled (eg Replace=*) 5e. Client responds to the 'Replace' with a valid package #3 message (see C.4). 5f. Test tool checks the response and sends a 'Replace' command on the Data of the leaf node to change the data to some value x. 5g. Client responds to the 'Replace' with a valid package #3 message (see C.4).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	6. Test tool checks the response and sends a 'Get' command on the leaf node whose data has been replaced. 7. Client responds to the 'Get' with a valid package #3 message (see C.4). 8. Test Tool checks the response and compares the value of the node to the value x. It sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 200 status code. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Replace' used b. a 'Data' tag set to "200", "404" or "425" Step 4b/5e: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Replace' used with '?prop=ACL' at the end b. a 'Data' tag set to "200", Step 5a/5c: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Add' used b. a 'Data' tag set to "200" Step 4d/5g: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Replace' used b. a 'Data' tag set to "200". Step 7:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b. a 'Data' tag set to "200". 2. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing the Data stored inside the node which must be equal to x.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Replace' command for a test leaf node in the Setup-Response (step 3 of the macro) to change the data to some value x
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the 'Replace' with a valid package #3 message (see C.4).
4			Server Checks Data received	Test tool checks the response and if it is a 200 (OK) response go to Step 6. If the response is '425' (Permission Denied) then go to step 4a. Otherwise if the response code is 404 then go to Step 5.
4a	←		Server Management Operations Message	Test tool sends a Replace command on the ACL rights of the test leaf node so that 'Replace' is allowed (eg Replace=*)
4b	→		Client Response	Client responds to the 'Replace' with a valid package #3 message (see C.4).
4c	←		Server Management Operations Message	Test tool checks the response and sends a 'Replace' command on the Data of the leaf node to change the data to some value x
4d	→		Client Response	Client responds to the 'Replace' with a valid package #3 message (see C.4). Go to Step 6.
5	←		Server Management Operations Message	Test tool sends an 'Add' command to the client with the path of an interior test node.
5a	→		Client Response	Client responds to the 'Add' with a valid package #3 message (see C.4).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
5b	←		Server Management Operations Message	Test tool checks the response and sends an 'Add' command on a leaf node under the interior node created in Step 5.
5c	→		Client Response	Client responds to the 'Add' with a valid package #3 message (see C.4).
5d	←		Server Management Operations Message	Test tool checks the response and sends a 'Replace' command on the ACL rights of the newly created leaf node such that Replace is enabled (eg Replace=*)
5e	→		Client Response	Client responds to the 'Replace' with a valid package #3 message (see C.4).
5f	←		Server Management Operations Message	Test tool checks the response and sends a 'Replace' command on the Data of the leaf node to change the data to some value x.
5g	→		Client Response	Client responds to the 'Replace' with a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test tool checks the response and sends a 'Get' command on the leaf node whose data has been replaced
7	→		Client Response	Client responds to the 'Get' with a valid package #3 message (see C.4).
8	←		Server Response	Test Tool checks the response and compares the value of the node to the value x. It sends a valid package #4 message (see C.5) to close the session.

5.6.2 DeviceManagement-v1.2-client-con-0602

Test Case Id	DeviceManagement-v1.2-client-con-0602
Test Object	Client device
Test Case Description	To check if the Test Object rejects a Replace on a non-existant node.
Specification Reference	[DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace'

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Tool	DM 1.2 Conformance test-tool
Preconditions	Client is not involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Replace' command on a non existent node (e.g. './NonExistantNode') in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the 'Replace' with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 404 status code. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Replace' used b. a 'Data' tag set to "404"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Replace' command on a non existent node (e.g. './NonExistantNode') in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the 'Replace' with a valid package #3 message (see C.4).
4	←		Server Response	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.7 Device Management Client Conformance Test Group #7

5.7.1 DeviceManagement-v1.2-client-con-0701

Test Case Id	DeviceManagement-v1.2-client-con-0701
Test Object	Client device
Test Case Description	To check if the Test Object handles the Sequence command correctly.
Specification Reference	[DMREPU] Chapter 6.6.14
SCR Reference	DMREPPRO-PCE-C-009 Support for receiving 'Sequence'
Test Tool	DM 1.2 Conformance test tool
Preconditions	Client is not involved in a session with the server The Test tool must have the Accesstype privileges to carry out Add and Get on the test interior node.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Add' command on the interior test node to check the existence of the same 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the 'Add' command with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a 'Sequence' containing 'Add' and 'Get' in this order on a test leaf node under the interior node. 5. Client responds to the 'Sequence' command with a valid package #3 message (see C.4). 6. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST behave according to the rules specified for Sequence. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'Cmd' tag set to 'Add' and a 'CmdRef' tag set to the value of 'CmdID' which the 'Add' used b. a 'Data' tag set to "200" or 415 Step 5:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'Cmd' tag set to 'Sequence' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain also a 'Status' tag with: a 'Cmd' tags set to each of the commands performed within the 'Sequence' command, these being 'Add' and 'Get' in the same sequence. b 'TargetRef' if present should be equal to the path of the node on which the operations are carried out c Each of the above should have a 'Data' tag set to "200". 3. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing the Data stored inside the node which must be equal to the data stored using the 'Add' command within the 'Sequence' command. b.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Add' command on the interior test node to check the existence of the same
2				If required by the client: 'DM Authentication Macro'
3		→	Client Response	Client responds to the 'Add' command with a valid package #3 message (see C.4).
4		←	Server Response	Test tool checks the response and sends a 'Sequence' containing 'Add' and 'Get' in this order on a test leaf node under the interior node whose rights have been changed in Step 1.
5		→	Client Response	Client responds to the 'Sequence' command with a valid

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
				package #3 message (see C.4).
6	←		Server Response	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.8 Device Management Client Conformance Test Group #8

5.8.1 DeviceManagement-v1.2-client-con-0801

Test Case Id	DeviceManagement-v1.2-client-con-0801
Test Object	Client device
Test Case Description	To check if the Test Object implements Confirmation User Interaction Alert command.
Specification Reference	DM-PRO-UI-C-002 Executing Confirm or Reject Alert (ics_support_confirm_reject_alert)
SCR Reference	DM-PRO-UI-C-002 Executing Confirm or Reject Alert
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Alert' command ('Confirmation Alert' with 'Data' = "1101") with optional parameters as first 'Item' (like the Minimum Display Time "MINDT=20") and, as second item, the following text to be displayed to the user: "Press 'Yes' to accept the Confirmation User Interaction Alert command" in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. The 'Confirmation Alert' text is displayed on the UE and the user accepts the command 4. Client sends a valid package #3 message (see C.4). 5. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The test object must accept the change and send a status of 200 status code on the Alert. Step 4: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'Cmd' tag set to "Alert" b. a 'Data' tag set to "200"

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Alert' command ('Confirmation Alert' with 'Data' = "1101") with optional parameters as first 'Item' (like the Minimum Display Time "MINDT=20") and, as second item, the following text to be displayed to the user: "Press 'Yes' to accept the Confirmation User Interaction Alert command" in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3				The 'Confirmation Alert' text is displayed on the UE and the user accepts the command.
4	→		Client-Response	Client sends a valid package #3 message (see C.4).
5	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.8.2 DeviceManagement-v1.2-client-con-0802

Test Case Id	DeviceManagement-v1.2-client-con-0802
Test Object	Client device
Test Case Description	To check if the Test Object implements Confirmation User Interaction Alert command.
Specification Reference	[DMPRO] Chapter 10.2
SCR Reference	DM-PRO-UI-C-002 Executing Confirm or Reject Alert (ics_support_confirm_reject_alert)
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Alert' command ('Confirmation Alert' with 'Data' = "1101") with optional parameters as first 'Item' (like the Minimum Display Time "MINDT=20") and, as second item, the following text to be displayed to the user: "Press 'No' to reject the Confirmation User Interaction Alert command" in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. The 'Confirmation Alert' text is displayed on the UE and the user rejects the command. 4. Client sends a valid package #3 message (see C.4). 5. Test Tool sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Pass-Criteria	The test object must reject the change and send a status of 304 status code on the Alert. Step 4: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'Cmd' tag set to "Alert" b. a 'Data' tag set to "304"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Alert' command ('Confirmation Alert' with 'Data' = "1101") with optional parameters as first 'Item' (like the Minimum Display Time "MINDT=20") and, as second item, the following text to be displayed to the user: "Press 'No' to reject the Confirmation User Interaction Alert command" in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3				The 'Confirmation Alert' text is displayed on the UE and the user rejects the command.
4	→		Client-Response	Client sends a valid package #3 message (see C.4).
5	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.9 Device Management Client Conformance Test Group #9

5.9.1 DeviceManagement-v1.2-client-con-0901

Test Case Id	DeviceManagement-v1.2-client-con-0901
Test Object	Client device
Test Case Description	To check if the Test Object deletes a interior node correctly.
Specification Reference	[DMREPU] Chapter 6.6.5
SCR Reference	DMREPPRO-PCE-C-006 Support for receiving 'Delete'
Test Tool	DM 1.2 conformance test tool
Preconditions	There should be interior node configured under test node. The client is not

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on the interior test node in the Setup-Response (step 3 of the macro) in order to test if the interior node exists. 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. If, at step 3, the Result of the Get is 404 'Not Found', Test tool starts the 'DM Node Creation Macro' (see E.3) with the interior test node as input. 5. Test Tool sends a valid package #4 message (see C.5) with a Delete command on the interior test node. 6. Client sends a valid package #3 message (see C.4). Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return either a 200/405 status code. Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Delete" used b. a 'Data' tag set to "200" or "405"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on the interior test node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4				If, at step 3, the Result of the Get is 404 'Not Found', Test tool starts the 'DM Node Creation Macro' (see E.3) with the interior test node as input.
5	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a Delete command on the interior test node.
6	→		Client-Response	Client sends a valid package #3 message (see C.4).
7	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.9.2 DeviceManagement-v1.2-client-con-0902

Test Case Id	DeviceManagement-v1.2-client-con-0902
Test Object	Client device
Test Case Description	To check if the Test Object sends a 404 status code for a Delete on a non-existant node.
Specification Reference	[DMREPU] Chapter 6.6.5
SCR Reference	DMREPPRO-PCE-C-006 Support for receiving 'Delete'
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on a non-existant node (like './NonExistantNode') in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). If the Result of the Get is not 404 'Not Found', the verdict of the test is INCONCLUSIVE and the next executed step is the step 6. 4. Test Tool sends a valid package #4 message (see C.5) with a Delete command on a non-existant node (like './NonExistantNode'). 5. Client sends a valid package #3 message (see C.4). 6. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 404 status code. Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Delete" used b. a 'Data' tag set to "404"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on a non-existant node (like './NonExistantNode') in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4). If the Result of the Get is

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
				not 404 'Not Found', the verdict of the test is INCONCLUSIVE and the next executed step is the step 6.
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a Delete command on a non-existant node (like './NonExistantNode').
5	→		Client-Response	Client sends a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.9.3 DeviceManagement-v1.2-client-con-0903

Test Case Id	DeviceManagement-v1.2-client-con-0903
Test Object	Client device
Test Case Description	To check if the Test Object rejects a delete on the Permanent node ./DevInfo
Specification Reference	[DMREPU] Chapter 6.6.5
SCR Reference	DMREPPRO-PCE-C-006 Support for receiving 'Delete'
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on './DevInfo' node in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). If the Result of the Get is not 200 'OK', the verdict of the test is INCONCLUSIVE and the next executed step is the step 6. 4. Test Tool sends a valid package #4 message (see C.5) with a Delete command on the permanent node './DevInfo'. 5. Client sends a valid package #3 message (see C.4). 6. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 405 status code. Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Delete" used

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	b. a ‘Data’ tag set to “405”

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a ‘DM Session Initialisation Macro’ (see E.1) with a Get command on the permanent node './DevInfo' in the Setup-Response (step 3 of the macro).
2				If required by the client: ‘DM Authentication Macro’.
3	→		Client-Response	Client sends a valid package #3 message (see C.4). If the Result of the Get is not 200 ‘OK’, the verdict of the test is INCONCLUSIVE and the next executed step is the step 6.
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a Delete command on the permanent node ‘./DevInfo’.
5	→		Client-Response	Client sends a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.10 Device Management Client Conformance Test Group #10

5.10.1 DeviceManagement-v1.2-client-con-1001

Test Case Id	DeviceManagement-v1.2-client-con-1001
Test Object	Client device
Test Case Description	To check if the Test Object can handle multiple messages.
Specification Reference	[SYNCMETA]
SCR Reference	DSDM-METINF-S-009 Support for MaxMsgSize element
Test Tool	DM 1.2 Conformance test-tool
Preconditions	None.
Test Procedure	1. The test procedure starts with a ‘DM Session Initialisation Macro’ (see E.1) with a set of multiple Get commands on a particular node of choice (eg './DevInfo' node) in the Setup-Response (step 3 of the macro). 2. If required by the client: ‘DM Authentication Macro’. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) with an ‘Alert’ command consisting of 1222 as Data. 5. Client sends a valid package #3 message (see C.4).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	6. Steps 4 and 5 are repeated for as long as the client does not send the 'Final' tag in the response. 7. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The session MUST complete successfully. The Test Object MUST respond with a Results. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain several 'Status' tags with: a. a 'CmdRef' tag set to the value of 'CmdID' which the 'Get' used b. a 'Data' tag set to "200" the message's 'SyncBody' SHALL contain several 'Results' tags with a 'Data' tag containing the content of the node c. The response should not contain a 'Final' tag Step 5: The client MUST send a valid package #3 (see C.4) as follows: 2. the message's 'SyncBody' SHALL contain several 'Status' tags with: a. a 'CmdRef' tag set to the value of 'CmdID'which the 'Get' used b. a 'Data' tag set to "200" the message's 'SyncBody' SHALL contain several 'Results' tags with a 'Data' tag containing the content of the node c. If the results to all the 'Get' commands are cumulatively available through these two responses then the Client should send a 'Final' tag in the response. Otherwise same as for Step 3 above

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a set of multiple Get commands on a particular node of choice (eg './DevInfo' node) in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with an 'Alert' command consisting of 1222 as Data.
5	→		Client-Response	Client sends a valid package #3 message (see C.4).
6			Server Management Operations Message	Steps 4 and 5 are repeated for as long as the client does not send the 'Final' tag in the response.
7	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.11 Device Management Client Conformance Test Group #11

5.11.1 DeviceManagement-v1.2-client-con-1101

Test Case Id	DeviceManagement-v1.2-client-con-1101
Test Object	Client device
Test Case Description	To check if the Test Object handles the Atomic command correctly.
Specification Reference	[DMREPU] Chapter 6.6.3
SCR Reference	DMREPPRO-PCE-C-004 Support for receiving 'Atomic'
Test Tool	DM 1.2 Conformance test-tool
Preconditions	Client supports Atomic command Client is not involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. Test tool checks the response. If the response is '200' (OK) then go to step 5 else if the result is '418' (already exists) go to step 4a. 4a. Test tool sends a valid package #4 message (see C.5) with a 'Delete' command on the test leaf node under the interior test node 4b. Client sends a valid package#3 message (see C.4). 5. Test tool checks the response and sends a valid package #4 message (see C.5) with an 'Atomic' command consisting of the following commands: (1) An 'Add' command on the test leaf node, (2) An 'Alert' command ('Confirmation Alert' with 'Data' = "1101") with optional parameters as first 'Item' (like the Minimum Display Time "MINDT=20") and, as second item, the following text to be

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	displayed to the user: “Press 'Yes' to accept the Confirmation User Interaction Alert command” and (3) A ‘Replace’ command on the test leaf node. 6. Client sends a valid package #3 message (see C.4). 7. Test tool checks the response and sends a valid package #4 message (see C.5) with a ‘Get’ command on the test leaf node. 8. Client sends a valid package #3 message (see C.4). 9. Test tool checks the response and sends a valid package #4 message (see C.5) with an ‘Atomic’ command consisting of the following commands: (1) A ‘Replace’ command on the test leaf node, (2) A ‘Delete’ command on the test leaf node and (3) A ‘Get’ command on the test leaf node. 10. Client sends a valid package #3 message (see C.4). 11. Test tool checks the response. If the response is “216” (Atomic roll back ok) for the commands ‘Replace’ and ‘Delete’ go to step 11a, else go to step 12. 11a. Test tool sends a valid package #4 message (see C.5) with a ‘Get’ command on the test leaf node. 11b. Client sends a valid package#3 message (see C.4). Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 200 status code on the Atomic. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message’s ‘SyncBody’ SHALL contain at least an ‘Add’ tag with: a. a ‘TargetRef’ tag set to the node on which the ‘Add’ was carried out b. a ‘Data’ tag set to “200” or “418”. Step 4b (If the response is ‘418’ at step 3): The client MUST send a valid package #3 (see C.4) as follows: 1. the message’s ‘SyncBody’ SHALL contain at least a ‘Delete’ tag with: a. a ‘TargetRef’ tag set to the node on which the ‘Delete’ was carried out b. a ‘Data’ tag set to “200” or “404” (Not Found). Step 6: The client MUST send a valid package #3 (see C.4) as follows: 1. the message’s ‘SyncBody’ SHALL contain at least an ‘Atomic’ tag with:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	a. a ‘Data’ tag set to “200”. 2. the message’s ‘SyncBody’ SHALL contain at least an ‘Add’ tag with: a. a ‘TargetRef’ tag set to the node on which the ‘Add’ was carried out b. a ‘Data’ tag set to “200”. 3. the message’s ‘SyncBody’ SHALL contain at least an ‘Alert’ tag with: a. a ‘Data’ tag set to “200”. 4. the message’s ‘SyncBody’ SHALL contain at least a ‘Replace’ tag with: a. a ‘TargetRef’ tag set to the node on which the ‘Replace’ was carried out b. a ‘Data’ tag set to “200”. 8: The client MUST send a valid package #3 (see C.4) as follows: 1. the message’s ‘SyncBody’ SHALL contain at least a ‘Get’ tag with: a. a ‘TargetRef’ tag set to the node on which the ‘Get’ was carried out b. a ‘Data’ tag set to “200”. c. a ‘Results’ tag with a ‘CmdRef’ equivalent to the ‘CmdID’ of the ‘Get’ command and a’Data’ tag containing the data retrieved from the node. This data should be equivalent to the data saved in the node using the ‘Replace’ command inside the Atomic. Step 10: The client MUST send a valid package #3 (see C.4) as follows: 1. the message’s ‘SyncBody’ SHALL contain at least an ‘Atomic’ tag with: a. a ‘Data’ tag set to “500”. 2. the message’s ‘SyncBody’ SHALL contain at least a ‘Replace’ tag with:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	a. a 'TargetRef' tag set to the node on which the 'Replace' was carried out b. a 'Data' tag set to "216" or "516" or "500". 3. the message's 'SyncBody' SHALL contain at least a 'Delete' tag with: a. a 'TargetRef' tag set to the node on which the 'Delete' was carried out b. a 'Data' tag set to "216" or "516" or "500". 4. the message's 'SyncBody' SHALL contain at least a 'Get' tag with: a. a 'TargetRef' tag set to the node on which the 'Get' was carried out b. a 'Data' tag set to "500". Step 11b: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Get' tag with: a. a 'TargetRef' tag set to the node on which the 'Get' was carried out b. a 'Data' tag set to "200". c. a 'Results' tag with a 'CmdRef' equivalent to the 'CmdID' of the 'Get' command and a'Data' tag containing the data retrieved from the node. This data should be equivalent to the data saved in the node using the 'Replace' command inside the first Atomic.

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4				Test tool checks the response. If the response is '200' (OK) then go to step 5 else if the result is '418' (already exists) go to step 4a.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
4a	←		Server Management Operations Message	Test tool sends a valid package #4 message (see C.5) with a 'Delete' command on the test leaf node under the interior test node
4b	→		Client-Response	Client sends a valid package #3 message (see C.4).
5	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) with an 'Atomic' command consisting of the following commands: (1) An 'Add' command on the test leaf node, (2) An 'Alert' command with text requesting the user to press 'yes' and (3) A 'Replace' command on the test leaf node.
6	→		Client-Response	Client sends a valid package #3 message (see C.4).
7	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) with a 'Get' command on the test leaf node
8	→		Client-Response	Client sends a valid package #3 message (see C.4).
9	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) with an 'Atomic' command consisting of the following commands: (1) A 'Replace' command on the test leaf node, (2) A 'Delete' command on the test leaf node and (3) A 'Get' command on the test leaf node.
10	→		Client-Response	Client sends a valid package #3 message (see C.4).
11a	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) with a 'Get' command on the test leaf node.
11b	→		Client-Response	Client sends a valid package #3 message (see C.4).
12	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.12 Device Management Client Conformance Test Group #12

5.12.1 DeviceManagement-v1.2-client-con-1201

Test Case Id	DeviceManagement-v1.2-client-con-1201
Test Object	Client device
Test Case Description	To check if the structure of the ./DevInfo standard object is correct.
Specification Reference	[DMSTDOBJ] Chapter 5.3.2

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	[DMREPU] Chapter 6.6.12
SCR Reference	SCR-DM-STDOBJ-C-001 Support of DevInfo object
Test Tool	DM 1.2 Conformance test tool
Preconditions	The client must not be involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the './DevInfo' node in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the Get with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Results returned by the Test Object MUST contain the names of all the mandatory nodes under /DevInfo seperated by / Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing at least the following node names separated with a /: 1. DevId 2. Man 3. Mod 4. DmV 5. Lang 6. But could OPTIONALLY also contain the following nodes Ext and Bearer, which would in turn contain further subnodes.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on the './DevInfo' node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
3	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.12.2 DeviceManagement-v1.2-client-con-1202

Test Case Id	DeviceManagement-v1.2-client-con-1202
Test Object	Client device
Test Case Description	To check if the structure of the ./DevDetail standard object is correct.
Specification Reference	[DMSTDOBJ] Chapter 5.3.3 [DMREPU] Chapter 6.6.12
SCR Reference	SCR-DM-STDOBJ-C-002 Support of DevDetail Object
Test Tool	DM 1.2 Conformance test tool
Preconditions	The client must not be involved in a session with the server
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the './DevDetail' node in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the Get with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4 message (see C.5) with a 'Get' command on the node './DevDetail/URI' 5. Client responds to the Get with a valid package #3 message (see C.4). 6. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Results returned by the Test Object MUST contain the names of all the mandatory nodes under ./DevDetail separated by / Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200"

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing at least the following node names separated with a /: 1. URI 2. DevTyp 3. OEM 4. FwV 5. SwV 6. HwV 7. LrgObj 8. But could OPTIONALLY also contain the following nodes Ext and Bearer, which would in turn contain further subnodes. Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing at least the following node names separated with a /: 1. MaxDepth 2. MaxTotLen 3. MaxSegLen

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on the './DevDetail' node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4	←		Server Management Operations	Test tool checks the response and sends a valid package #4 message (see C.5) with a 'Get' command on the node

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
			Message	'./DevDetail/URI'.
5	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.12.3 DeviceManagement-v1.2-client-con-1203

Test Case Id	DeviceManagement-v1.2-client-con-1203
Test Object	Client device
Test Case Description	To check if the structure of the DMAcc Management Object is correct.
Specification Reference	[DMSTDOBJ] Chapter 5.3.1
SCR Reference	[DMSTDOBJ] SCR-DM-STDOBJ-C-003 Support of DM Account Object
Test Tool	DM 1.2 Conformance test tool
Preconditions	The client must not be involved in a session with the server. 1. Client can submit to server a DDF or XML schema description of the expected node structure 2. Client can enter <Interior Node> location into the test tool 3. Server address, port number, authentication settings, and connectivity definitions provisioned into the client as applicable.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the ixit_PathToDMAccMO in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the Get with a valid package #3 message (see C.4) and with at least one <Interior Node> as Results. 4. For each <Interior Node> node at step 3 or for only the <Interior Node> specified as precondition: 5a. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>' node. 5b. Client responds to the Get with a valid package #3 message (see C.4). 5c. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr' node. 5d. Client responds to the Get with a valid package #3

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	message (see C.4) and with at least one <X> node as Results for the 'Get' on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr' node. 5e. For each <X> node at step 4.d: i. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>' node ii. Client responds to the Get with a valid package #3 message (see C.4) iii. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/AddrType' node and, if applicable, on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port' node. iv. Client responds to the Get with a valid package #3 message (see C.4) v. If applicable, for each <Y> node under 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port' node: 1. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port/<Y>' node 2. Client responds to the Get with a valid package #3 message (see C.4) 5f. If applicable: i. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAuth' node ii. Client responds to the Get with a valid package #3 message (see C.4) and with at least one <X> node as Results for the 'Get' on the 'ixit_PathToDMAccMO/<Interior Node>/AppAuth' node. iii. For each <X> node under 'ixit_PathToDMAccMO/<Interior Node>/AppAuth' node: 3. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAuth/<X>/AAuthLevel' node and a 'Get' command on the 'ixit_PathToDMAccMO/<Interior

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Node>/AppAuth/<X>/AAuthType' node. 4. Client responds to the 'Get' commands with a valid package #3 message (see C.4). 5g. If applicable: i. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/ToConRef' node ii. Client responds to the Get with a valid package #3 message (see C.4) and with at least one <X> node as Results for the 'Get' on the 'ixit_PathToDMAccMO/<Interior Node>/ ToConRef' node. iii. For each <X> node under 'ixit_PathToDMAccMO/<Interior Node>/ ToConRef' node: 5. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/ ToConRef/<X>' node 6. Client responds to the 'Get' command with a valid package #3 message (see C.4). 5h. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Results returned by the Test Object MUST contain the names of all the mandatory nodes under <Interior Node> separated by /. GET on <Interior Node> MUST also return appropriate optional nodes as defined in the DDF or XML Schema submitted by Client 1. GET on <Interior Node> MUST at least return: AppId / ServerId / AppAddr GET on <Interior Node> MAY also return the following node names as appropriate to submitted DDF Name / PrefConRef / ToConRef / AAuthPref / AppAuth / Ext 2. GET on <Interior Node>/AppAddr/<x>/ MUST at least return: Addr / AddrType GET on <Interior Node>/AppAddr/<x>/ MAY also return the following node names as appropriate to submitted DDF:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Port 3. If applicable, GET on <Interior Node>/AppAddr/<x>/Port/<x> MUST at least return: PortNbr 4. GET on <Interior Node>/ AppAuth /<x>/ MUST at least return : AAuthLevel/ AuthType GET on <Interior Node>/ AppAuth /<x>/ MAY also return the following node names as appropriate to submitted DDF: AAuthName / AAuthSecret / AAuthData 5. If applicable, GET on <Interior Node>/ ToConRef /<x> MUST at least return: ConRef object Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with ixit_PathToDMAccMO/ as value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to ixit_PathToDMAccMO/ b. a 'Data' tag set to a not empty value containing the names of at least one <Interior Node> node. Step 4.b: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>' (where <Interior Node> is the name of the interior node) as the value of 'Target'. b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing at least the following node names separated with a /:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	1. AppID 2. ServerID 3. AppAddr 4. But could OPTIONALLY also contain the following nodes Name, PrefConRef, ToConRef, AAuthPref, AppAuth and Ext, which would in turn contain further sub-nodes. Step 4.d: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' of the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAddr' as the value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/AppAddr' b. a 'Data' tag set to a not empty value containing the names of at least one <X> node. Step 4.e.ii: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>' as the value of 'Target'. b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing at least the following node names separated with a /: 1. Addr 2. AddrType 3. But could OPTIONALLY also contain the following node Port, which would in turn contain further sub-nodes. Step 4.e.iv: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	a. a 'CmdRef' tag set to value of 'CmdID' which "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/AddrType' as the value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/AddrType' b. a 'Data' tag set to an empty value or "URI" or "IPv4" or "IPv6". 3. If applicable, the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port' as the value of 'Target' b. a 'Data' tag set to "200" 4. If applicable, the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port' b. a 'Data' tag set to a not empty value containing the names of at least one <Y> node. If applicable, Step 4.e.v.2: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port/<Y>' as the value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port/<Y>' b. a 'Data' tag set to "PortNbr". If applicable, Step 4.f.ii: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAuth' as the value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/AppAuth' b. a 'Data' tag set to a not empty value containing the names of at least one <X> node. If applicable, Step 4.f.iii.2: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAuth/<X>/AAuthLevel' as the value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/AppAuth/<X>/AAuthLevel' b. a 'Data' tag set to "CLCRED" or "SRVCRED" or "OBEX" or "HTTP". 3. If applicable, the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/AppAuth/<X>/AAuthType' as tha value of 'Target' b. a 'Data' tag set to "200" 4. If applicable, the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/AppAuth/<X>/AAuthType' b. a 'Data' tag set to "HTTP-BASIC" or "HTTP-DIGEST" or "BASIC" or "DIGEST" or "HMAC" or "X509" or "SECUREID" or "SAFEWORD" or "DIGIPASS" or "TRANSPORT". If applicable, Step 4.g.ii: The client MUST send a valid package #3 (see C.4) as follows:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/ToConRef' as the value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/ToConRef' b. a 'Data' tag set to a not empty value containing the names of at least one <X> node. If applicable, Step 4.g.iii.2: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used with 'ixit_PathToDMAccMO/<Interior Node>/ToConRef/<X>' as the value of 'Target' b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an 'Item' tag with: a. a 'Source' tag with a 'LocURI' tag set to 'ixit_PathToDMAccMO/<Interior Node>/ ToConRef /<X>' b. a 'Data' tag set to "ConRef".

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the ixit_PathToDMAccMO in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4) and with at least one <Interior Node> as Results.
4a	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>' node.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
4b	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4c	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr' node.
4d	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4) and with at least one <X> node as Results for the 'Get' on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr' node.
4e.i	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>' node.
4e.ii	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4e.iii	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/AddrType' node and, if applicable, on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port' node.
4e.iv	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4e.v.1 (conditional)	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAddr/<X>/Port/<Y>' node.
4e.v.2 (conditional)	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4f.i (conditional)	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAuth' node.
4f.ii (conditional)	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4) and with at least one <X> node as Results for the 'Get' on the 'ixit_PathToDMAccMO/<Interior Node>/AppAuth' node.
4f.iii.1 (conditional)	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAuth/<X>/AAuthLevel' node and a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/AppAuth/<X>/AAuthType' node.
4f.iii.2 (conditional)	→		Client Response	Client responds to the Get with a valid package #3 message (see C.4).
4g.i (conditional)	←		Server Management Operations	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/ToConRef' node.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
			Message	
4f.iii.2 (conditional)		→	Client Response	Client responds to the Get with a valid package #3 message (see C.4) and with at least one <X> node as Results for the 'Get' on the 'ixit_PathToDMAccMO/<Interior Node>/ ToConRef' node.
4g.iii.1 (conditional)	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the 'ixit_PathToDMAccMO/<Interior Node>/ ToConRef/<X>' node.
4g.iii.2 (conditional)		→	Client Response	Client responds to the Get with a valid package #3 message (see C.4).
5	←		Server Management Operations Message	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.13 Device Management Client Conformance Test Group #13

5.13.1 DeviceManagement-v1.2-client-con-1301

Test Case Id	DeviceManagement-v1.2-client-con-1301
Test Object	Client device
Test Case Description	To check if the Root node has ACL.
Specification Reference	[DMTND] Chapter 7.7.1 [DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMTND-Prop-C-001 Support for the ACL property
Test Tool	DM 1.2 Conformance test-tool
Preconditions	The client is not involved in a session with the test tool
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on '.?prop=ACL' command in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	Response to Get on '.?prop=ACL' MUST be 200 and the Results must contain valid ACL . Step 3:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with: a. a 'Item' tag with a 'Data' tag set to a valid ACL

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on '.?prop=ACL' command in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.13.2 DeviceManagement-v1.2-client-con-1302

Test Case Id	DeviceManagement-v1.2-client-con-1302
Test Object	Client device
Test Case Description	To check if Test Object supports Get on Format property on the Root node.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 7.2
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMTND-Prop-C-002 Support for the Format property
Test Tool	DM 1.2 Conformance test-tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on '.?prop=Format' command in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	Status to Get on '.?prop=Format' MUST be 200 and the Results must contain the data 'node'. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with: a 'Item' tag with a 'Data' tag set to 'node'

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on '.?prop= Format' command in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.13.3 DeviceManagement-v1.2-client-con-1303

Test Case Id	DeviceManagement-v1.2-client-con-1303
Test Object	Client device
Test Case Description	To check if Test Object supports Get on the Type property on the Root node.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 7.2
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMTND-Prop-C-008 Support for the Type property
Test Tool	DM 1.2 Conformance test-tool

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on '.?prop=Type' command in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	Response to Get on '.?prop=Type' MUST be 200 and the Results must be null or point to DDF document. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with: a 'Item' tag with a 'Data' tag set to null or point to DDF document

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on '.?prop= Type' command in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.13.4 DeviceManagement-v1.2-client-con-1304

void

5.13.5 DeviceManagement-v1.2-client-con-1305

Test Case Id	DeviceManagement-v1.2-client-con-1305
Test Object	Client device

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Case Description	To check if the Test Object supports Get on the Name property on a Interior node ('./DevDetail/URI').
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMTND-Prop-C-003 Support for the Name property
Test Tool	DM 1.2 Conformance test-tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on './DevDetail/URI?prop=Name' command in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	Status to Get on '.DevDetail/URI?prop=Name' MUST be 200 and the Results must have the data as 'URI'. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with: a 'Item' tag with a 'Data' tag set 'URI'.

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on './DevDetail/URI?prop=Name' command in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.13.6 DeviceManagement-v1.2-client-con-1306

Test Case Id	DeviceManagement-v1.2-client-con-1306
Test Object	Client device
Test Case Description	To check if the Test Object supports Get on the Size property on a leaf node('./DevDetail/URI/MaxTotLen').
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMTND-Prop-C-004 Support for the Size property in leaf nodes
Test Tool	DM 1.2 Conformance test-tool
Preconditions	The client is not involved in a session with the test tool. ics_prop_size is set to true.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on './DevDetail/URI?prop=Size' command in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200"

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on './DevDetail/URI?prop=Size' command in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.13.7 DeviceManagement-v1.2-client-con-1307

Test Case Id	DeviceManagement-v1.2-client-con-1307
Test Object	Client device
Test Case Description	To check the behaviour of the Test Object for Replace on the Name property on a permanent node('./DevDetail').
Specification Reference	[DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace' DMTND-Prop-C-003 Support for the Name property
Test Tool	DM 1.2 Conformance test-tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Replace on './DevDetail?prop=Name' command in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	Status to Replace on '.DevDetail?prop=Name' MUST be 405. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used a 'Data' tag set to "405"

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get on './DevDetail/URI?prop= Name' command in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.13.8 DeviceManagement-v1.2-client-con-1308

Test Case Id	DeviceManagement-v1.2-client-con-1308
Test Object	Client device
Test Case Description	To check if the Test Object supports Replace on the ACL property on the interior test node.
Specification Reference	[DMTND] Chapter 7.7.1 [DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace' DMTND-Prop-C-001 Support for the ACL property
Test Tool	DM 1.2 Conformance test-tool
Preconditions	An interior test node must be set prior to the execution of this test and there should be replace access rights for the DM 1.2 conformance test tool. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Replace on the interior test node command in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	Status to Replace on the interior test node MUST be 200. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200"

MESSAGE ´SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Replace on the interior test node command in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management	Test Tool sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
			Operations Message	

5.14 Device Management Client Conformance Test Group #14

5.14.1 DeviceManagement-v1.2-client-con-1401

Test Case Id	DeviceManagement-v1.2-client-con-1401
Test Object	Client device
Test Case Description	To check if the Test Object enforces ACL. DM 1.2 conformance test toolreplaces the ACL of the test interior node to 'Get=*&Add=*&Replace=*' and issues a Get and Delete command.
Specification Reference	[DMTND] Chapter 7.7.1
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-C-006 Support for receiving 'Delete' DMTND-Prop-C-001 Support for the ACL property
Test Tool	DM 1.2 conformance test tool
Preconditions	There should be interior node configured under test node. DM 1.2 conformance test tool should have replace access rights on the test interior node. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro) to be sure that the interior node exists. 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4) with a "200" or "418" status for the 'Add' command and "200" or "418" status as 'Result'. 4. Test Tool sends a valid package #4 message (see C.5) with a 'Replace' command on the ACL of the test interior node to 'Get=*&Add=*&Replace=*'. 5. Client sends a valid package #3 message (see C.4). 6. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the test interior node 7. Client sends a valid package #3 message (see C.4). 8. Test Tool sends a valid package #4 (see C.5) with a 'Delete' command on the interior node. 9. Client sends a valid package #3 message (see C.4). 10. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	DM 1.2 Conformance test-tool should get a 200 status code for Get and 425

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	status code for Delete. Step 7: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" Step 9: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Delete" used b. a 'Data' tag set to "425"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro) to be sure that the interior node exists.
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4) with a "200" or "418" status for the 'Add' command and "200" or "418" status as 'Result'.
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Replace' command on the ACL of the test interior node to 'Get=*&Add=*&Replace=*'.
5	→		Client-Response	Client sends a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the interior test node.
7	→		Client-Response	Client sends a valid package #3 message (see C.4).
8	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a 'Delete' command on the interior test node.
9	→		Client-Response	Client sends a valid package #3 message (see C.4).
10	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.15 Device Management Client Conformance Test Group #15

5.15.1 DeviceManagement-v1.2-client-con-1501

Test Case Id	DeviceManagement-v1.2-client-con-1501
Test Object	Client device
Test Case Description	DM 1.2 conformance test tool removes the Replace access right for the test leaf node and tries to replace the leaf node under the test interior node.
Specification Reference	[DMTND] Chapter 7.7.1 [DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace' DMTND-Prop-C-001 Support for the ACL property
Test Tool	DM 1.2 conformance test tool
Preconditions	DM 1.2 conformance test toolshould have replace access right on the test interior node and this node should have atleast one leaf node as its child. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro) to be sure that the interior node exists. 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4) with a "200" or "418" status for the 'Add' command and a "200" or "418" status as Result. 4. Test Tool sends a valid package #4 message (see C.5) with an 'Add' command on the leaf node under the interior test node. 5. Client sends a valid package #3 message (see C.4). 6. Test tool sends a valid package #4 (see C.5) with a 'Replace' command on the 'ACL' of the leaf node to an ACL without 'Replace' access like 'Get=*&Add=*&Delete=*' 7. Client sends a valid package #3 message (see C.4). 8. Test tool sends a valid package #4 (see C.5) with a 'Replace' command on the the leaf node value. 9. Client sends a valid package #3 message (see C.4). 10. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	DM 1.2 conformance test toolshould receive a 425 status code for the Replace. Step 9: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	with: a. a ‘CmdRef’ tag set to the value of ‘CmdID’ which the “Replace” used b. a ‘Data’ tag set to “425”

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a ‘DM Session Initialisation Macro’ (see E.1) with an ‘Add' command on the interior test node in the Setup-Response (step 3 of the macro) to be sure that the interior node exists.
2				If required by the client: ‘DM Authentication Macro’.
3	→		Client-Response	Client sends a valid package #3 message (see C.4) with a “200” or “418” status for the ‘Add’ command and a “200” or “418” status as Result.
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with an ‘Add’ command on the leaf node under the interior test node.
5	→		Client-Response	Client sends a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test tool sends a valid package #4 (see C.5) with a ‘Replace’ command on the ‘ACL’ of the leaf node to an ACL without ‘Replace’ access like 'Get=*&Add=*&Delete=*
7	→		Client-Response	Client sends a valid package #3 message (see C.4).
8	←		Server Management Operations Message	Test tool sends a valid package #4 (see C.5) with a ‘Replace’ command on the leaf value
9	→		Client-Response	Client sends a valid package #3 message (see C.4).
10	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.16 Device Management Client Conformance Test Group #16

5.16.1 DeviceManagement-v1.2-client-con-1601

Test Case Id	DeviceManagement-v1.2-client-con-1601
Test Object	Client device
Test Case Description	To check if the Test Object deletes a leaf node correctly.
Specification Reference	[DMREPU] Chapter 6.6.5

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

SCR Reference	DMREPPRO-PCE-C-006 Support for receiving 'Delete'
Test Tool	DM 1.2 conformance test tool
Preconditions	The device should allow Adding and Deleting of nodes. There should be a leaf node under the test node. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro) to be sure that the interior node exists. 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4) with a "200" or "418" status for the 'Add' command and a "200" or "418" status as Results. 4. Test Tool sends a valid package #4 (see C.5) with a 'Replace' command on the 'ACL' of the interior node to an ACL with 'Add' and 'Delete' access like 'Get=*&Add=*&Delete=*' and an 'Add' command on the leaf node under the interior test node. 5. Client sends a valid package #3 message (see C.4). 6. Test tool sends a valid package #4 (see C.5) with a 'Delete' command on the leaf node under the interior test node. 7. Client sends a valid package #3 message (see C.4). 8. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 200/405 status code. Step 7: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Delete" used b. a 'Data' tag set to "200" or "405"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro) to be sure that the interior node exists.
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4) with a "200" or "418"

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
				status for the ‘Add’ command and a “200” or “418” status as Results.
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with a ‘Replace’ command on the ‘ACL’ of the interior node to an ACL with ‘Add’ and ‘Delete’ access like 'Get=*&Add=*&Delete=*' and an ‘Add’ command on the leaf node under the interior test node.
5	→		Client-Response	Client sends a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test tool sends a valid package #4 (see C.5) with a ‘Delete’ command on the leaf node value under the interior test node.
7	→		Client-Response	Client sends a valid package #3 message (see C.4).
8	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.17 Device Management Client Conformance Test Group #17

5.17.1 DeviceManagement-v1.2-client-con-1701

Test Case Id	DeviceManagement-v1.2-client-con-1701
Test Object	Client device
Test Case Description	To check if the Test Object supports Large Object Delivery Mechanism. DM 1.2 conformance test tool issues a Get on ./DevDetail/LrgObj.
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving ‘Get’ SCR-DM-STDOBJ-C-002 Support of DevDetail Object DM-PRO-C-004 Support of Large Object Handling. This is RECOMMENDED for clients. . (ics_large_object)
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a ‘DM Session Initialisation Macro’ (see E.1) with a ‘Get’ command on the './DevDetail/LrgObj' node in the Setup-Response (step 3 of the macro). 2. If required by the client: ‘DM Authentication Macro’. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	DM 1.2 conformance test tool should receive a status of 200 on Get and the result should contain a value of either 'true' or 'false'. Step 3:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with: a. a 'Item' tag with a 'Data' tag set to 'true' or 'false'

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on the './DevDetail/LrgObj' node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4).
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.17.2 DeviceManagement-v1.2-client-con-1702

Test Case Id	DeviceManagement-v1.2-client-con-1702
Test Object	Client device
Test Case Description	To checks if the Test Object follows the Large Object Delivery rules. DM 1.2 conformance test tool adds in the interior test node a leaf node with a Large Object.
Specification Reference	[DMREPU] Chapter 6.6.1
SCR Reference	DMREPPRO-PCE-C-003 Support for receiving 'Add' DM-PRO-C-004 Support of Large Object Handling. This is RECOMMENDED for clients. (ics_large_object)
Test Tool	DM 1.2 conformance test tool
Preconditions	This test is executed only if the Test Object indicates support for Large Object. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Setup-Response (step 3 of the macro) to be sure that the interior node exists. 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4) with a "200" or "418" status for the 'Add' command and a "200" or "418" status as Result. 4. Test Tool sends a valid package #4 message (see C.5) with an 'Add' command on a leaf node under the interior test node with a data size larger than the 'MaxMsgSize' and smaller than the 'MaxObjSize' (NOTE: if the 'MaxObjSize' is smaller than the 'MaxMsgSize', the test is then INCONCLUSIVE). 5. Client sends a valid package #3 message (see C.4). 6. Test Tool sends a valid package #4 message (see C.5) with an 'Add' command on the leaf node with the rest of the large object data. 7. Client sends a valid package #3 message (see C.4). 8. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	DM 1.2 conformance test tool should receive a status code of 200. Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Alert' tag with: a. a 'Data' tag set to "1222" 2. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Add" used b. a 'Data' tag set to "213" Step 7: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Add" used b. a 'Data' tag set to "200"

MESSAGE SEQUENCE

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with an 'Add' command on the interior test node in the Setup-Response (step 3 of the macro) to be sure that the interior node exists.
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4) with a "200" or "418" status for the 'Add' command and a "200" or "418" status as Result.
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with an 'Add' command on a leaf node under the interior test node with a data size larger than the 'MaxMsgSize' and smaller than the 'MaxObjSize' (NOTE: if the 'MaxObjSize' is smaller than the 'MaxMsgSize', the test is then INCONCLUSIVE).
5	→		Client-Response	Client sends a valid package #3 message (see C.4).
6	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with an 'Add' command on the leaf node with the rest of the large object data.
7	→		Client-Response	Client sends a valid package #3 message (see C.4).
8	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.17.3 DeviceManagement-v1.2-client-con-1703

Test Case Id	DeviceManagement-v1.2-client-con-1703
Test Object	Client device
Test Case Description	To check if the Test Object can send Results with a Large Object. DM 1.2 conformance test tool issues a Get on the Large Object node added by the previous test case.
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-S-010 Support for receiving 'Results' DMREPPRO-PCE-C-010 Support for sending 'Results' DM-PRO-C-004 Support of Large Object Handling. This is RECOMMENDED for clients. (ics_large_object)
Test Tool	DM 1.2 conformance test tool
Preconditions	This test is executed only if Test Case 1702 successfully added a leaf node with Large Object. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the leaf node under the interior test node in the Setup-Response (step 3 of the macro).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4) with a 'Results' tag containing leaf node data. 4. If data completed: a. Test Tool sends a valid package #4 message (see C.5) with a 'Status' code 200 to close the session. Else b. Test Tool sends a valid package #4 message (see C.5) with an 'Alert' 1222 command, a 'Status' code 212 for the SyncHdr and a 'Status' code 213 for large object. 5. Repeat Step 3 and 4 until step 4a is done.
Pass-Criteria	DM 1.2 conformance test tool should receive a status code of 200 on the Get and valid results. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. 'Data' tag set to "200" Step4a: Leaf node data is complete and valid.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on the leaf node under the interior test node in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4) with a 'Results' tag containing leaf node data
4a (conditional)	←		Server Management Operations Message	If data completed: Test Tool sends a valid package #4 message (see C.5) with a 'Status' code 200 to close the session.
4b (conditional)	←		Server Management Operations Message	Else: Test Tool sends a valid package #4 message (see C.5) with an 'Alert' 1222 command, a 'Status' code 212 for the SyncHdr and a 'Status' code 213 for the large object. Repeat Step 3 and 4.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.17.4 DeviceManagement-v1.2-client-con-1704

Test Case Id	DeviceManagement-v1.2-client-con-1704
Test Object	Client device
Test Case Description	To check if the Test Object honors the server MaxObjSize. DM 1.2 conformance test tool sends a small MaxObjSize and issues a Get on the Large Object added by Test case 1702
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DM-PRO-C-004 Support of Large Object Handling. This is RECOMMENDED for clients. . (ics_large_object) DMREPPRO-MIE-C-005 Support for 'MaxObjSize' (ics_max_object_size)
Test Tool	DM 1.2 conformance test tool
Preconditions	This test is executed only if Test Case 1702 successfully added a leaf node with Large Object. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the leaf node under the interior test node and with a small 'MaxObjSize' in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4). 4. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	DM 1.2 conformance test tool should receive a status code of 413. Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. 'Data' tag set to "413"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a Get command on the leaf node under the interior test node and with a small 'MaxObjSize' in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
3	→		Client-Response	Client sends a valid package #3 message (see C.4
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.18 Device Management Client Conformance Test Group #18

5.18.1 DeviceManagement-v1.2-client-con-1801

Test Case Id	DeviceManagement-v1.2-client-con-1801
Test Object	Client device
Test Case Description	To check if the Test Object can handle a Get with 'list=Struct'. DM 1.2 Conformance test-tool issues a Get on './DevDetail?list=Struct'.
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get'
Test Tool	DM 1.2 Conformance test-tool
Preconditions	None.
Pass-Criteria	DM 1.2 Conformance test-tool should receive a status code of either 200/406. If status is 200, DM 1.2 Conformance test-tool should receive valid results.

5.19 Device Management Client Conformance Test Group #19

5.19.1 DeviceManagement-v1.2-client-con-1901

Test Case Id	DeviceManagement-v1.2-client-con-1901
Test Object	Client device
Test Case Description	To check if the Test Object can handle a Get with 'list=StructData'. DM 1.2 Conformance test-tool issues a Get on './DevDetail?list=StructData'.
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get'
Test Tool	DM 1.2 Conformance test-tool
Preconditions	None.
Pass-Criteria	DM 1.2 Conformance test-tool should receive a status code of either 200/406. If status is 200, DM 1.2 Conformance test-tool should receive valid results.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.20 Device Management Client Conformance Test Group #20

5.20.1 DeviceManagement-v1.2-client-con-2001

Test Case Id	DeviceManagement-v1.2-client-con-2001
Test Object	Client device
Test Case Description	To check if the Test Object supports Notification Initiated Session using HTTP.
Specification Reference	[DMNOTI] Chapter 6
SCR Reference	SCR-DM-NOTI-C-002 Receiving Notification message DM-PRO-Session-C-001 (ics_notification) Support Server Notification DMREPPRO-PCE-C-001 Support for sending 'Alert'
Test Tool	DM 1.2 conformance test tool
Preconditions	The client is not involved in a session with the test tool.
Test Procedure	1. Test Tool sends a Server Notification message (Package #0 see C.1). 2. If needed the user accepts the DM session establishment. 3. Client sends a valid package #1 message (Setup-Request, see C.2). 4. Test Tool sends a valid package #2 message (Setup-Response see C.3) to close the session.
Pass-Criteria	The Test Object must verify the Notification HTTP headers and data format and connect to DM 1.2 Conformance test-tool with a Alert of 1200. Step 3: The client MUST send a valid package #1 (see C.2) as follows: 1. the message's 'SyncBody' SHALL contain at least an 'Alert' tag with: a. a 'Data' tag set to "1200"

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1	←		Server Notification (Package #0 see C.1)	Test Tool sends a Server Notification message (Package #0 see C.1).
2				If needed the user accepts the DM session establishment.
3	→		Setup-Request	Client sends a valid package #1 message (Setup-Request, see C.2).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
			(Package #1 see C.2)	
4	←		Setup-Response (Package #2 see C.3)	Test Tool sends a valid package #2 message (Setup-Response see C.3) to close the session.

5.21 Device Management Client Conformance Test Group #21

5.21.1 DeviceManagement-v1.2- client-con-2101

Test Case Id	DeviceManagement-v1.2- client-con-2101
Test Object	Client device
Test Case Description	Purpose of this test case is to check if the Test Object returns 405 for an Exec on a node where AccessType property does not contain Exec?.
Specification Reference	[DMREPU] Chapter 6.6.6 [DMTND] Chapter 9.4.3
SCR Reference	DMREPRO-PCE-C-007
Test Tool	DM 1.2 Conformance test-tool
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 405 status code.

5.21.2 DeviceManagement-v1.2-client-con-2102

Test Case Id	DeviceManagement-v1.2-client-con-2102
Test Object	Client device
Test Case Description	Purpose of this test case is to check if the Test Object returns 425 for an Exec on an ACL protected node.
Specification Reference	[DMREPU] Chapter 6.6.6 [DMTND] Chapter 7
SCR Reference	DMREPRO-PCE-C-007, DMTND-Prop-C-001
Test Tool	DM 1.2 Conformance test-tool
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 425 status code.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5.22 Device Management Client Conformance Test Group #22

5.22.1 DeviceManagement-v1.2-client-con-2201

Test Case Id	DeviceManagement-v1.2-client-con-2201
Test Object	Device Management Client
Test Case Description	Purpose of this test case is to check if the Test Object supports implicit addition of parent interior nodes for an addition of a child node whose valid parent/parents does not exist in the DM Tree
Specification Reference	[DMREPU] Chapter 6.6.1 [DMTND] Chapter 7
SCR Reference	NA
Test Tool	DM 1.2 conformance test tool
Preconditions	Parent Node of the child node to be added does not exist. DM 1.2 conformance test tool has sufficient rights to add a node on the DM tree. The client is not involved in a session with the test tool.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Replace' command on the 'ACL' of the parent node of the interior parent node to an ACL with 'Delete' and 'Add' access like 'Get=*&Add=*&Delete=*' and with an 'Delete' command on the interior parent node in the Setup-Response to be sure that the DM 1.2 conformance test tool has sufficient rights to add a node on the DM tree and that the interior parent node doesn't exist. 2. If required by the client: 'DM Authentication Macro'. 3. Client sends a valid package #3 message (see C.4) with a "200" status for the 'Replace' command and with a "200" or "404" status for the 'Delete' command. 4. Test Tool sends a valid package #4 message (see C.5) with an 'Add' command on including the complete URI of the leaf node under the interior parent node. 5. Client sends a valid package #3 message (see C.4) with a "200" status for the 'Add' command. 6. Test Tool sends a valid package #4 message (see C.5) with a 'Get' command on the newly added leaf node 7. Client sends a valid package #3 message (see C.4) with a "200" status for the 'Get' command and with a 'Results' with the value of the newly added leaf node. 8. Test Tool sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	The Test Object MUST return a 200 status code and the value of the newly Added child node.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Step 5: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the complete value of 'CmdID' which the "Add" used b. a 'Data' tag set to "200" Step 7: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'CmdRef' tag set to the value of 'CmdID' which the "Get" used b. a 'Data' tag set to "200" 2. the message's 'SyncBody' SHALL contain at least a 'Results' tag with an Item tag with: a. a 'Source' tag with a 'LocURI' tag set to the complete URI of the leaf node the interior node b. a 'Data' tag set to the value of the newly added leaf node.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Replace' command on the 'ACL' of the parent node of the interior parent node to an ACL with 'Delete' and 'Add' access like 'Get=*&Add=*&Delete=*' and with an 'Delete' command on the interior parent node in the Setup-Response to be sure that the DM 1.2 conformance test tool has sufficient rights to add a node on the DM tree and that the interior parent node doesn't exist.
2				If required by the client: 'DM Authentication Macro'.
3	→		Client-Response	Client sends a valid package #3 message (see C.4) with a "200" status for the 'Replace' command and with a "200" or "404" status for the 'Delete' command.
4	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) with an 'Add' command on including the complete URI of the leaf node under the interior parent node.
5	→		Client-Response	Client sends a valid package #3 message (see C.4) with a "200" status for the 'Add' command.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
6	←		Server Management Operations Message	Test tool sends a valid package #4 (see C.5) with a 'Get' command on the newly added leaf node.
7	→		Client-Response	Client sends a valid package #3 message (see C.4) with a "200" status for the 'Get' command and with a 'Results' with the value of the newly added leaf node.
8	←		Server Management Operations Message	Test Tool sends a valid package #4 message (see C.5) to close the session.

5.23 Device Management Client Conformance TestGroup #23

5.23.1 DeviceManagement-v1.2-client-con-2301

Test Case Id	DeviceManagement-v1.2-client-con-2301
Test Object	Client device
Test Case Description	To check if the Test Object can handle a Get with 'list=TNDS'. Test tool issues a Get on './DevDetail?list=TNDS+ACL+Format+Value'
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 8 and Appendix B
SCR Reference	DMTND-Prop-C-012 Support Get? list=TNDS
Test Tool	DM 1.2 Conformance test tool
Preconditions	Client must not be involved in a session with the server Test Tool has sufficient rights on /DevDetail. /DevDetail node exists on the DM Client Tree and contains some sub nodes.
Test Procedure	1. The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the './DevDetail' node as follows './DevDetail?list=TNDS+ACL+Format+Value' in the Setup-Response (step 3 of the macro). 2. If required by the client: 'DM Authentication Macro'. 3. Client responds to the 'Get' command with a valid package #3 message (see C.4). 4. Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.
Pass-Criteria	DM 1.2 Conformance test tool should receive valid results in TNDS format

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	(including ACLs, Format and Value). Step 3: The client MUST send a valid package #3 (see C.4) as follows: 1. the message's 'SyncBody' SHALL contain at least a 'Status' tag with: a. a 'Cmd' tag set to 'Get' b. a 'Data' tag set to "200" c. a 'CmdRef' tag set to value of 'CmdID' which the "Get" used 2. the message's 'SyncBody' SHALL contain a 'Results' tag with: a. a 'Data' tag containing a TNDS file for the contents of the ./DevDetail' node.

MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1				The test procedure starts with a 'DM Session Initialisation Macro' (see E.1) with a 'Get' command on the './DevDetail' node as follows './DevDetail?list=TNDS+ACL+Format+Value' in the Setup-Response (step 3 of the macro).
2				If required by the client: 'DM Authentication Macro'
3	→		Client Response	Client responds to the 'Get' command with a valid package #3 message (see C.4).
4	←		Server Response	Test tool checks the response and sends a valid package #4 message (see C.5) to close the session.

5.24 Device Management Client Conformance Test Group #24

5.24.1 DeviceManagement-v1.2-client-con-2401

Test Case Id	DeviceManagement-v1.2-client-con-2401
Test Object	Client Device
Test Case Description	To check if the Test Object can handle the copy command, It would be followed by a Get command on both the URI
Specification Reference	[DMREPU] Chapter 6.6.4 and Annex B.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

SCR Reference	DMREPPRO-PCE-C-005 Support for receiving 'Copy' command
Test Tool	DM 1.2 Conformance test-tool
Test Code	
Preconditions	An stablished DM session between Test Tool and DM Client. DM 1.2 Conformance test-tool has sufficient rights on target node. DM 1.2 Conformance test-tool has sufficient rights on sorce node.
Test-Procedure	1- DM 1.2 Conformance test-tool running as a Server issues a copy to the DM Client. 2- DM Client returns 200 Status Code. 3- SCTS Server issues a Get on target node. 4- DM Client returns 200 Status Code. 5- DM Client returns result code. 6- DM 1.2 Conformance test-tool Server issues a Get on source node. 7- DM Client returns 200 Status Code. 8- DM Client returns result code.
Pass-Criteria	- DM Client returs a status 200 code for the copy. - Results from the get commend at the source at the same as results for the get command at the target.

5.25 Device Management Client Conformance Test Group #25

5.25.1 DeviceManagement-v1.2-client-con-2501

void

5.26 Device Management Client Conformance Test Group #26

5.26.1 DeviceManagement-v1.2-client-con-2601

Test Case Id	DeviceManagement-v1.2-client-con-2601
Test Object	Client device
Test Case Description	To test if the Test Object supports transport layer authentication using TLS/SSL over HTTP
Specification Reference	[DMSEC] Chapter 5.3 and 5.5.1.1
SCR Reference	DM-SEC-C-003 Support for transport layer authentication DM-SEC-C-004 Support for HTTP transport (ics_http) DM-SEC-C-013 Identifying that the server is using TLS1.0 or SSL3.0 (ics_secure)

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	DM-SEC-C-016 Supporting at least one of the cipher suites TLS_RSA_WITH_AES _128_CBC_SHA-1, SSL_RSA_WITH_3DES_EDE_CBC_SHA and SSL_RSA_WITH_RC4_128_SHA
Test Tool	DM 1.2 conformance test tool
Preconditions	Client Device supports HTTP (ics_http). Client Device supports HTTPS (ics_secure) using TLS1.0 or SSL3.0. Credentials / certificates necessary to perform authentication have been provisioned in Test Object and Tool.
Test Procedure	1. The test tool asks the user to initiate a Secure DM Session over secure HTTP (HTTPS) 2. Client sends a TLS/SSL "Client Hello" message with the supported cipher suites. 3. Test Tool sends a TLS/SSL "Server Hello" message with the selected cipher suite. 4. TLS/SSL key exchange and authentication procedure. 5. After a successful TLS/SSL Handshake, the Client sends a valid Setup-Request message (Package #1, see C.2) over HTTPS transport using the selected cipher suite. 6. Test Tool sends a valid Setup-Response (Package #2, see C.3) to close the session.
Pass-Criteria	- Test Tool authenticates the DM client - DM client authenticates the Test Tool - DM session is correctly established This is tested as follows: Step 1: The client sends a TLS1.0 or SSL3.0 Handshake Message Type "Client Hello" with a list of its supported cipher suites containing at least one of the following cipher suites: TLS_RSA_WITH_AES _128_CBC_SHA-1 (if using TLS1.0) SSL_RSA_WITH_3DES_EDE_CBC_SHA SSL_RSA_WITH_ RC4_128_SHA Step 5: Client sends a valid Setup-Request message (Package#1 see C.2) ciphered by the cipher suite selected during the successful TLS/SSL authentication. The reception of this Setup-Request means then that: - DM 1.2 conformance Test Tool has authenticated successfully the DM Client - The DM Client has authenticated successfully the DM 1.2 conformance Test Tool - The DM Session is correctly established

MESSAGE SEQUENCE

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Step	Direction		Message	Comment
	UE	SS		
1				The test tool asks the user to initiate a Secure DM Session over secure HTTP (HTTPS).
2	→		TLS/SSL Handshake Message Type: Client Hello	Client sends a TLS/SSL "Client Hello" message with the supported cipher suites.
3	←		TLS/SSL Handshake Message Type: Server Hello	Test Tool sends a TLS/SSL "Server Hello" message with the selected cipher suite.
4	←→		TLS/SSL Handshake messages	Ciphered TLS/SSL authentication and key exchange.
5	→		Ciphered Setup-Request (Package #1 see C.2)	Client sends a valid Setup-Request message.
6	←		Ciphered Setup-Response (Package #2 see C.3)	Test Tool sends a valid Setup-Response (Package #2, see C.3) to close the session.

5.26.2 DeviceManagement-v1.2-client-con-2602

void

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6. Device Management Server Conformance Test Cases

The Test cases are listed according to the Test Groups. Each Test Group describes its Test cases and the relevant information regarding the message exchanged.

6.1 Device Management Server Conformance Test Group #1

6.1.1 DeviceManagement-v1.2-server-con-0101

Test Case Id	DeviceManagement-v1.2-server-con-0101
Test Object	Server device
Test Case Description	To check if the Test Object implements 'Server Layer Authentication'. SCTS sends SyncHdr with wrong credentials.
Specification Reference	[DMSEC] Chapter 5.1 [REPRO] Chapter 6.2.2
SCR Reference	DSDM-RepPro-MCE-S-002 Support SyncHdr
Test Tool	DM 1.2 Conformance test-tool as a client
Preconditions	UserID and Password should be configured for the DMAccount in use.
Pass-Criteria	The Test Object MUST return a 401 status code on the SyncHdr

6.2 Device Management Server Conformance Test Group #2

6.2.1 DeviceManagement-v1.2-server-con-0201

Test Case Id	DeviceManagement-v1.2-server-con-0201
Test Object	Server device
Test Case Description	To check if the Test Object implements 'Server Layer Authentication'. SCTS sends SyncHdr with no credentials.
Specification Reference	[DMSEC] Chapter 5.1 [REPRO] Chapter 6.2.2
SCR Reference	DSDM-RepPro-MCE-S-002 Support SyncHdr
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 407 status code on the first SyncHdr

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.2.2 DeviceManagement-v1.2-server-con-0202

Test Case Id	DeviceManagement-v1.2-server-con-0202
Test Object	Server device
Test Case Description	To check if the Test Object accepts the credentials sent and proceeds with the Sync Session.
Specification Reference	[DMSEC] Chapter 5.1 [REPRO] Chapter 6.2.2
SCR Reference	DSDM-RepPro-MCE-S-002 Support SyncHdr
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	The Test Object MUST return either a 200 or 212 status code on the first/second SyncHdr.

6.2.3 DeviceManagement-v1.2-server-con-0203

Test Case Id	DeviceManagement-v1.2-server-con-0203
Test Object	Server device
Test Case Description	To check if the Test Object processed the Replace command with devInfo without errors.
Specification Reference	[DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-S-002 Support for 'Replace' SCR-DM-STDOBJ-S-001 Support of DevInfo object
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 200 status code on the Replace.

6.2.4 DeviceManagement-v1.2-server-con-0204

Test Case Id	DeviceManagement-v1.2-server-con-0204
Test Object	Server device
Test Case Description	To check if the Test Object processed the Alert command without errors.
Specification Reference	[DMREPU] Chapter 6.6.2
SCR Reference	DMREPPRO-PCE-S-001 Support for 'Alert'

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 200 status code on the Alert.

6.3 Device Management Server Conformance Test Group #3

6.3.1 DeviceManagement-v1.2-server-con-0301

Test Case Id	DeviceManagement-v1.2-server-con-0301
Test Object	Server device
Test Case Description	To check if the Test Object generates a valid Get command on a existing interior node (Root Node, '.').
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a 200 status code on the Get.

6.4 Device Management Server Conformance Test Group #4

6.4.1 DeviceManagement-v1.2-server-con-0401

Test Case Id	DeviceManagement-v1.2-server-con-0401
Test Object	Server device
Test Case Description	To check if the Test Object generates a valid Add command to add a leaf node (./SCTSValue).
Specification Reference	[DMREPU] Chapter 6.6.1
SCR Reference	DMREPPRO-PCE-S-003 Support for sending 'Add'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a 200 status code on the Add.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.5 Device Management Server Conformance Test Group #5

6.5.1 DeviceManagement-v1.2-server-con-0501

Test Case Id	DeviceManagement-v1.2-server-con-0501
Test Object	Server device
Test Case Description	To check if the Test Object generates a valid Replace command to replace the contents of a leaf node (./SCTSValue).
Specification Reference	[DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-S-002 Support for 'Replace'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a 200 status code on the Replace.

6.6 Device Management Server Conformance Test Group #6

6.6.1 DeviceManagement-v1.2-server-con-0601

Test Case Id	DeviceManagement-v1.2-server-con-0601
Test Object	Server device
Test Case Description	To check if the Test Object generates a valid Delete command to delete a leaf node (./SCTSValue).
Specification Reference	[DMREPU] Chapter 6.6.5
SCR Reference	DMREPPRO-PCE-S-006 Support for sending 'Delete'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a 200 status code on the Delete.

6.7 Device Management Server Conformance Test Group #7

6.7.1 DeviceManagement-v1.2-server-con-0701

Test Case Id	DeviceManagement-v1.2-server-con-0701
Test Object	Server device
Test Case Description	To check if the Test Object can handle multiple messages.
Specification Reference	[DMPRO] – Chapter 6, 8

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	[DMREPU] – Chapter 6.1.7
SCR Reference	DM-PRO-Mul-S-001 Last message within multiple messages must contain Final DM-PRO-Mul-S-002 If message that is not the last one within Multiple Messages then the Next Message or Abort Alert must be sent DMREPPRO-MIE-S-003 Support for sending 'MaxMsgSize' DMREPPRO-MIE-S-004 Support for receiving 'MaxMsgSize'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	The session MUST complete successfully.

6.8 Device Management Server Conformance Test Group #8

6.8.1 DeviceManagement-v1.2-server-con-0801

Test Case Id	DeviceManagement-v1.2-server-con-0801
Test Object	Server device
Test Case Description	To check if the Test Object generates a valid Sequence command. Sequence should contain two Replace commands.
Specification Reference	[DMREPU] Chapter 6.6.14 [DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-S-002 Support for 'Replace' DMREPPRO-PCE-S-009 Support for sending 'Sequence'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a 200 status code on the Sequence.

6.9 Device Management Server Conformance Test Group #9

6.9.1 DeviceManagement-v1.2-server-con-0901

Test Case Id	DeviceManagement-v1.2-server-con-0901
Test Object	Server device
Test Case Description	To check if the Test Object generates a valid Atomic command. Atomic should contain two Replace commands.
Specification Reference	[DMREPU] Chapter 6.6.3

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

SCR Reference	DMREPPRO-PCE-S-002 Support for 'Replace' DMREPPRO-PCE-S-004 Support for sending 'Atomic'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a 200 status code on the Atomic.

6.10 Device Management Server Conformance Test Group #10

6.10.1 DeviceManagement-v1.2-server-con-1001

Test Case Id	DeviceManagement-v1.2-server-con-1001
Test Object	Server device
Test Case Description	To check if the Test Object can Add a text Large Object.(./SCTSLrgObjText).
Specification Reference	[DMREPU] Chapter 6.6.1
SCR Reference	DMREPPRO-PCE-S-003 Support for sending 'Add'DMREPPRO-MIE-S-003 Support for sending 'MaxMsgSize' DMREPPRO-MIE-S-005 Support for 'MaxObjSize' DMREPPRO-MIE-S-008 Support for 'Size'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	Successfully Add the text based large object that is larger than MaxMsgSize.Final status to Add should be 200.

6.10.2 DeviceManagement-v1.2-server-con-1002

Test Case Id	DeviceManagement-v1.2-server-con-1002
Test Object	Server device
Test Case Description	To check if the Test Object can Get a text Large Object.(./SCTSLrgObjText).
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-S-008 Support for sending 'Get' DMREPPRO-MIE-S-004 Support for receiving 'MaxMsgSize' DMREPPRO-MIE-S-005 Support for 'MaxObjSize' DMREPPRO-MIE-S-008 Support for 'Size'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Pass-Criteria	Successfully Get the text based large object.

6.11 Device Management Server Conformance Test Group #11

6.11.1 DeviceManagement-v1.2-server-con-1101

Test Case Id	DeviceManagement-v1.2-server-con-1101
Test Object	Server device
Test Case Description	To check if the Test Object can Add a binary Large Object.(./SCTSLrgObjBin).
Specification Reference	[DMREPU] Chapter 6.6.1
SCR Reference	DMREPPRO-PCE-S-003 Support for sending 'Add' DMREPPRO-MIE-S-003 Support for sending 'MaxMsgSize' DMREPPRO-MIE-S-005 Support for 'MaxObjSize' DMREPPRO-MIE-S-008 Support for 'Size'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	Successfully Add the binary large object that is larger than MaxMsgSize. Final status to Add should be 200.

6.11.2 DeviceManagement-v1.2-server-con-1102

Test Case Id	DeviceManagement-v1.2-server-con-1102
Test Object	Server device
Test Case Description	To check if the server can Get a binary Large Object from the client.
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-MIE-S-004 Support for receiving 'MaxMsgSize' DMREPPRO-MIE-S-005 Support for 'MaxObjSize' DMREPPRO-MIE-S-008 Support for 'Size'
Test Tool	SCTS DM 1.2 as a client
Preconditions	Passed test case 1101
Pass-Criteria	Final status to Get on <Leaf> MUST be 200 and the Results must contain the large object.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.12 Device Management Server Conformance Test Group #12

6.12.1 DeviceManagement-v1.2-server-con-1201

Test Case Id	DeviceManagement-v1.2-server-con-1201
Test Object	Server device
Test Case Description	To check if the Test Object responds with a status after SCTS has sent a Session Abort Alert 1223.
Specification Reference	[DMPRO] Chapter 8.1
SCR Reference	DM-PRO-Abort-S-002 Receiving Session Abort Alert
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	Response to Session Abort Alert MUST be 200.

6.13 Device Management Server Conformance Test Group #13

6.13.1 DeviceManagement-v1.2-server-con-1301

Test Case Id	DeviceManagement-v1.2-server-con-1301
Test Object	Server device
Test Case Description	To check if the Test Object can send a Get with '.?list=Struct' and handle the results correctly.
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-S-008 Support for sending ‘Get’
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should receive a valid Get and the Test Object should send status of 200 to all the results sent by SCTS.

6.14 Device Management Server Conformance Test Group #14

6.14.1 DeviceManagement-v1.2-server-con-1401

Test Case Id	DeviceManagement-v1.2-server-con-1401
Test Object	Server device
Test Case Description	To check if the Test Object can send a Get with '.?list=StructData' and handle the results correctly.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	DMREPPRO-PCE-S-008 Support for sending 'Get'
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should receive a valid Get and the Test Object should send status of 200 to all the results sent by SCTS.

6.15 Device Management Server Conformance Test Group #15

6.15.1 DeviceManagement-v1.2-server-con-1501

Test Case Id	DeviceManagement-v1.2-server-con-1501
Test Object	Server device
Test Case Description	To check if the Test Object can send a UI Display Alert.
Specification Reference	[DMPRO] Chapter 10.2.1
SCR Reference	DM-PRO-UI-S-001 Sending Display Alert
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a status code of 200 to the Alert.

6.16 Device Management Server Conformance Test Group #16

6.16.1 DeviceManagement-v1.2-server-con-1601

Test Case Id	DeviceManagement-v1.2-server-con-1601
Test Object	Server device
Test Case Description	To check if the Test Object can send a UI Confirmation Alert.
Specification Reference	[DMPRO] Chapter 10.2.2
SCR Reference	DM-PRO-UI-S-002 Sending Confirm or Reject Alert
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a status code of 200/304 to the Alert.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.17 Device Management Server Conformance Test Group #17

6.17.1 DeviceManagement-v1.2-server-con-1701

Test Case Id	DeviceManagement-v1.2-server-con-1701
Test Object	Server device
Test Case Description	To check if the Test Object can send a UI User Input Alert.
Specification Reference	[DMPRO] Chapter 10.2.3
SCR Reference	DM-PRO-UI-S-003 Sending Text Input Alert
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	SCTS should respond with a status code of 200 to the Alert.

6.18 Device Management Server Conformance Test Group #18

6.18.1 DeviceManagement-v1.2-server-con-1801

Test Case Id	DeviceManagement-v1.2-server-con-1801
Test Object	Server device
Test Case Description	To check if the Test Object supports Notification Initiated Session using HTTP.
Specification Reference	[DMNOTI] Chapter 6
SCR Reference	SCR-DM-NOTI-S-002 Sending of Notification message
Test Tool	SCTS DM 1.2 as a client
Preconditions	None.
Pass-Criteria	The Test Object must verify the Notification HTTP headers and data format and connect to SCTS with a Alert of 1200.

6.19 Device Management Server Conformance Test Group #19

6.19.1 DeviceManagement-v1.2-server-con -1901

Test Case Id	DeviceManagement-v1.2-server-con -1901
Test Object	Device management server
Test Case Description	Purpose of this test case is to check if the Test Object can receive, parse and send status back to the Test Tool for a Generic alert 1226.
Specification Reference	[DMPRO] Chapter 8.7

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

SCR Reference	DM-PRO-S-009 DM-PRO-GAlert-S-001 DM-PRO-GAlert-S-002
Tool	Test Tool as DM1.2 client
Test Code	
Preconditions	None.
Test Procedure	1. Test tool sends Package #1 (Client Initiated Management Alert) to the Test Object. 2. Test tool sends Generic Alert message to the Test Object.
Pass-Criteria	The Test Object should respond with a status code of 200 or 202 to the Alert.

6.20 Device Management Server Conformance Test Group #20

6.20.1 DeviceManagement-v1.2-server-con-2001

Test Case Id	DeviceManagement-v1.2-server-con-2001
Test Object	DM Server
Test Case Description	To check if the Test Object can support sending and receiving a correlator
Specification Reference	[DMREPU] Chapter 6.3 and Chapter 6.6.2
SCR Reference	DMREPPRO-PCE-S-007 Support for Sending 'Exec' DM-PRO-S-009 Support of 'Generic Alert' DMREPPRO-DDE-S-001 Support for sending 'Correlator' DMREPPRO-DDE-S-002 Support for receiving 'Correlator' DM-PRO-Galert-S-001 Support for receiving, parsing and send Status Back to Client
Test Tool	SCTS DM 1.2 as a Client
Test Code	
Preconditions	An established DM session between Test Tool and DM Server. A node capable of receiving an exec node exists in the DM Tree of the SCTS.(e.g. ../x*/TestExec) DM Server has sufficient rights to exec a node on that node of the SCTS Tool.
Test procedure	1- Test object need to be configured to send an Exec to the specified node

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2- DM Server sends an Exec command to the node with a Correlator. 3- SCTS returns a Generic Alert including the same correlator.
Pass-Criteria	-DM server is able to send exec including the correlator. -SCTS returns 200 for a valid Exec command -DM Server returns a status code 200 or 202 in response to the Generic alert..

6.21 Device Management Server Conformance Test Group #21

6.21.1 DeviceManagement-v1.2-server-con-2101

Test Case Id	DeviceManagement-v1.2-server-con-2101
Test Object	Server device
Test Case Description	To test if the Test Object supports transport layer authentication using TLS over HTTP
Specification Reference	[DMSEC] Chapter 5.3 and 5.5.1.1
SCR Reference	DM-SEC-S-002 Support for client authentication at the transport layer DM-SEC-S-014 Support for HTTP transport DM-SEC-S-015 Support for TLS 1.0 [TLS] DM-SEC-S-017 Using OMA DM over HTTP DM-SEC-S-018 Using TLS DM-SEC-S-020 Supporting all three cipher suites TLS_RSA_WITH_AES_128_CBC_SHA-1, TLS_RSA_WITH_3DES_EDE_CBC_SHA and TLS_RSA_WITH_RC4_128_SHA
Test Tool	SCTS DM 1.2 as a client
Preconditions	Server Device supports HTTP. Credentials / certificates necessary to perform authentication have been provisioned in Test Object and Tool.
Test Procedure	2. Test Tool initiates a session with the Server requesting transport layer authentication using TLS and indicating that it wishes to use cipher suite TLS_RSA_WITH_AES_128_CBC_SHA-1 3. Server authenticates the Test Tool and sends the information the Test Tool needs to authenticate it. 4. Test Tool authenticates the Server and the TLS session is established. 5. DM session is established between Server and Test Tool 6. Repeat steps 1-4 using cipher suites TLS_RSA_WITH_3DES_EDE_CBC_SHA and TLS_RSA_WITH_RC4_128_SHA
Pass-Criteria	- Test Tool authenticates the DM server

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	- DM server authenticates the Test Tool - DM session is correctly established

6.21.2 DeviceManagement-v1.2-server-con-2102

Test Case Id	DeviceManagement-v1.2-server-con-2102
Test Object	Server device
Test Case Description	To test if the Test Object supports transport layer authentication using SSL 3.0 over HTTP
Specification Reference	[DMSEC] Chapter 5.3 and 5.5.1.1
SCR Reference	DM-SEC-S-002 Support for client authentication at the transport layer DM-SEC-S-014 Support for HTTP transport DM-SEC-S-016 Support for SSL3.0 [SSL3.0] DM-SEC-S-017 Using OMA DM over HTTP DM-SEC-S-019 Using SSL3.0 DM-SEC-S-021 Support for both of SSL_RSA_WITH_RC4_128_SHA and SSL_RSA_WITH_3DES_EDE_CBC_SHA
Test Tool	SCTS DM 1.2 as a client
Preconditions	Server Device supports HTTP. Credentials / certificates necessary to perform authentication have been provisioned in Test Object and Tool.
Test Procedure	1- Test Tool initiates a session with the Server requesting transport layer authentication using SSL 3.0 and indicating that it wishes to use cipher suite SSL_RSA_WITH_RC4_128_SHA 2- Server authenticates the Test Tool and sends the information the Test Tool needs to authenticate it. 3- Test Tool authenticates the Server and the TLS session is established. 4- DM session is established between Server and Test Tool 5- Repeat steps 1-4 using cipher suite SSL_RSA_WITH_3DES_EDE_CBC_SHA
Pass-Criteria	- Test Tool authenticates the DM server - DM server authenticates the Test Tool - DM session is correctly established

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

7. Device Management Interoperability Test Cases

7.1 DeviceManagement-v1.2-int-001

Test Case Id	DeviceManagement-v1.2-int-001
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with MD-5 client authentication.
Specification Reference	[DMSEC] Chapter 5.3
SCR Reference	DM-SEC-C-001 Client must authenticate itself to a server DM-SEC-C-005 Send credentials to server DM-SEC-C-008 Support for OMA DM syncml:auth-md5 type authentication DM-SEC-S-006 MD5 challenge to client
Preconditions	None.
Test Procedure	1. Configure the SyncML DM Server to require MD5 authentication from the client. The client credentials shall be sent in Package 1, thereby avoiding the need for the server to challenge for them. 2. Establish the connection from the client. 3. Complete the DM session. 4. Check both the server and the client to verify the DM session has completed without any failures
Pass-Criteria	1. DM session runs through without any communication problem.

7.2 DeviceManagement-v1.2-int-002

Test Case Id	DeviceManagement-v1.2-int-002
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with MD-5 server authentication.
Specification Reference	[DMSEC] Chapter 5.3
SCR Reference	DM-SEC-C-002 Client must authenticate a server DM-SEC-C-006 Challenge Server DM-SEC-C-008 Support for OMA DM syncml:auth-md5 type authentication DM-SEC-S-006 MD5 challenge to client

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Preconditions	None.
Test Procedure	5h. Configure the SyncML DM client to require MD5 authentication from the server. The server credentials may be sent in Package 2 or not. If not, the client will issue a challenge and the server will subsequently provide the credentials 5i. Establish the connection from the client. 5j. Complete the DM session. 5k. Check both the server and the client to verify the DM session has completed without any failures.
Pass-Criteria	1. DM session runs through without any communication problem.

7.3 DeviceManagement-v1.2-int-003

Test Case Id	DeviceManagement-v1.2-int-003
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the GET command on a leaf node.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 6
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-S-008 Support for sending 'Get'
Preconditions	None.
Test Procedure	1. Use the client default authentication and connection settings. 2. In the server, configure it to perform a Get command on a leaf node. ■ Example: In the server choose to receive the data value of a leaf node of Device Detail by sending a Get command: Get ./DevDetail/LrgObj 3. Establish the connection from the client 4. Client returns data value for .the given leaf node. ■ Example: In the server sent a Get command on ./DevDetail/LrgObj, the client returns true or false (must be lowercase) The Meta Format is also returned and MUST be bool. 5. Complete the DM session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	6. Verify the DM session completes without any errors. 7. Check the server received the data value from the client.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Server received the proper device detail from client. (Check from XML log if necessary.)

7.4 DeviceManagement-v1.2-int-004

Test Case Id	DeviceManagement-v1.2-int-004
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the GET command on a node that doesn't exist.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 6
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-S-008 Support for sending 'Get'
Preconditions	None.
Test Procedure	1. Use the client default authentication and connection settings. 2. In the server, configure it to perform a Get on a non-existant node. ■ Example: In the server choose to receive the data value of a non-existant URI node by sending a Get command: Get ./XYZ 3. Establish the connection from the client 4. The client returns a status code of 404 (Not found). 5. Complete the DM session. 6. Verify the DM session completes without any errors. 7. Check the server received the data value from the client.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Server received the proper device detail from client. (Check from XML log if necessary.)

7.5 DeviceManagement-v1.2-int-005

Test Case Id	DeviceManagement-v1.2-int-005

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the GET command on an interior node.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 6.2.2
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-S-008 Support for sending 'Get'
Preconditions	None.
Test Procedure	1. Use the client default authentication and connection settings. 2. In the server, configure it to perform a Get on an interior node. ■ Example: In the server choose to receive the data value of the URI node of Device Detail by sending a Get command: Get ./DevDetail/URI 3. Establish the connection from the client 4. Client returns data value that includes leaf node names. ■ Example: Client returns a data value that includes the leaf node names MaxDepth, MaxTotLen, and MaxSegLen separated by the "/" character. Note: The names can appear in any order. 5. Complete the DM session. 6. Verify the DM session completes without any errors. 7. Check the server received the data value from the client.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Server received the proper device detail from client. (Check from XML log if necessary.)

7.6 DeviceManagement-v1.2-int-006

Test Case Id	DeviceManagement-v1.2-int-006
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the GET on an inaccessible leaf node.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 6.2.5
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get'

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	DMREPPRO-PCE-S-008 Support for sending 'Get'
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings 2. In the server, configure it to perform a Get command on an inaccessible leaf node. (An inaccessible leaf node for a Get is determined by looking at the DDF for the object and making sure the <DFProperties><AccessType> does not allow Get.) ■ Example: In the server choose to receive the server password for reading DM Account settings. By assumption, the DM Server provisioned the DM Account parameters at an earlier time. Thus, to read the DM Account settings, the server sends the following commands, in which instance_name is replaced by the DM Account name of the server: Get ./SyncML/DMAcc/instance_name/ServerPW 3. Establish the connection from the client. 4. The client returns a status code of 405 (Command not allowed). 5. Complete the DM session.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Client and Server show proper error messages.

7.7 DeviceManagement-v1.2-int-007

Test Case Id	DeviceManagement-v1.2-int-007
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with REPLACE on permanent leaf node.
Specification Reference	[DMREPU] Chapter 6.6.11 [DMTND] Chapter 6.2.3
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace' DMREPPRO-PCE-S-002 Support for 'Replace'
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings. 2. In the server, configure it to perform a Replace command on a permanent leaf node. ■ Example: In the server choose to replace the manufacturer identifier by sending a Replace command: Replace ./DevInfo/Man

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	3. Establish the connection from the client 4. Client returns a status of 405 (Command not allowed). 5. Complete the DM session. 6. Check the DM session goes without any errors.
Pass-Criteria	1. The server successfully sent the requested value to the client. 2. Client and Server show proper error messages. 3. The session runs through without any communication problem till the end.

7.8 DeviceManagement-v1.2-int-008

Test Case Id	DeviceManagement-v1.2-int-008
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with management node ACL behaviour.
Specification Reference	[DMTND] Chapter 7.7.1
SCR Reference	DMTND-Prop-C-001 Support for the ACL property DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-S-008 Support for sending 'Get' DMREPPRO-PCE-C-002 Support for 'Replace' DMREPPRO-PCE-S-002 Support for 'Replace'
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings. 2. In the server, configure it to perform a Sequence containing the following commands: ■ Example (the URI is negotiated between client and server vendor): a. Get ./SyncML/DMAcc/*instance_name*?prop=ACL. b. Replace ./SyncML/DMAcc/instance_name?prop=ACL <Data>Add=*&Delete=*&Replace=*</Data> c. Get ./SyncML/DMAcc/instance_name?prop=ACL 3. Establish the connection from the client 4. Complete the DM session. 5. Check the DM session goes without any errors.
Pass-Criteria	1. The server successfully sent the requested value to the client.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2. Client and Server show proper error messages. In a successful test, the status will be 200 for the sequence, 200 for the first Get, 200 for the Replace, and 425 (Permission Denied) for the 2nd Get. 3. The session runs through without any communication problem till the end.

7.9 DeviceManagement-v1.2-int-009

Test Case Id	DeviceManagement-v1.2-int-009
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the error handling when connection failure occurs during the SyncML DM session.
Specification Reference	[DMPRO] Chapter 8
SCR Reference	
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings. 2. Establish the connection from the client, but Stop the client-side data connection after the actual DM object exchange starts. For example, the client may be powered off during the session. 3. Check from the server that the server shows the proper error message. 4. Establish the connection from the client. 5. Complete the DM session. 6. Check the DM session goes without any errors. 7. Check both from the server and the client that DM session has completed without any failures.
Pass-Criteria	1. Second DM session runs through without any communication problem till the end. 2. Client and Server show proper error messages.

7.10 DeviceManagement-v1.2-int-010

Test Case Id	DeviceManagement-v1.2-int-010
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with HMAC client authentication.
Specification Reference	[DMSEC] Chapter 5.4
SCR Reference	DM-SEC-C-010 Integrity checking using HMAC-MD5 DM-SEC-C-011 Inserting HMAC in transport

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	DM-SEC-C-012 Using HMAC for all subsequent messages DM-SEC-S-011 Integrity checking using HMAC-MD5 DM-SEC-S-012 Inserting HMAC in transport DM-SEC-S-013 Using HMAC for all subsequent messages
Preconditions	None.
Test Procedure	1. Configure DM settings on the SyncML DM Server and Client. 2. Configure the SyncML DM Server to require HMAC authentication from the client. 3. Establish the connection from the client. 4. Complete the DM session. 5. Check both the server and the client to verify the DM session has completed without any failures.
Pass-Criteria	1. DM session runs through without any communication problem.

7.11 DeviceManagement-v1.2-int-011

Test Case Id	DeviceManagement-v1.2-int-011
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with HMAC server authentication.
Specification Reference	[DMSEC] Chapter 5.4
SCR Reference	DM-SEC-C-010 Integrity checking using HMAC-MD5 DM-SEC-C-011 Inserting HMAC in transport DM-SEC-C-012 Using HMAC for all subsequent messages DM-SEC-S-011 Integrity checking using HMAC-MD5 DM-SEC-S-012 Inserting HMAC in transport DM-SEC-S-013 Using HMAC for all subsequent messages
Preconditions	None.
Test Procedure	1. If applicable, set in the SyncML DM client to require HMAC authentication from the server. The server credentials may be sent in Package 2 or not. If not, the client will issue a challenge and the server will subsequently provide the credentials. (In the case it is not possible in the client, go on to the procedure 3.)

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2. Establish the connection from the client. 3. Complete the DM session. 4. Check both the server and the client to verify the DM session has completed without any failures.
Pass-Criteria	1. DM session runs through without any communication problem.

7.12 DeviceManagement-v1.2-int-012

Test Case Id	DeviceManagement-v1.2-int-012
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the large object/multiple commands.
Specification Reference	[DMPRO] Chapter 6, 7
SCR Reference	DMREPPRO-PCE-C-003 Support for receiving 'Add' DMREPPRO-PCE-S-003 Support for sending 'Add' DMREPPRO-PCE-C-010 Support for sending 'Results' DMREPPRO-PCE-S-010 Support for receiving 'Results' DM-PRO-C-004 Support of Large Object Handling. This is RECOMMENDED for clients. DM-PRO-S-004 Support of Large Object Handling DMREPPRO-MIE-C-005 Support for 'MaxObjSize'
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings 2. In the server, configure it to perform an Add of a Large Object. Note: This test also exercises multiple commands per package. a. Example: Add <Node>/LargeObj (this URI is negotiated between client and server vendor). 3. Establish the connection from the client. 4. Complete the DM session. 5. Check the DM session goes without any errors. 6. Check the DM server sent the proper response to the client.
Pass-Criteria	1. DM session runs through without any communication problem.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2. Synchronisation runs through with a basic DM authentication.

7.13 DeviceManagement-v1.2-int-013

Test Case Id	DeviceManagement-v1.2-int-013
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with notification initiated session.
Specification Reference	[DMNOTI] Chapter 5, 6 [DMSEC] Chapter 5.6 [DMREPU] Chapter 7
SCR Reference	SCR-DM-NOTI-C-001 Support of Server-Alerted Management Session SCR-DM-NOTI-S-001 Support of Server-Alerted Management Session DM-PRO-Session-C-003 Sending Server-Initiated mgmt Alert
Preconditions	None.
Test Procedure	1. Make the server initiate the client connecting into the server using the Notification Initiated Session mechanism. 2. Client should use data in the notification to start a SyncML DM session with the server. 3. The server should receive an Alert 1200 (Server Initiated Management) in package 1 from the client. 4. Complete the DM session. 5. Check the DM session goes without any errors.
Pass-Criteria	1. Client received the proper data in the notification to start a SyncML session with the server. 2. DM session runs through without any communication problem.

7.14 DeviceManagement-v1.2-int-014

Test Case Id	DeviceManagement-v1.2-int-014
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with Server Initiated bootstrap using Client Provisioning Profile.
Specification Reference	[DMBOOT] Chapter 5.3

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	[DMSEC] Chapter 5.7.1
SCR Reference	DM-BOOT-C-001 Support for OMA Client Provisioning Profile DM-BOOT-S-001 Support for OMA Client Provisioning Profile
Preconditions	• A DM Client to be bootstrapped supporting CP Profile • A DM Server supporting CP profile with bootstrap information (DM account and connectivity information) (Reference Content stored in the server CP_Prov_doc_1.xml)
Test Procedure	1. DM server sends out the bootstrap message 2. On the client select to accept the incoming bootstrap message if necessary. 3. Check that the device is bootstrapped with the bootstrap information sent by the DM server.
Pass-Criteria	1. DM client processes correctly the bootstrap message 2. Bootstrap information contained in the w7 AC is successfully mapped to DM tree and DM client is correctly configured 3. DM client is able to successfully establish a DM session with the server that initiated the bootstrap

7.15 DeviceManagement-v1.2-int-015

Test Case Id	DeviceManagement-v1.2-int-015
Test Object	Client and Smart Card device
Test Case Description	Purpose of this test is to check that a Device Management client supports bootstrap from the Smart Card using the Client Provisioning profile and EF Bootstrap
Specification Reference	[DMBOOT] Section 5.3 [PROVSC]
SCR Reference	DM-BOOT-C-001 DM-BOOT-C-003 DM-BOOT-C-004 DM-BOOT-C-005 DM-BOOT-C-006 DM-BOOT-C-008 DM-BOOT-C-009
Preconditions	• A DM client to be configured supporting Client Provisioning Profile.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	• A Smart Card with bootstrap information containing a w7 APPLICATION characteristic (Reference content stored in the Smart Card: CP_Prov_doc_1.xml) in EF Bootstrap. Provisioning files EF Config1 and EF Config2 are not present. No DM Profile files (e.g. EF_DM_Bootstrap) are present in the Smart Card.
Test Procedure	1. Insert a smart card in the handset. 2. Select to configure the terminal from the Smart Card if necessary. 3. On the client, select to save the information read from the Smart Card on the client if necessary. 4. Check that the bootstrap information is saved in the client. 5. Check that the client can use the bootstrap information
Pass-criteria	1. The client is able to read the bootstrap configuration correctly from the smart card. 2. The bootstrap information contained in the w7 application characteristic is correctly mapped into the DM tree 3. DM client is able to successfully establish a DM session with the DM server indicated in the bootstrap message

7.16 DeviceManagement-v1.2-int-015a

Test Case Id	DeviceManagement-v1.2-int-015a
Test Object	Client and Smart Card device
Test Case Description	Purpose of this test is to check that a Device Management client supports bootstrap from the Smart Card using the Client Provisioning profile and EF_Config1
Specification Reference	[DMBOOT] Section 5.3 [PROVSC]
SCR Reference	DM-BOOT-C-001 DM-BOOT-C-003 DM-BOOT-C-004 DM-BOOT-C-005 DM-BOOT-C-006 DM-BOOT-C-008 DM-BOOT-C-009
Preconditions	• A DM client to be configured supporting Client Provisioning Profile. • A Smart Card with bootstrap information containing a w7 APPLICATION characteristic (Reference content stored in the

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Smart Card: CP_Prov_doc_1.xml) in EF_Config1. Provisioning files EF_Bootstrap and EF_Config2 are not present. No DM Profile files (e.g. EF_DM_Bootstrap) are present in the Smart Card.
Test Procedure	1. Insert a smart card in the handset. 2. Select to configure the terminal from the Smart Card if necessary. 3. On the client, select to save the information read from the Smart Card on the client if necessary. 4. Check that the bootstrap information is saved in the client. 5. Check that the client can use the bootstrap information
Pass-criteria	1. The client is able to read the bootstrap configuration correctly from the smart card. 2. The bootstrap information contained in the w7 application characteristic is correctly mapped into the DM tree 3. DM client is able to successfully establish a DM session with the DM server indicated in the bootstrap message

7.17 DeviceManagement-v1.2-int-015b

Test Case Id	DeviceManagement-v1.2-int-015b
Test Object	Client and Smart Card device
Test Case Description	Purpose of this test is to check that a Device Management client supports bootstrap from the Smart Card using the Client Provisioning profile and EF_Config2
Specification Reference	[DMBOOT] Section 5.3 [PROVSC]
SCR Reference	DM-BOOT-C-001 DM-BOOT-C-003 DM-BOOT-C-004 DM-BOOT-C-005 DM-BOOT-C-006 DM-BOOT-C-008 DM-BOOT-C-009
Preconditions	• A DM client to be configured supporting Client Provisioning Profile. • A Smart Card with bootstrap information containing a w7 APPLICATION characteristic (Reference content stored in the Smart Card: CP_Prov_doc_1.xml) in EF_Config2. Provisioning files EF_Bootstrap and EF_Config1 are not present. No DM Profile files (e.g. EF_DM_Bootstrap) are present in the Smart Card.
Test Procedure	1. Insert a smart card in the handset.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2. Select to configure the terminal from the Smart Card if necessary. 3. On the client, select to save the information read from the Smart Card on the client if necessary. 4. Check that the bootstrap information is saved in the client. 5. Check that the client can use the bootstrap information
Pass-criteria	1. The client is able to read the bootstrap configuration correctly from the smart card. 2. The bootstrap information contained in the w7 application characteristic is correctly mapped into the DM tree 3. DM client is able to successfully establish a DM session with the DM server indicated in the bootstrap message

7.18 DeviceManagement-v1.2-int-015c

Test Case Id	DeviceManagement-v1.2-int-015c
Test Object	Client and Smart Card device
Test Case Description	Purpose of this test is to check that a Device Management client supports bootstrap from the Smart Card using the Client Provisioning profile when each of the provisioning files (i.e. EF_Bootstrap, EF_Config1 and EF_Config2) contain a different w7 APPLICATION characteristic.
Specification Reference	[DMBOOT] Section 5.3 [PROVSC]
SCR Reference	DM-BOOT-C-001 DM-BOOT-C-003 DM-BOOT-C-004 DM-BOOT-C-005 DM-BOOT-C-006 DM-BOOT-C-008 DM-BOOT-C-009
Preconditions	• A DM client to be configured supporting Client Provisioning Profile. • A Smart Card with bootstrap information containing one w7 APPLICATION characteristic per provisioning file (Reference content stored in the Smart Card: CP_Prov_doc_1.xml for EF_Bootstrap, EF_Config1, and for EF_Config2). No DM Profile files (e.g. EF_DM_Bootstrap) are present in the Smart Card. Note: Content of the different CP_Prov_doc_1.xml files needs to include minimal changes to allow differentiation (e.g. server name parameter can be set as "SERVERNAME" in EF_Bootstrap, "servername" in EF_Config1 and "ServerName" in EF_Config2).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Procedure	1. Insert a smart card in the handset. 2. Select to configure the terminal from the Smart Card if necessary. 3. On the client, select to save the information read from the Smart Card on the client if necessary. 4. Check that all the bootstrap information is saved in the client. 5. Check that the client can use any of the w7 characteristics from the bootstrap information
Pass-criteria	1. The client is able to read all the bootstrap configuration correctly from the smart card. 2. The bootstrap information contained in each of the w7 application characteristics is correctly mapped into the DM tree 3. DM client is able to successfully establish a DM session with at least one of the DM server(s) indicated in the bootstrap message.

7.19 DeviceManagement-v1.2-int-016

Test Case Id	DeviceManagement-v1.2-int-016
Test Object	Client and Smart Card device
Test Case Description	Purpose of this test is to check that a Device Management client supports bootstrap from a Smart Card using the Device Management Profile and WBXML encoded TNDS objects for the bootstrap information
Specification Reference	[DMBOOT] Section 5.4 [DMBOOT] Appendix D
SCR Reference	DM-BOOT-C-002 DM-BOOT-C-006 DM-BOOT-C-007 DM-BOOT-C-008 DM-BOOT-C-009 DM-BOOT-C-010
Preconditions	• A DM client to be configured supporting the DM profile. • A Smart Card with DM bootstrap information (DM Account and Connectivity Information) contained in WBXML encoded TNDS object (Reference content stored in the Smart Card: TNDS.xml). No CP Profile files (e.g. EF_Bootstrap, EF_Config1, EF_Config2) are present in the Smart Card.
Test Procedure	1. Insert a smart card in the handset. 2. Check that the bootstrap information is saved in the client. 3. Check that the client can use the bootstrap information
Pass-criteria	1. The client is able to read the bootstrap configuration correctly from

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	the smart card. 2. The bootstrap information contained is correctly mapped to the DM tree 3. The device is correctly configured according to bootstrap information from the smart card 4. DM client is able to successfully establish a DM session with the DM server indicated in the bootstrap message

7.20 DeviceManagement-v1.2-int-016b

Test Case Id	DeviceManagement-v1.2-int-016b
Test Object	Client and Smart Card device (DM Profile focus)
Test Case Description	Purpose of this test is to check that a Device Management client supports bootstrap from a Smart Card using the Device Management Profile and WBXML encoded TNDS objects for the bootstrap information; either because a) priority is given to the DM Profile or b) the device only supports the DM Profile.
Specification Reference	▪ For DM Profile: [DMBOOT] Section 5.4 [DMBOOT] Appendix D ▪ For CP Profile: [DMBOOT] Section 5.3 [PROVSC]
SCR Reference	DM-BOOT-C-001 DM-BOOT-C-002 DM-BOOT-C-003 DM-BOOT-C-004 DM-BOOT-C-005 DM-BOOT-C-006 DM-BOOT-C-007 DM-BOOT-C-008 DM-BOOT-C-009 DM-BOOT-C-010
Preconditions	• A DM client to be configured supporting: a) Either the DM profile and the CP Profile or b) DM Profile only. • A Smart Card personalized with: 1. DM bootstrap information (DM Account and

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Connectivity Information) contained in WBXML encoded TNDS object (Reference content stored in the Smart Card: TNDS.xml) 2. CP bootstrap information (w7 APPLICATION characteristic) contained in EF_Bootstrap (Reference content stored in the Smart Card: CP_Prov_doc_1.xml). EF_Config1 and EF_Config2 must be empty. 3. DM Account and w7 APPLICATION must have some minor difference in order to determine which one has effectively used by the client. • DM bootstrap information is given priority in the EF_ODF.
Test Procedure	1. Insert a smart card in the handset. 2. Check that the bootstrap information is saved in the client. 3. Check that the client can use the bootstrap information
Pass-criteria	1. The client is able to read the bootstrap configuration correctly from the smart card. 2. The bootstrap information contained is correctly mapped to the DM tree 3. The device is correctly configured according to bootstrap information from the DM Profile in the smart card 4. DM client is able to successfully establish a DM session with the DM server indicated in the DM bootstrap message

7.21 DeviceManagement-v1.2-int-016c

Test Case Id	DeviceManagement-v1.2-int-016c
Test Object	Client and Smart Card device (CP Profile focus)
Test Case Description	Purpose of this test is to check that a Device Management client supports bootstrap from a Smart Card using the Client Provisioning Profile and w7 APPLICATION characteristic because a) priority is given to the CP Profile or b) the device only supports the CP Profile.
Specification Reference	▪ For DM Profile: [DMBOOT] Section 5.4 [DMBOOT] Appendix D ▪ For CP Profile: [DMBOOT] Section 5.3 [PROVSC]
SCR Reference	DM-BOOT-C-001 DM-BOOT-C-002 DM-BOOT-C-003

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	DM-BOOT-C-004 DM-BOOT-C-005 DM-BOOT-C-006 DM-BOOT-C-007 DM-BOOT-C-008 DM-BOOT-C-009 DM-BOOT-C-010
Preconditions	• A DM client to be configured supporting: a) Either the DM profile and the CP Profile or b) CP Profile only. • A Smart Card personalized with: 1. DM bootstrap information (DM Account and Connectivity Information) contained in WBXML encoded TNDS object (Reference content stored in the Smart Card: TNDS.xml) 2. CP bootstrap information (w7 APPLICATION characteristic) contained in EF_Bootstrap (Reference content stored in the Smart Card: CP_Prov_doc_1.xml). EF_Config1 and EF_Config2 must be empty. 3. DM Account and w7 APPLICATION must have some minor difference in order to determine which one has effectively used by the client. • CP bootstrap information is given priority in the EF_ODF.
Test Procedure	1. Insert a smart card in the handset. 2. Check that the bootstrap information is saved in the client. 3. Check that the client can use the bootstrap information
Pass-criteria	1. The client is able to read the bootstrap configuration correctly from the smart card. 2. The bootstrap information contained is correctly mapped to the DM tree 3. The device is correctly configured according to bootstrap information from the CP Profile in the smart card 4. DM client is able to successfully establish a DM session with the DM server indicated in the CP bootstrap information

7.22 DeviceManagement-v 1.2-int-017

Test Case Id	DeviceManagement-v 1.2-int-017
Test Object	Client and Smart Card device (removal of DM Profile)

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Case Description	Purpose of this test is to check that a Client removes from the DM tree the account information for a DM Server previously bootstrapped from the Smart Card when that information is no longer present in the Smart Card
Specification Reference	[DMBOOT] Section 5.4.6.
SCR Reference	DM-BOOT-C-002 DM-BOOT-C-006 DM-BOOT-C-007 DM-BOOT-C-008 DM-BOOT-C-009 DM-BOOT-C-010

Preconditions	• A DM client supporting bootstrap from the Smart Card using the Device Management profile
	• Two Smart Cards containing only DM bootstrap information
	  • Smart Card A: DM Account and Connectivity Information contained in WBXML encoded TNDS object (Reference content: TNDS.xml)
	  • Smart Card B: DM Account and Connectivity Information contained in WBXML encoded TNDS object (Reference content: A small but identifiable variation of TNDS.xml)

Test Procedure	6. Introduce Smart Card A in the device 7. Verify that the DM client is provisioned with the corresponding account information contained in Smart Card A 8. Remove Smart Card A from the terminal. 9. Introduce Smart Card B (with different bootstrap information) in the device
Pass-criteria	The DM client should remove from the DM management tree the account information corresponding to the DM server bootstrapped from Smart Card A.

7.23 DeviceManagement-v 1.2-int-017a

Test Case Id	DeviceManagement-v 1.2-int-017a
Test Object	Client and Smart Card device (removal of CP Profile)
Test Case Description	Purpose of this test is to check that a Client removes from the DM tree the account information for a DM Server previously bootstrapped from the Smart Card when that information is no longer present in the Smart Card
Specification Reference	[DMBOOT] Section 5.3.5.1

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

SCR Reference	DM-BOOT-C-001 DM-BOOT-C-003 DM-BOOT-C-004 DM-BOOT-C-005 DM-BOOT-C-006 DM-BOOT-C-008 DM-BOOT-C-009
Preconditions	• A DM client supporting bootstrap from the Smart Card using the Client Provisioning profile. • Two Smart Cards containing only CP bootstrap information: 1. Smart Card A: w7 APPLICATION characteristic in EF_Bootstrap (Reference content: CP_Prov_doc_1.xml). 2. Smart Card B: w7 APPLICATION characteristic in EF_Bootstrap (Reference content: CP_Prov_doc_1.xml). Note: Content of the different CP_Prov_doc_1.xml files needs to include minimal changes to allow differentiation (e.g. server name parameter can be set as "SERVERNAME" for Smart Card A, "servername" for Smart Card B). • In both Smart Cards, EF_Config1 and EF_Config2 must be empty.
Test Procedure	• Introduce a Smart Card A containing bootstrap information in the device • Verify that the DM client is provisioned with the corresponding account information contained in the Smart Card A • Remove the Smart Card A from the terminal. • Introduce Smart Card B (with different bootstrap information) in the terminal
Pass-criteria	The DM client should remove from the DM management tree the account information corresponding to the DM server bootstrapped from Smart Card A

7.24 DeviceManagement-v 1.2-int-018

Test Case Id	DeviceManagement-v 1.2-int-018
Test Object	DM client and DM server
Test Case Description	Purpose of this test is to check that a DM client supports server initiated bootstrap using the DM profile, WBXML encoded TNDS objects and the Inbox, under transport neutral security when the transport method used does not have appropriate security. NETWORKID is used.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Specification Reference	[TS-DM-Bootstrap] Section 5.4.1 [TS-DM-Security] Section 5.7.2.3	
SCR Reference	DM-BOOT-C-002 DM-BOOT-C-010 DM-BOOT-C-011 DM-BOOT-S-002 DM-BOOT-S-003	DM-SEC-C-022 DM-SEC-S-026
Preconditions	• A DM Client to be bootstrapped supporting DM Profile • A DM Server supporting DM profile with bootstrap information (DM account and connectivity information) (Reference Content stored in the server: TNDS.xml) • A transport without appropriate security mechanisms for bootstrapping a device securely (e.g SMS, USSD) supported by both the DM client and DM server. • Client and server support for NETWORKID security mechanism	
Test Procedure	1. DM server sends out the bootstrap message along with the HMAC calculated using NETWORKID. 2. On the client select to accept the incoming bootstrap message if necessary. 3. Check that the device is bootstrapped with the bootstrap information sent by the DM server.	
Pass-criteria	1. The server is authenticated by the client 2. DM client processes correctly the bootstrap message 3. Bootstrap information is successfully mapped to DM tree and DM client is correctly configured. 4. DM client is able to successfully establish a DM session with the server that initiated the bootstrap	

7.25 DeviceManagement-v 1.2-int-019

Test Case Id	DeviceManagement-v 1.2-int-019
Test Object	DM client and DM server
Test Case Description	Purpose of this test is to check that a DM client supports server initiated bootstrap using the DM profile WBXML encoded TNDS objects and the Inbox under transport neutral security when the transport method used does not have appropriate security. USERPIN is used.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Specification Reference	[TS-DM-Bootstrap] Section 5.4.1 [TS-DM-Security] Section 5.7.2.3	
SCR Reference	DM-BOOT-C-002 DM-BOOT-C-010 DM-BOOT-C-011 DM-BOOT-S-002 DM-BOOT-S-003	DM-SEC-C-023 DM-SEC-S-027
Preconditions	• A DM Client to be bootstrapped supporting DM Profile • A DM Server supporting DM profile with bootstrap information (DM account and connectivity information) (Reference Content stored in the server: TNDS.xml) • A transport without appropriate security mechanisms for bootstrapping a device securely (e.g SMS, USSD) supported by both the DM client and DM server • Client and server support for USERPIN security mechanism	
Test Procedure	1. DM server sends out the bootstrap message along with the HMAC, calculated using USERPIN. 2. On the client select to accept the incoming bootstrap message if necessary. 3. Check that the device is bootstrapped with the bootstrap information sent by the DM server.	
Pass-criteria	1. The server is authenticated by the client 2. DM client processes correctly the bootstrap message 3. Bootstrap information is successfully mapped to DM tree and DM client is correctly configured. 4. DM client is able to successfully establish a DM session with the server that initiated the bootstrap	

7.26 DeviceManagement-v 1.2-int-020

Test Case Id	DeviceManagement-v 1.2-int-020
Test Object	DM client and DM server
Test Case Description	Purpose of this test is to check that a DM client supports server initiated bootstrap using the DM profile, WBXML encoded TNDS objects and the Inbox, under transport neutral security when the transport method used does not have appropriate security. USERPIN_NETWORKID is used.
Specification Reference	[TS-DM-Bootstrap] Section 5.4.1

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	[TS-DM-Security] Section 5.7.2.3	
SCR Reference	DM-BOOT-C-002 DM-BOOT-C-010 DM-BOOT-C-011 DM-BOOT-S-002 DM-BOOT-S-003	DM-SEC-C-021 DM-SEC-S-025
Preconditions	• A DM Client to be bootstrapped supporting DM Profile • A DM Server supporting DM profile with bootstrap information (DM account and connectivity information) (Reference Content stored in the server: TNDS.xml) • A transport without appropriate security mechanisms for bootstrapping a device securely (e.g SMS, USSD) supported by both the DM client and DM server. • Client and server support for USERPIN_NETWORKID security mechanism	
Test Procedure	4. DM server sends out the bootstrap message along with the HMAC, calculated using USERPIN_NETWORKID. 5. On the client select to accept the incoming bootstrap message if necessary. 6. Check that the device is bootstrapped with the bootstrap information sent by the DM server.	
Pass-criteria	4. The server is authenticated by the client DM 5. DM client processes correctly the bootstrap message 6. Bootstrap information is successfully mapped to DM tree and DM client is correctly configured 7. DM client is able to successfully establish a DM session with the server that initiated the bootstrap	

7.27 DeviceManagement-v1.2-int-021

Test Case Id	DeviceManagement-v1.2-int-021
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with UI Display Alert.
Specification Reference	[DMPRO] Chapter 10.2.1 [DMREPU] Chapter 7
SCR Reference	DM-PRO-UI-C-001 Executing Display Alert

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	DM-PRO-UI-S-001 Sending Display Alert
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings 2. In the server, configure it to send the UI Display Alert to the client with a message ■ With this UI Alert, the user interaction options (MAXDT, MINDT, DR, MAXLEN, IT, ET) MAY be specified by the server. 3. Establish the connection from the client. 4. The message is displayed on the client device. ■ The client MUST ignore all interaction options it does not understand. 5. Complete the DM session. 6. Check the DM session goes without any errors.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Client received the proper UI Display Alert from the server.

7.28 DeviceManagement-v1.2-int-022

Test Case Id	DeviceManagement-v1.2-int-022
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with UI Confirmation Alert.
Specification Reference	[DMPRO] Chapter 10.2.2 [DMREPU] Chapter 7
SCR Reference	DM-PRO-UI-C-002 Executing Confirm or Reject Alert DM-PRO-UI-S-002 Sending Confirm or Reject Alert
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings 2. In the server, configure it to send the UI Confirmation Alert to the client with a message ■ With this UI Alert, the user interaction options (MAXDT, MINDT, DR, MAXLEN, IT, ET) MAY be specified by the server. 3. Establish the connection from the client. 4. The message is displayed on the client device.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	■ The client MUST ignore all interaction options it does not understand. 5. Depending on the client action, the status in the response will be 200 (Yes), 304 (No), or 214 (Cancel). 6. Note: Optionally, the server could send the Alert within a Sequence or Atomic to verify the status returned for commands when the user action is No. 7. Complete the DM session. 8. Check the DM session goes without any errors.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Client received the proper UI Confirmation Alert from the server.

7.29 DeviceManagement-v1.2-int-023

Test Case Id	DeviceManagement-v1.2-int-023
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with UI Text Input Alert.
Specification Reference	[DMPRO] Chapter 10.2.3 [DMREPU] Chapter 7
SCR Reference	DM-PRO-UI-C-003 Executing Text Input Alert DM-PRO-UI-S-003 Sending Text Input Alert
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings 2. In the server, configure it to send the UI Text Input Alert to the client with a message ■ With this UI Alert, the user interaction options (MAXDT, MINDT, DR, MAXLEN, IT, ET) MAY be specified by the server. 3. Establish the connection from the client. 4. The message is displayed on the client device and the user is allowed to enter some text. ■ The client MUST ignore all interaction options it does not understand. 5. The text is returned to the server. 6. Complete the DM session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	7. Check the DM session goes without any errors.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Client received the proper UI Text Input Alert from the server.

7.30 DeviceManagement-v1.2-int-024

Test Case Id	DeviceManagement-v1.2-int-024
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with UI Single Choice Alert.
Specification Reference	[DMPRO] Chapter 10.2.4 [DMREPU] Chapter 7
SCR Reference	DM-PRO-UI-C-004 Executing Single Choice Alert DM-PRO-UI-S-004 Sending Single Choice Alert
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings 2. In the server, configure it to send the UI Single Choice Alert to the client with a message and several choices. ■ With this UI Alert, the user interaction options (MAXDT, MINDT, DR, MAXLEN, IT, ET) MAY be specified by the server. 3. Establish the connection from the client. 4. The message is displayed on the client device and the user is allowed to select one item from the supplied choices. ■ The client MUST ignore all interaction options it does not understand. 5. The index of the selected item (1 based) is returned to the server. 6. Complete the DM session. 7. Check the DM session goes without any errors.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Client received the proper UI Single Choice Alert from the server.

7.31 DeviceManagement-v1.2-int-025

Test Case Id	DeviceManagement-v1.2-int-025

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with UI Multiple Choice Alert.
Specification Reference	[DMPRO] Chapter 10.2.4 [DMREPU] Chapter 7
SCR Reference	DM-PRO-UI-C-005 Executing Multiple Choice Alert DM-PRO-UI-S-005 Sending Multiple Choice Alert
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings 2. In the server, configure it to send the UI Multiple Choice Alert to the client with a message and several choices. ■ With this UI Alert, the user interaction options (MAXDT, MINDT, DR, MAXLEN, IT, ET) MAY be specified by the server. 3. Establish the connection from the client. 4. The message is displayed on the client device and the user is allowed to select one or more items from the supplied choices. ■ The client MUST ignore all interaction options it does not understand. 5. The indexes of the selected items (1 based) are returned to the server. 6. Complete the DM session. 7. Check the DM session goes without any errors.
Pass-Criteria	1. DM session runs through without any communication problem. 2. Client received the proper UI Multiple Choice Alert from the server.

7.32 DeviceManagement-v1.2-int-026

Test Case Id	DeviceManagement-v1.2-int-026
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the server reading subtree structure without data from part of the management tree.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 8
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving ‘Get’

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	DMREPPRO-PCE-S-008 Support for sending 'Get'
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings. 2. In the server, configure it to perform a Get command on an interior node to read a part of the subtree structure. Note: If this feature is not supported, the client should return status 406 (Optional feature not supported). ■ Example: In the server choose to receive the subtree structure of Device Detail: Get ./SyncML/DMAcc?list=Struct 3. Establish the connection from the client. 4. Complete the DM session. 5. Check the DM session goes without any errors. 6. Check the DM server sent the proper response to the client.
Pass-Criteria	1. Client didn't receive any DM data from the server. 2. Server received part of the subtree structure. 3. DM session runs through without any communication problem. 4. Synchronisation runs through with a basic DM authentication.

7.33 DeviceManagement-v1.2-int-027

Test Case Id	DeviceManagement-v1.2-int-027
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show compliance with the server reading subtree structure and data from part of the management tree.
Specification Reference	[DMREPU] Chapter 6.6.7 [DMTND] Chapter 8
SCR Reference	DMREPPRO-PCE-C-008 Support for receiving 'Get' DMREPPRO-PCE-S-008 Support for sending 'Get'
Preconditions	None.
Test Procedure	1. Use client default authentication and connection settings. 2. In the server, configure it to perform a Get command on an interior node to read a part of the subtree structure and data. Note: If this feature is not supported, the client should return status 406 (Optional feature not supported).

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	■ Example: In the server choose to receive the subtree structure and data of Device Detail: Get ./SyncML/DMAcc?list=StructData 3. Establish the connection from the client. 4. Complete the DM session. 5. Check the DM session goes without any errors. 6. Check the DM server sent the proper response to the client.
Pass-Criteria	1. Client didn't receive any DM data from the server. 2. Server received part of the subtree structure and data. 3. DM session runs through without any communication problem. 4. Synchronisation runs through with a basic DM authentication.

7.34 DeviceManagement-v1.2-int-028

Test Case Id	DeviceManagement-v1.2-int-028
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to verify creation of new Application Setting in client using DM server
Specification Reference	[DMREPU] Chapter 6.6.7
SCR Reference	
Preconditions	• One Access Point exists in client
Test Procedure	1. Establish the connection from the client. 2. Request the supported fields of the Application from the client. 3. Depending on the server functionality, fill the supported fields with new Application server data and send the Application Settings to the client. The Application server data can also be filled in server database or file before the connection to the client is established. 4. Complete the DM session. 5. Test the results by making a session using existing Access Point and new Application Settings set by DM session.
Pass-Criteria	1. Server accepts incoming call. 2. Client sends supported fields. 3. Server sends new settings. 4. Connection closed. 5. Connection to the Application server established.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

7.35 DeviceManagement-v1.2-int-029

Test Case Id	DeviceManagement-v1.2-int-029
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to verify modification of Application Settings in client using DM server.
Specification Reference	[DMREPU] Chapter 6.6.11
SCR Reference	
Preconditions	• Application Settings exist in client. • Have another Application Settings available.
Test Procedure	1. Establish the connection from the client. 2. Request the supported fields of the Application from the client. 3. Depending on the server functionality, modify the supported fields with new Application server data and send the settings to the client. The Application server data can also be filled in server database or file before the connection to the client is established. 4. Complete the DM session. 5. Test the results by making a session using existing Access Point and new Application Settings set by DM session.
Pass-Criteria	1. Server accepts incoming call. 2. Client sends supported fields. 3. Server sends the new settings. 4. Connection closed. 5. Connection to the Application server established.

7.36 DeviceManagement-v1.2-int-030

Test Case Id	DeviceManagement-v1.2-int-030
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to verify deletion of Application Settings in client using DM server.
Specification Reference	[DMREPU] Chapter 6.6.5
SCR Reference	
Preconditions	• Application Settings exist in client. • Have another Application Settings available.
Test Procedure	1. Establish the connection from the client.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	2. Request the supported fields of the Application from the client. 3. Depending on the server functionality, delete the Application Settings from the client. The Application server data can also be filled in server database or file before the connection to the client is established. 4. Complete the DM session. 5. Check that no SyncML settings exist in the client.
Pass-Criteria	1. Server accepts incoming call. 2. Client sends supported fields. 3. Server deletes the Access Point. 4. Connection closed. 5. No Access Point in client.

7.37 DeviceManagement-v 1.2-int-031

Test Case Id	DeviceManagement-v 1.2-int-031
Test Object	Client and Server device
Test Case Description	Purpose of this test is to check that a DM client supports the Inbox object and that the information in the Inbox is correctly mapped onto the DM tree
Specification Reference	[DMSTDOBJ] Section 5.3.4
SCR Reference	DM-STDOBJ-C-004 DM-STDOBJ-S-004
Preconditions	• A DM client and DM server supporting the Inbox object
Test Procedure	1. The DM Client and Server establish a DM session 2. The DM server sends an "ADD" command for a Standardized Management Object (e.g. DMAcc) with the URI: "./Inbox"
Pass-criteria	1. The device, using the management object identifier, resolves the correct location in the management tree to add that Management Object initially stored in the "./Inbox"

7.38 DeviceManagement-v 1.2-int-032

Test Case Id	DeviceManagement-v 1.2-int-032
Test Object	Client and Server device
Test Case Description	Purpose of this test is to check that a Device Management client rejects

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Get operations from a DM server on the "./Inbox" node.
Specification Reference	[DMSTDOBJ] Section 5.3.4
SCR Reference	DM-BOOT-C-011 DM-STDOBJ-C-004
Preconditions	• A DM client supporting the Inbox object . • A DM server interested in retrieving information from the "./Inbox" of the DM client tree.
Test Procedure	1. The DM server establishes a DM session with the DM client 2. The DM server performs a *Get* operation on the Inbox node in the DM client.
Pass-criteria	1. The client does not permit a *Get* operation on the "./Inbox" to be done from any server. 2. The DM Client returns the status code "*Command not allowed*" (405) in response to a Get which targets "./Inbox" or any direct or indirect child node of "./Inbox".

7.39 DeviceManagement-v 1.2-int-033

Test Case Id	DeviceManagement-v 1.2-int-033
Test Object	DM 1.1.2 client and DM 1.2 server
Test Case Description	Purpose of this test is to verify backwards compatibility between a DM 1.2 server and a DM 1.1.2 client.
Specification Reference	[OMA-SyncML-DMProtocol-V1_1_2] Section 8.3
SCR Reference	[OMA-SyncML-DMConReqs-V_1_1_2] Section 6
Preconditions	• A DM 1.2 server • A bootstrapped DM 1.1.2 client
Test Procedure	4. Establish the connection from the client 5. In the server, configure it to perform a Get on a specific node in the client. Example: Get ./DevDetail/URI 6. Client returns a data value that includes leaf node names:. Example: Client returns a data value that includes the leaf node names MaxDepth, MaxTotLen and MaxSegLen
Pass-criteria	1. The DM Session is correctly established between the DM server and the DM client 2. The DM 1.2 server responds to the Pkg#1 from the DM 1.1.2 client by using the protocol version specified by DM 1.1.2 enabler release for the remainder of that session.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	3. The Server receives the proper information from the client as response to the Get command

* If this test case is successfully passed, the DM server should act as a DM 1.1.2 server (although it is supporting DM 1.2 too) in the rest of the test session, so the applicable test cases in this scenario are those contained in the DM 1.1.2 Enabler Test Specification

7.40 DeviceManagement-v 1.2-int-034

Test Case Id	DeviceManagement-v1.2-int-034
Test Object	Client and Server device
Test Case Description	Purpose of this test case is to check if the Test Object supports implicit addition of parent interior nodes for an addition of a child node whose valid parent/parents does not exist in the DM Tree
Specification Reference	[DMREPU] Chapter 6.6.1 [DMTND] Chapter 7
SCR Reference	N/A
Preconditions	• Parent Node of the child node to be added does not exist. • An stablished DM session between DM Server and DM Client. • DM Server has sufficient rights to add a node on the DM tree.
Test Procedure	1- DM Server sends an add command including the complete URI of the child node. 2- DM Clients Returns a 200 Status code. 3- DM Server sends a get command on the newly added node. 4- DM Client returns 200 Status Code. 5- DM Client returns a result code with the value of the Node.
Pass-Criteria	The Test Object MUST return a 200 status code and the value of the newly Added child node.

7.41 DeviceManagement-v 1.2-int-035

Test Case Id	DeviceManagement-v1.2-int-035
Test Object	Client and Server device
Test Case Description	To check if the Test Object can handle a Get with 'list=TNDS'. DM Server issues a Get on './DevDetail?list=TNDS+ACL+Format+Value'
Specification Reference	[DMREPU] Chapter 6.6.7

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	[DMTND] Chapter 8 and Appendix B
SCR Reference	DMTND-Prop-C-012 Support Get? list=TNDS DMTND-Prop-S-012 Support Get?list=TNDS
Preconditions	• An stablished DM session between DM Server and DM Client. • DM Server has sufficient rights on /DevDetail. • /DevDetail node exists on the DM Client Tree and contains some sub nodes.
Test-Procedure	1-DM Server issues a Get on './DevDetail?list=TNDS+ACL+Format+Value' 2- DM Client returns 200 Status Code. 3- DM Client returns a result code with the TNDS file fo the contents of the DevInfo (including ACLs, Format and Value)
Pass-Criteria	DM Server should receive valid results in TNDS format (including ACLs, Format and Value) .

7.42 DeviceManagement-v 1.2-int-036

Test Case Id	DeviceManagement-v1.2-int-036
Test Object	Client and Server device
Test Case Description	To check if the Test Object can handle the copy command, It would be followed by a Get command on both the URI
Specification Reference	[DMREPU] Chapter 6.6.4 and Annex B.
SCR Reference	DMREPPRO-PCE-C-005 Support for receiving 'Copy' command DMREPPRO-PCE-S-005 Support for receiving 'Copy' command
Preconditions	• An stablished DM session between Test Tool and DM Client. • DM Server has sufficient rights on target node. • DM Server has sufficient rights on source node.
Test-Procedure	1. DM Server issues a copy to the DM Client. 2. DM Client returns 200 Status Code. 3. DM Server issues a Get on target node. 4. DM Client returns 200 Status Code. 5. DM Client returns result code. 6. DM Server issues a Get on source node. 7. DM Client returns 200 Status Code. 8. DM Client returns result code.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Pass-Criteria	1. DM Client returs a status 200 code for the copy. 2. Results from the get commend at the source at the same as results for the get command at the target.

7.43 DeviceManagement-v 1.2-int-037

Test Case Id	DeviceManagement-v1.2-int-037
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to show capability of correlator use
Specification Reference	[DMREPU] Chapter 6.3 and Chapter 6.6.2
SCR Reference	DMREPPRO-PCE-C-007 Support for 'Exec' DM-PRO-GAlert-C-004 DMREPPRO-PCE-S-007 Support for Sending 'Exec' DM-PRO-S-009 Support of 'Generic Alert' DMREPPRO-DDE-S-001 Support for sending 'Correlator' DMREPPRO-DDE-S-002 Support for receiving 'Correlator' DM-PRO-Galert-S-001 Support for receiving, parsing and send Status Back to Client
Preconditions	• An established DM session between DM Client and DM Server. • A node capable of receiving an exec node exists in the DM Tree of the DM Client. • DM Server has sufficient rights to exec a node on that node of the DM Client • Client is capable of supporting Correlator.
Test Procedure	1. DM Server sends an Exec command to the node with a Correlator. 2. DM Client returns a Generic Alert including the same correlator.
Pass-Criteria	1. DM server is able to send exec including the correlator. Client returns 200 for a valid Exec command. 2. DM Server returns a status code 200 or 202 in response to the Generic alert.. 3. Correlator of Exec node and Generic Alert are the same

7.44 DeviceManagement-v 1.2-int-038

Test Case Id	DeviceManagement-v1.2-int-038
Test Object	Client and Server device

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Case Description	Purpose of this verification is to show capability to add a serialized management object to the DM tree
Specification Reference	[DMTNDS] Chapters 5 & 6
SCR Reference	DM-TNDS-C-002 Support of Decoding a TNDS object DM-TNDS-S-001 Support of Encoding a TNDS object
Preconditions	• A established DM session between server and client • A node in the DM tree in which the DM server has sufficient rights to perform an Add command
Test Procedure	3. DM Server sends an Add command with a TNDS serialized management object (for instance DMAcc) to a specific node in the tree 4. DM Server performs a Get for one of the child nodes included under the root of the added serialized Management Object (for instance if DMAcc is used it could be DMAcc/ServerId)
Pass-Criteria	4. Client responds to the Add command with a 200 status 5. The serialized MO is correctly mapped into the device DM tree. 6. Client responds to the Get command with a 200 status and the correct value for the requested node

7.45 DeviceManagement-v1.2-int-039

Test Case Id	DeviceManagement-v1.2-int-039
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to test transport layer authentication using TLS 1.0
Specification Reference	[DMSEC] Chapter 5.3 and 5.5.1.1
SCR Reference	DM-SEC-C-003 Support for transport layer authentication DM-SEC-C-013 Identifying that the server is using TLS1.0 or SSL3.0 DM-SEC-C-014 Support for TLS DM-SEC-C-016 Supporting at least one of the cipher suites TLS_RSA_WITH_AES _128_CBC_SHA-1, TLS_RSA_WITH_3DES_EDE_CBC_SHA and TLS_RSA_WITH_RC4_128_SHA DM-SEC-S-002 Support for client authentication at the transport layer DM-SEC-S-015 Support for TLS 1.0 [TLS] DM-SEC-S-018 Using TLS DM-SEC-S-020 Supporting all three cipher suites TLS_RSA_WITH_AES_128_CBC_SHA-1, TLS_RSA_WITH_3DES_EDE_CBC_SHA and

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	TLS_RSA_WITH_RC4_128_SHA
Preconditions	DM Client and Server support DM over the same transport protocol (HTTP or other) Credentials / certificates necessary to perform authentication have been provisioned in Server and Device prior to the test.
Test Procedure	5l. DM client initiates a session with the DM Server requesting transport layer authentication using TLS and indicating cipher settings. 5m. DM Server authenticates the DM client and sends the information the DM client needs to authenticate it. 5n. DM client authenticates the DM Server and the TLS session is established. 5o. DM session is established between DM client and DM Server
Pass-Criteria	7. DM Server authenticates the DM client 8. DM client authenticates the DM Server 9. DM session is correctly established.

7.46 DeviceManagement-v1.2-int-040

Test Case Id	DeviceManagement-v1.2-int-040
Test Object	Client and Server device
Test Case Description	Purpose of this verification is to test transport layer authentication using SSL 3.0
Specification Reference	[DMSEC] Chapter 5.3 and 5.5.1.1
SCR Reference	DM-SEC-C-003 Support for transport layer authentication DM-SEC-C-013 Identifying that the server is using TLS1.0 or SSL3.0 DM-SEC-C-015 Support for SSL 3.0 DM-SEC-C-017 Support for at least one of SSL_RSA_WITH_RC4_128_SHA and SSL_RSA_WITH_3DES_EDE_CBC_SHA DM-SEC-S-002 Support for client authentication at the transport layer DM-SEC-S-016 Support for SSL3.0 [SSL3.0] DM-SEC-S-019 Using SSL3.0 DM-SEC-S-021 Support for both of SSL_RSA_WITH_RC4_128_SHA and SSL_RSA_WITH_3DES_EDE_CBC_SHA
Preconditions	DM Client and Server support DM over the same transport protocol (HTTP or other) Credentials / certificates necessary to perform authentication have been provisioned in Server and Device prior to the test.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Test Procedure	1. DM client initiates a session with the DM Server requesting transport layer authentication using SSL 3.0 and indicating cipher settings. 2. DM Server authenticates the DM client and sends the information the DM client needs to authenticate it. 3. DM client authenticates the DM Server and the SSL session is established. 4. DM session is established between DM client and DM Server
Pass-Criteria	1. DM Server authenticates the DM client 2. DM client authenticates the DM Server 3. DM session is correctly established.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft/Candidate Version 1.2 History

Document Identifier	Date	Sections	Description
Draft Versions: OMA-ETS-DM-V1_2	01 Mar 2006		Draft ETS baseline Agreed in the 01 March 2006 CC (OMA-IOP-PRC-2006-0036R01)
	03 May 2006	7.14 7.31 7.27, 7.28, 7.29 5.26, 6.21, 7.32, 7.33	OMA-IOP-PRC-2006-0064-DM1.2.-Bootstrap-Test-Cases-improvement incorporated OMA-IOP-PRC-2006-0065-DM-1.2.TNDS-test-cases incorporated OMA-IOP-PRC-2006-0066-Remove-SCTS-references-in-DM-1.2.-interoperability-test-cases incorporated OMA-IOP-PRC-2006-0067R01-DM-1.2.-Transport-Layer-Security-Test-Cases incorporated
	10 May 2006	5.12.3.	OMA-IOP-PRC-2006-0074-CR-DM1_2-DMAcc-test-case incorporated
	11 May 2006	n/a	Agreed by IOP WG prepared for TP approval
Candidate Versions: OMA-ETS-DM-V1_2	24 May 2006	n/a	Status changed to Candidate by TP R&A (2006-05-17 to 2006-05-23) OMA-TP-2006-0192-ETS_INP_DM_1_2_for_Candidate_Approval
	05 Jul 2006	n/a	INT and interoperability removed from the filename and title page as the ETS covers both INT and CON test cases
	06 Jul 2006	n/a	The updated ETS, available as OMA-IOP-2006-0172 agreed in IOP. ETS prepared for notification OMA-TP-2006-0277-OMA-TP-2006-0277-OMA-ETS-DM_INT-V1_2_for_Notification.
	19 Apr 2007	App. C, all	Incorprated CR: OMA-IOP-PRC-2006-0077R04 Editorial change: Test case headings modified to match the template
		n/a	IOP WG Agreed
		all	ETS prepared for notification OMA-TP-2007-0113-INP_ETS_OMA_DM_v1_2_for_Notification.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Document Identifier	Date	Sections	Description
Draft Versions OMA-ETS-DM-V1_2	12 Jun 2007	5.1.1, 5.1.2 5.1.3, 5.2.1 5.3.1, 5.3.2 5.3.3, 5.3.4 5.4.1, 5.5.1 5.5.2, 5.5.3 5.6.1, 5.6.2 5.7.1, 5.8 5.9. , 5.12.3 5.13, 5.13.4 5.14, 5.15 5.16, 5.17 5.20, 5.22 5.23.1, 5.25 5.26 App C, C.1, C.4, C.5 App D, D.5, App E	Incorprated CR: OMA-IOP-PRC-2006-0052 OMA-IOP-PRC-2007-0016R03 OMA-IOP-PRC-2007-0025 OMA-IOP-PRC-2007-0026 OMA-IOP-PRC-2007-0028R02 OMA-IOP-PRC-2007-0030R01 OMA-IOP-PRC-2007-0032R01 OMA-IOP-PRC-2007-0043R01 OMA-IOP-PRC-2007-0044R01 OMA-IOP-PRC-2007-0045R01 OMA-IOP-PRC-2007-0046R01 OMA-IOP-PRC-2007-0049R01 OMA-IOP-PRC-2007-0050R01 OMA-IOP-PRC-2007-0051R01 OMA-IOP-PRC-2007-0052 OMA-IOP-PRC-2007-0055 OMA-IOP-PRC-2007-0057 OMA-IOP-PRC-2007-0058 OMA-IOP-PRC-2007-0061 OMA-IOP-PRC-2007-0062 OMA-IOP-PRC-2007-0063 OMA-IOP-PRC-2007-0064 OMA-IOP-PRC-2007-0065 OMA-IOP-PRC-2007-0066 OMA-IOP-PRC-2007-0076
	14 Jun 2007	n/a	IOP WG agreed, ETS prepared for candidate re-arppoval OMA-TP-2007-0280-INP_ETS_OMA_DM_v1_2_for_Candidate_Re_Approval
Candidate Version OMA-ETS-DM-V1_2	17 Jul 2007	n/a	Re-approval as Candidate (TP R&A 2007-07-04 to 2007-07-17) TP ref # OMA-TP-2007-0280R02-INP_ETS_OMA_DM_v1_2_for_Candidate_Re_Approval
Draft Version OMA-ETS-DM-V1_2	17 Mar 2008	5.3.2, 5.10.1	CRs incorporated: OMA-IOP-MEC-2008-0044 OMA-IOP-MEC-2008-0045 OMA-IOP-MEC-2008-0047
	22 Apr 2008	all	CR incorporated: OMA-IOP-MEC-0068R01
	15 May 2008	2.1, 7, App F	CR incorporated: OMA-IOP-MEC-0027R01
	18 Jun 2008	7.18, 7.23	CR incorporated: OMA-IOP-MEC-2008-0100
	04 Jul 2008	App C	CRs incorporated: OMA-IOP-MEC-2008-0101 OMA-IOP-MEC-2008-0046R01
Candidate Version OMA-ETS-DM-V1_2	18 Jul 2008	n/a	Status changed to Candidate by TP TP ref # OMA-TP-2008-0281-INP_DM_1.2_ETS_for_notification
Draft Version OMA-ETS-DM-V1_2	03 Sep 2008	5.15.1	CR incorporated: OMA-IOP-MEC-2008-0130R01
	07 Oct 2009	5.7.1, App B.1	CR incorporated: OMA-IOP-MEC-2009-0126 OMA-IOP-MEC-2009-0148R01
	30 Mar 2010	5.17.3	CR incorporated: OMA-IOP-MEC-2010-0020
Candidate Version OMA-ETS-DM-V1_2	06 May 2010	n/a	Status changed to Candidate by TP TP Ref# OMA-IOP-2010-0056-INP_DM_12_ETS_for_TPnotification
Draft Version OMA-ETS-DM-V1_2	08 Jul 2010	5.17.3	CR incorporated: OMA-IOP-MEC-2010-0052R01

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Document Identifier	Date	Sections	Description
Candidate Version OMA-ETS-DM-V1_2	19 Aug 2010	n/a	Status changed to Candidate by TP TP Ref# OMA-TP-2010-0381-INP_DM_12_ETS_for_notification

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix B. Reference Configuration Messages (Normative)

B.1 TNDS.xml

These reference document should be completed according to the specifics of the corresponding DM server being bootstrapped by substituting the bold text with the appropriate parameters for that server. Besides the nodes already marked as such, the nodes ToConRef and AppAuth are also optional and may not be included in the document.

```
<?xml version="1.0" encoding="UTF-8"?>
<SyncML  xmlns='SYNCML:SYNCML1.2'>
        <SyncHdr>
                <VerDTD>1.2</VerDTD>
                <VerProto>DM/1.2</VerProto>
                <SessionID>0</SessionID>
                <MsgID>0</MsgID>
                <Target>
                        <LocURI>./</LocURI>
                </Target>
                <Source>
                        <LocURI>http://www.operator.com/dm-server</LocURI>
                </Source>
        </SyncHdr>

        <SyncBody>
                <Add>
                        <CmdID>1</CmdID>
                        <Item>
                                <Target>
                                        <LocURI>./Inbox</LocURI>
                                </Target>
                                <Meta>
                                        <Format xmlns='syncml:metinf'>xml</Format>
                                        <Type xmlns='syncml:metinf'>
                                                application/vnd.syncml.dmtnds+xml
                                        </Type>
                                </Meta>
                                <Data>
                                        <![CDATA[
<MgmtTree xmlns='syncml:dmddf1.2'>
        <VerDTD>1.2</VerDTD>
        <!-- <Man>The device manufacturer</Man> -->
        <!-- <Mod>The device model</Mod> -->
        <Node>
                <NodeName>DMAcc</NodeName>
                <RTProperties>
                        <Format>
                                <node/>
```

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
                </Format>
        </RTProperties>

        <Node>
                <NodeName>AppID</NodeName>
                <RTProperties>
                        <Format>
                                <chr/>
                        </Format>
                        <Type><MIME>text/plain</MIME></Type>
                </RTProperties>
                <Value>org.openmobilealliance/1.0/w7</Value>
        </Node>

        <Node>
                <NodeName>ServerID</NodeName>
                <RTProperties>
                        <Format>
                                <chr/>
                        </Format>
                        <Type><MIME>text/plain</MIME></Type>
                </RTProperties>
                <Value>DM Server Identifier</Value>
        </Node>


        <Node>
                <NodeName>Name</NodeName>
                <RTProperties>
                        <Format>
                                <chr/>
                        </Format>
                        <Type><MIME>text/plain</MIME></Type>
                </RTProperties>
                <Value>Optional DM Server Displayable Name</Value>
        </Node>
        <Node>
                <NodeName>PrefConRef</NodeName>
                <RTProperties>
                        <Format>
                                <chr/>
                        </Format>
                        <Type><MIME>text/</MIME></Type>
                </RTProperties>
                <Value>Optional Reference to preferred connectivity information </Value>
        </Node>

    <Node>
```

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
                <NodeName>ToConRef</NodeName>
                <RTProperties>
                        <Format>
                                <node/>
                        </Format>
                </RTProperties>
        <Node>
                        <NodeName>Connectivity Reference Name</NodeName>
                        <RTProperties>
                                <Format>
                                        <node/>
                                </Format>
                        </RTProperties>
                        <Node>
                                <NodeName>ConRef</NodeName>
                                <RTProperties>
                                        <Format>
                                                <chr/>
                                        </Format>
                                        <Type><MIME>text/plain</MIME></Type>
                                </RTProperties>
                                <Value>Reference to Connectivity Information</Value>
                        </Node>
        </Node>
</Node>

<Node>
                <NodeName>AppAddr</NodeName>
                <RTProperties>
                        <Format>
                                <node/>
                        </Format>
                </RTProperties>
                <Node>
                        <NodeName>Management Server Address Name</NodeName>
                        <RTProperties>
                                <Format>
                                        <node/>
                                </Format>
                        </RTProperties>
                        <Node>
                                <NodeName>Addr</NodeName>
                                <RTProperties>
                                        <Format>
                                                <chr/>
                                        </Format>
                                        <Type><MIME>text/plain</MIME></Type>
                                </RTProperties>
                                <Value>Management Server Address</Value>
```

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
                    </Node>
                    <Node>
                        <NodeName>AddrType</NodeName>
                        <RTProperties>
                            <Format>
                                <chr/>
                            </Format>
                            <Type><MIME>text/plain<MIME></Type>
                        </RTProperties>
                        <Value>URI, IPv4 or IPv6</Value>
                    </Node>
                    <Node>
                        <NodeName>Port</NodeName>
                        <RTProperties>
                            <Format>
                                <node/>
                            </Format>
                        </RTProperties>
                        <Node>
                            <NodeName>Port Name</NodeName>
                            <RTProperties>
                                <Format>
                                    <node/>
                                </Format>
                            </RTProperties>
                            <Node>
                                <NodeName>PortNbr</NodeName>
                                <RTProperties>
                                    <Format>
                                        <int/>
                                    </Format>
                                    <Type><MIME>text/plain</MIME></Type>
                                </RTProperties>
                                <Value>Port Number</Value>
                            </Node>
                        </Node>
                    </Node>
                </Node>
            </Node>
        <Node>
            <NodeName>AauthPref</NodeName>
            <RTProperties>
                <Format>
                    <chr/>
                </Format>
                <Type><MIME>text/plain</MIME></Type>
            </RTProperties>
            <Value>Optional Preferred auth mechanism (see section 5.3.1.20 in [DMSTDOBJ]) </Value>
        </Node>
```

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
<Node>
        <NodeName>AppAuth</NodeName>
        <RTProperties>
            <Format>
                <node/>
            </Format>
        </RTProperties>
        <Node>
            <NodeName>Authentication Settings Name</NodeName>
            <RTProperties>
                <Format>
                    <node/>
                </Format>
            </RTProperties>
            <Node>
          <Node>
                <NodeName>AAuthLevel</NodeName>
                <RTProperties>
                    <Format>
                        <chr/>
                    </Format>
                    <Type><MIME>text/plain</MIME></Type>
                </RTProperties>
                <Value>Auth Level Value (section 5.3.1.19 in [DMSTDOBJ]) </Value>
          </Node>
          <Node>
                <NodeName>AAuthType</NodeName>
                <RTProperties>
                    <Format>
                        <chr/>
                    </Format>
                    <Type><MIME>text/plain</MIME></Type>
                </RTProperties>
                <Value>Auth Type Value (section 5.3.1.20 in [DMSTDOBJ]) </Value>
          </Node>
          <Node>
                <NodeName>AAuthName</NodeName>
                <RTProperties>
                    <Format>
                        <chr/>
                    </Format>
                    <Type><MIME>text/plain</MIME></Type>
                </RTProperties>
                <Value>Auth Name </Value>
          </Node>
          <Node>
                <NodeName>AAuthSecret</NodeName>
                <RTProperties>
                    <Format>
```

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
                                                    <chr/>
                                                </Format>
                                                <Type><MIME>text/plain</MIME></Type>
                                            </RTProperties>
                                            <Value>Auth Secret</Value>
                                        </Node>
                                        <Node>
                                            <NodeName>AAuthData</NodeName>
                                            <RTProperties>
                                                <Format>
                                                    <chr/>
                                                </Format>
                                                <Type><MIME>text/plain</MIME></Type>
                                            </RTProperties>
                                            <Value>Auth Nonce </Value>
                                        </Node>
                                    </Node>
                                </Node>
                            </Node>
</MgmtTree>
</SyncML>
                                            ]]>
                                        </Data>
                                    </Item>
                                </Add>
                            </SyncBody>
</SyncML>
```

B.2 CP_Prov_doc_1.xml

This reference document should be completed according to the specifics of the corresponding DM server being bootstrapped and test fest infrastructure by substituting the bold text with the appropriate parameters.

```
<?xml version="1.0"?>
<!DOCTYPE wap-provisioningdoc PUBLIC "-//WAPFORUM//DTD PROV 1.0//EN" "http://www.wapforum.org/DTD/prov.dtd">
<wap-provisioningdoc version="1.0">

<!-- Connectivities Definition -->

 <characteristic type="PXLOGICAL">
  <parm name="PROXY-ID" value="Logical Proxy ID"/>
  <parm name="NAME" value="Logical Proxy Name"/>
  <parm name="STARTPAGE" value="Logical Proxy Startpage"/>

  <characteristic type="PXPHYSICAL">
   <parm name="PHYSICAL-PROXY-ID" value="Physical Proxy ID "/>
   <parm name="PXADDR" value="Physical Proxy Address"/>
   <parm name="PXADDRTYPE" value="Physical Proxy Address Type"/>
```

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
    <parm name="TO-NAPID" value="Reference to Access Point"/>
    <characteristic type="PORT">
     <parm name="PORTNBR" value="Port Number"/>
    </characteristic>
   </characteristic>
  </characteristic>

  <characteristic type="NAPDEF">
   <parm name="NAPID" value="Access Point ID"/>
   <parm name="BEARER" value="Bearer type"/>
   <parm name="NAME" value="Access Point Name"/>
   <parm name="NAP-ADDRESS" value="Access Point Address"/>
   <parm name="NAP-ADDRTYPE" value="Access Point Address Type"/>
   <characteristic type="NAPAUTHINFO">
    <parm name="AUTHTYPE" value="Authentication Type"/>
    <parm name="AUTHNAME" value="Authentication Name"/>
    <parm name="AUTHSECRET" value="Authentication Secret"/>
   </characteristic>
  </characteristic>

  <!-- APPLICATION characteristic for DM -->
  <characteristic type="APPLICATION">
   <parm name="APPID" value="w7"/>
   <parm name="PROVIDER-ID" value="DM Server ID"/>
   <parm name="NAME" value="DM Server Name"/>
   <parm name="ADDR" value="DM Server Address"/>
   <parm name="TO-NAPID" value="Reference to Access Point"/>
   <characteristic type="APPAUTH">
    <parm name="AAUTHLEVEL" value="Authentication Level"/>
    <parm name="AAUTHTYPE" value="Authentication Type"/>
    <parm name="AAUTHNAME" value="Authentication Name"/>
    <parm name="AAUTHSECRET" value="Authentication Secret"/>
    <parm name="AAUTHDATA" value="Authentication Nonce"/>
   </characteristic>
  </characteristic>

</wap-provisioningdoc>
```

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix C. OMA DM Protocol packages

The contents of the messages described in the present Appendix is not complete - only the fields and headers required to be checked or generated by test tools are listed here. The messages sent by the UE may contain additional parameters, fields and headers which are not checked and must thus be ignored by test tools.

C.1 Package 0:Management Initiation Alert from server to client

As specified in [DMPRO] chapter 8.2 and in [DMNOTI] chapter 6, the DM 1.2 conformance test tool MAY send a Package 0 message in a WAP Push Server Notification with the following recommendation:

1. The package 0 message SHALL contain the following fields:
    a. 'version'. Binary value MUST be set to '0000001011' (Version = 1.21)
    b. 'session id'. Binary value MUST be set to '0001001000110100' (i.e. 0x1234)
    c. 'length-identifier'. Value depends upon the next field.
    d. 'server identifier'. Value MAY be set to ixit_ServerName
    e. future use' Binary value MUST be set to '00000000000000000000000000'
    f. 'initiator'. Binary value MUST be set to '1' (Server Initiated Management Session)
    g. 'user interaction mode'. Binary value SHOULD be set to '01' (i.e. background management). It MAY be '11' (i.e. user interaction) before a management action takes place.

C.2 Package 1: Initialization from client to server

As specified in [DMPRO] chapter 8.3, the detailed requirements for a valid package 1 are:

1. The package 1 message shall contain a 'SyncHdr' element
    a. The element 'SyncHdr' shall contain a 'VerDTD' element with value '1.2'.
    b. The element 'SyncHdr' shall contain a 'VerProto' element with value 'DM/1.2'.
    c. The element 'SyncHdr' shall contain a 'SessionID' element with non empty value.
    d. The element 'SyncHdr' shall contain a 'MsgID' element with non empty value.
    e. The element 'SyncHdr' shall contain a 'Target' element with non empty value.
    f. The element 'SyncHdr' shall contain a 'Source' element with non empty value.
2. The package 1 message shall contain a 'SyncBody' element
    a. The element 'SyncBody' shall contain a 'Alert' element.
    b. The element 'SyncBody' shall contain a 'Replace' element.

C.3 Package 2: Initialization from server to client

As specified in [DMPRO] chapter 8.4, the detailed requirements for a valid package 2 are:

1. The package 2 message shall contain a 'SyncHdr' element
    a. The element 'SyncHdr' shall contain a 'VerDTD' element with value '1.2'.
    b. The element 'SyncHdr' shall contain a 'VerProto' element with value 'DM/1.2'.
    c. The element 'SyncHdr' shall contain a 'SessionID' element with non empty value.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

d. The element ‘SyncHdr’ shall contain a ‘MsgID’ element with non empty value.
e. The element ‘SyncHdr’ shall contain a ‘Target’ element with non empty value.
f. The element ‘SyncHdr’ shall contain a ‘Source’ element with non empty value.

2. The package 1 message shall contain a ‘SyncBody’ element
   a. The element ‘SyncBody’ shall contain at least two ‘Status’ elements for ‘SyncHdr’ and ‘Alert’ commands.

C.4 Package 3: Client response sent to server

As specified in [DMPRO] chapter 8.5, the detailed minimal requirements for a valid package 3 are:

1. The package 3 message SHALL contain a ‘SyncHdr’ element
   a. The element ‘SyncHdr’ SHALL contain a ‘VerDTD’ element with value ‘1.2’.
   b. The element ‘SyncHdr’ SHALL contain a ‘VerProto’ element with value ‘DM/1.2’.
   c. The element ‘SyncHdr’ SHALL contain a ‘SessionID’ element with non empty value.
   d. The element ‘SyncHdr’ SHALL contain a ‘MsgID’ element with non empty value.
   e. The element ‘SyncHdr’ SHALL contain a ‘Target’ element with non empty value.
   f. The element ‘SyncHdr’ SHALL contain a ‘Source’ element with non empty value.
2. The package 3 message SHALL contain a ‘SyncBody’ element
   a. The element ‘SyncBody’ SHALL contain at least one ‘Status’ element for ‘SyncHdr’ command.

C.5 Package 4: Further server management operations

As specified in [DMPRO] chapter 8.6, the detailed minimal requirements for a valid package 4 are:

1. The package 4 message SHALL contain a ‘SyncHdr’ element
   a. The element ‘SyncHdr’ SHALL contain a ‘VerDTD’ element with value ‘1.2’.
   b. The element ‘SyncHdr’ SHALL contain a ‘VerProto’ element with value ‘DM/1.2’.
   c. The element ‘SyncHdr’ SHALL contain a ‘SessionID’ element with non empty value.
   d. The element ‘SyncHdr’ SHALL contain a ‘MsgID’ element with non empty value.
   e. The element ‘SyncHdr’ SHALL contain a ‘Target’ element with non empty value.
   f. The element ‘SyncHdr’ SHALL contain a ‘Source’ element with non empty value.
2. The package 4 message SHALL contain a ‘SyncBody’ element
   a. The element ‘SyncBody’ SHALL contain at least one ‘Status’ element for ‘SyncHdr’ command.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix D. Testcases applicability

D.1 Introduction

This section shall help implementers of the DM Enabler to select appropriate test cases that are applicable to the features implemented.

This appendix lists all the test cases testing only mandatory features and test cases that include optional. For the test cases implementing optional features, ICS (Implementation Conformance Specification) and IXIT (protocol implementation extra information) were identified based on the preconditions and SCR items – this results in a mapping from ICS/IXIT to applicable test cases as defined by Open Mobile Alliance.

D.2 Client Test cases testing only mandatory features

These test cases are independent from any precondition, are testing only mandatory SCRs and SHALL be run with every terminal.

Test Case

D.3 Client ICS

Preconditions		SCR Reference	Applicable
ICS	*Description*		*(yes/no)*
ics_md5_auth	Client supports MD5 authentication		
ics_command_exec	Client supports receiving command 'Exec'	DMREPPRO-PCE-C-007	
ics_exec_user_interaction	Client supports executing User Interaction Commands	DM-PRO-C-007	
ics_generic_alert	Clients supports sending generic alert	DM-PRO-C-011	
ics_large_object	Client supports large object handling	DM-PRO-C-004	
ics_md5_auth	Client supports MD5 authentication	DM-SEC-C-008	
ics_multi_message	Client supports multiple messages	DM-PRO-C-003	
ics_notification	Client supports Server Notification	DM-PRO-Session-C-001	
ics_prop_size	Client supports the Size property in leaf nodes	DMTND-Prop-C-004	
ics_secure	Client supports TLS1.0 or SSL3.0	DM-SEC-C-014 or DM-SEC-C-	

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

		015	
ics_sending_abort_alert	Client supports sending session abort alert	DM-PRO-Abort-C-001	
ics_test_auto	Client supports Test Automation Mode	N/A	
ics_transport_auth	Client supports transport layer authentication	DM-SEC-C-003 DM-SEC-C-010	
ics_http	Client supports HTTP transport	DM-SEC-C-004	
ics_secure	Client supports identification using TLS1.0 or SSL3.0 and supports at least one of the cipher suites : TLS_RSA_WITH_AES _128_CBC_SHA-1 (only with TLS1.0), SSL_RSA_WITH_3DES_EDE_CBC_SHA, SSL_ RSA_WITH_RC4_128_SHA	DM-SEC-C-013, DM-SEC-C-016, DM-SEC-C-017	
ics_support_confirm_reject_alert	Client supports Confirmation User Interaction Alert command	DM-PRO-UI-C-002	
ics_large_object	Client supports Large Object Handling. This is RECOMMENDED for clients.	DM-PRO-C-004	
ics_max_object_size	Client supports 'MaxObjSize' tag	DMREPPRO-MIE-C-005	
ics_command_atomic	Client supports receiving command 'Atomic'	DMREPPRO-PCE-C-004	

D.4 Client IXIT

Value column shall be filled with appropriate values that are supported by the device.

Preconditions		**Unit**	**Value**
IXIT	*Description*		
ixit_ServerName	Name of the DM 1.2 conformance test tool server	string	
ixit_ServerPass	Password of the DM 1.2 conformance test tool server	string	
ixit_UserName	DM User Name of the DM 1.2 client (used in Authentication procedures)	string	
ixit_UserPass	DM User Password of the DM 1.2 client (used in Authentication procedures)	string	
ixit_NextNonce	Initial Next Nonce used in authentication procedures	Octetstring	e.g. "6E6F6E6365"

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

D.5 Client ICS/IXIT to test case mapping

According to the ICS and IXIT marked in section D.3 and D.4 the applicable test cases can be derived from the following table.

Preconditions	Test Case
ics_md5_auth	DeviceManagement-v1.2-client-con-0201
ics_prop_size	DeviceManagement-v1.2-client-con-1306
ics_http and ics_notification	DeviceManagement-v1.2-client-con-2001
ics_http and ics_secure	DeviceManagement-v1.2-client-con-2601
ics_support_confirm_reject_alert	DeviceManagement-v1.2-client-con-0801 DeviceManagement-v1.2-client-con-0802
ics_large_object	DeviceManagement-v1.2-client-con-1701 DeviceManagement-v1.2-client-con-1702 DeviceManagement-v1.2-client-con-1703
ics_large_object and ica_max_object_size	DeviceManagement-v1.2-client-con-1704
ics_ transport_auth	DeviceManagement-v1.2-client-con-0401
ics_command_atomic	DeviceManagement-v1.2-client-con-1101

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix E. Optional Message handling macros

The macros described in this appendix are referenced in the test cases to avoid redundancy in the test cases itself.

E.1 DM Session Initialisation macro

Preconditions	The client is not involved in a session with the test tool

Macro steps: MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1a	←		Server Notification (Package #0 see C.1)	If ics_test_auto and ics_notification, the server sends a Server Notification with User Information Mode set to "Background Management Action"
1b				Else, test tool asks the user to initiate a DM Session
2	→		Setup-Request (Package #1 see C.2)	The client sends a Setup-Request message
3	←		Setup-Response (Package #2 see C.3)	The Test Tool sends OK response to the client with optional commands to continue the Session or without to close the Session

E.2 DM Authentication macro

As specified in [DMSEC] Chapter 5.3, the DM client or the DM server MAY require DM authentication. This authentication is a sequence of valid package #3 (see C.4) and package #4 (see C.5) messages containing authentication information as 'Cred', 'Chal' and 'NextNonce' elements.

E.3 DM Node Creation macro

Preconditions	The client is involved in a session with the test tool, the client expects a Setup-Response or a Server-Management Operations Message from the Test Tool

Macro steps: MESSAGE SEQUENCE

Step	Direction		Message	Comment
	UE	SS		
1	←		Setup-Response (Package #2 see C.3) or Server-Management Operations Message	The Test Tool sends Setup-Response or Server Management Operations Message (according to the precedent client message) with an 'Add' command on a interior or leaf node specified as macro input.

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

2	→	Setup-Request (Package #1 see C.2)	The client sends a Client-Response message with Status 200 for the 'Add' command.	

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix F. SCR mapping to Test Case (Informative)

F.1 SCR for DM Client

DM Protocol Requirements

Item	Function	Test Case
DM-PRO-C-001	Support of Session Setup Phase	Implicit
DM-PRO-C-002	Support of Session Abort	N/A
DM-PRO-C-003	Support of Multiple Messages	DeviceManagement-v1.2-client-con-1001
DM-PRO-C-004	Support of Large Object Handling. This is RECOMMENDED for clients.	DeviceManagement-v1.2-client-con-1701 DeviceManagement-v1.2-client-con-1702 DeviceManagement-v1.2-client-con-1703 DeviceManagement-v1.2-client-con-1704 DeviceManagement-v1.2-int-012
DM-PRO-C-005	Support of Management Phase	Implicit
DM-PRO-C-006	Support for executing Management Commands	Implicit
DM-PRO-C-007	Executing User Interaction Commands	DeviceManagement-v1.2-client-con-0801 DeviceManagement-v1.2-client-con-0802
DM-PRO-C-008	Support for sending Status and Result after receiving Management Operations	Implicit
DM-PRO-C-009	Support for standard SyncML command Format and Status and Result reporting	Implicit
DM-PRO-C-010	Support for sending asynchronous data via client initiated Alerts	DeviceManagement-v1.2-client-con-0102
DM-PRO-C-011	Sending Generic Alert	DeviceManagement-v1.2-int-037

DM Session Setup Phase

Item	Function	Test Case
DM-PRO-Session-C-001	Support Server Notification	DeviceManagement-v1.2-int-013
DM-PRO-Session-C-002	Sending Client Initiation and Device Info (Package #1) including Final element	DeviceManagement-v1.2-int-013 DeviceManagement-v1.2-client-con-0103
DM-PRO-Session-C-003	Sending Server-Initiated mgmt Alert	DeviceManagement-v1.2-int-013 DeviceManagement-v1.2-client-con-0103

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM-PRO-Session-C-004	Sending Client-Initiated mgmt Alert	DeviceManagement-v1.2-int-013 DeviceManagement-v1.2-client-con-0103
DM-PRO-Session-C-005	Sending Device Info in Replace Command in Package #1	DeviceManagement-v1.2-int-013 DeviceManagement-v1.2-client-con-0103

Session Abort

Item	Function	Test Case
DM-PRO-Abort-C-001	Sending Session Abort Alert	N/A
DM-PRO-Abort-C-002	Receiving Session Abort Alert	N/A
DM-PRO-Abort-C-003	Session Abort message includes Status and Results of executed commands	N/A
DM-PRO-Abort-C-004	Include Final in Message	N/A
DM-PRO-Abort-C-005	Sender of Abort discards the response if response is received	N/A

Multiple Messages

Item	Function	Test Case
DM-PRO-Mul-C-001	Last message within multiple messages must contain Final	DeviceManagement-v1.2-client-con-1001
DM-PRO-Mul-C-002	If message that is not the last one within Multiple Messages then the Next Message or Abort Alert must be sent	DeviceManagement-v1.2-client-con-1001

Large Object

Item	Function	Test Case
DM-PRO-LO-C-001	Response with Status 213 if data chunk that is not the last one is received	DeviceManagement-v1.2-client-con-1703
DM-PRO-LO-C-002	Management Commands inside Large Object is handled as Atomic	DeviceManagement-v1.2-client-con-1702
DM-PRO-LO-C-003	While sending data chunks all chunks except the last one must include “MoreData”	DeviceManagement-v1.2-client-con-1703
DM-PRO-LO-C-004	Indicate support for Large Object in DevDetail	DeviceManagement-v1.2-client-con-1701
DM-PRO-LO-C-005	Data chunks must be sent in continuous order without any new commands	DeviceManagement-v1.2-client-con-1703
DM-PRO-LO-C-006	Data that fits into a single message must be sent in a single message	DeviceManagement-v1.2-client-con-1701
DM-PRO-LO-C-007	Sending MaxObjSize to indicate size limitations for Package	DeviceManagement-v1.2-client-con-1701

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM-PRO-LO-C-008	Never encode packages bigger than the server indicated in MaxObjSize	DeviceManagement-v1.2-client-con-1704
DM-PRO-LO-C-009	Include Size in first data chunk	DeviceManagement-v1.2-client-con-1703
DM-PRO-LO-C-010	Validation of actual size and the Size value and report if not equal	DeviceManagement-v1.2-client-con-1702 (Implicit)

User Interaction Commands

Item	Function	Test Case
DM-PRO-UI-C-001	Executing Display Alert	DeviceManagement-v1.2-int-021
DM-PRO-UI-C-002	Executing Confirm or Reject Alert	DeviceManagement-v1.2-client-con-0801 DeviceManagement-v1.2-client-con-0802 DeviceManagement-v1.2-int-022
DM-PRO-UI-C-003	Executing Text Input Alert	DeviceManagement-v1.2-int-023
DM-PRO-UI-C-004	Executing Single Choice Alert	DeviceManagement-v1.2-int-024
DM-PRO-UI-C-005	Executing Multiple Choice Alert	DeviceManagement-v1.2-int-025
DM-PRO-UI-C-006	Order of the Items MUST be used in the same order as in the DM message	N/A

Generic Alert

Item	Function	Test Case
DM-PRO-GAlert-C-001	The Generic Alert have a relation to a Management Object	DeviceManagement-v1.2-int-037
DM-PRO-GAlert-C-002	The Generic Alert does not have a relation to a Management Object	N/A
DM-PRO-GAlert-C-003	LocURI must reference the address to the corresponding Management Object	DeviceManagement-v1.2-int-037
DM-PRO-GAlert-C-004	Support for Correlator	DeviceManagement-v1.2-client-con-2501 DeviceManagement-v1.2-int-037
DM-PRO-GAlert-C-005	Type must be included and it is RECOMMENDED to include URN or registered MIME-type as Type	N/A
DM-PRO-GAlert-C-006	Support for importance level, Mark	N/A

DM Representation Protocol

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Common use elements

Item	Function	Test Case
DMREPPRO-CUE-C-001	Support for 'Chal'	DeviceManagement-v1.2-client-con-0201
DMREPPRO-CUE-C-002	Support for 'Cmd'	Implicit
DMREPPRO-CUE-C-003	Support for 'CmdId'	Implicit
DMREPPRO-CUE-C-004	Support for 'CmdRef'	Implicit
DMREPPRO-CUE-C-005	Support for 'Cred'	Implicit
DMREPPRO-CUE-C-006	Support for 'Final'	DeviceManagement-v1.2-int-013 DeviceManagement-v1.2-client-con-0103
DMREPPRO-CUE-C-007	Support for 'LocName'	Implicit
DMREPPRO-CUE-C-008	Support for 'LocURI'	DeviceManagement-v1.2-client-con-0104
DMREPPRO-CUE-C-009	Support for 'MoreData'	DeviceManagement-v1.2-client-con-1703
DMREPPRO-CUE-C-010	Support for 'MsgID'	Implicit
DMREPPRO-CUE-C-011	Support for 'MsgRef'	Implicit
DMREPPRO-CUE-C-012	Support for sending 'RespURI'	N/A
DMREPPRO-CUE-C-013	Support for receiving 'RespURI'	Implicit
DMREPPRO-CUE-C-014	Support for 'SessionID'	Implicit
DMREPPRO-CUE-C-015	Support for 'Source'	Implicit
DMREPPRO-CUE-C-016	Support for 'SourceRef'	Implicit
DMREPPRO-CUE-C-017	Support for 'Target'	Implicit
DMREPPRO-CUE-C-018	Support for 'TargetRef'	Implicit

Meta Information elements

Item	Function	Test Case
DMREPPRO-MIE-C-001	Support for 'EMI'	N/A
DMREPPRO-MIE-C-002	Support for 'Format'	DeviceManagement-v1.2-int-003
DMREPPRO-MIE-C-003	Support for sending 'MaxMsgSize'	DeviceManagement-v1.2-client-con-1001
DMREPPRO-MIE-C-004	Support for receiving 'MaxMsgSize'	DeviceManagement-v1.2-client-con-1001
DMREPPRO-MIE-C-005	Support for 'MaxObjSize'	DeviceManagement-v1.2-client-con-1704 DeviceManagement-v1.2-int-012
DMREPPRO-MIE-C-006	Support for 'MetInf'	Implicit
DMREPPRO-MIE-C-007	Support for 'NextNonce'	N/A

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DMREPPRO-MIE-C-008	Support for 'Size'	DeviceManagement-v1.2-client-con-1304
DMREPPRO-MIE-C-009	Support for 'Type'	DeviceManagement-v1.2-client-con-1303

Data description elements

Item	Function	Test Case
DMREPPRO-DDE-C-001	Support for sending 'Correlator'	DeviceManagement-v1.2-int-037
DMREPPRO-DDE-C-002	Support for receiving 'Correlator'	DeviceManagement-v1.2-int-037

Protocol command elements

Item	Function	Test Case
DMREPPRO-PCE-C-001	Support for sending 'Alert'	DeviceManagement-v1.2-client-con-0102 DeviceManagement-v1.2-client-con-2001
DMREPPRO-PCE-C-002	Support for 'Replace'	DeviceManagement-v1.2-client-con-0103 DeviceManagement-v1.2-client-con-0601 DeviceManagement-v1.2-client-con-0602 DeviceManagement-v1.2-client-con-1307 DeviceManagement-v1.2-client-con-1308 DeviceManagement-v1.2-client-con-1501 DeviceManagement-v1.2-int-007 DeviceManagement-v1.2-int-008
DMREPPRO-PCE-C-003	Support for receiving 'Add'	DeviceManagement-v1.2-client-con-0501 DeviceManagement-v1.2-client-con-0502 DeviceManagement-v1.2-client-con-0503 DeviceManagement-v1.2-client-con-1702 DeviceManagement-v1.2-int-012
DMREPPRO-PCE-C-004	Support for receiving 'Atomic'	DeviceManagement-v1.2-client-con-1101
DMREPPRO-PCE-C-005	Support for receiving 'Copy'	DeviceManagement-v1.2-client-con-2401 DeviceManagement-v1.2-int-036
DMREPPRO-PCE-C-006	Support for receiving 'Delete'	DeviceManagement-v1.2-client-con-0901 DeviceManagement-v1.2-client-con-0902 DeviceManagement-v1.2-client-con-

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

		0903 DeviceManagement-v1.2-client-con-1401 DeviceManagement-v1.2-client-con-1601
DMREPPRO-PCE-C-007	Support for receiving 'Exec'	DeviceManagement-v1.2- client-con-2101 DeviceManagement-v1.2- client-con-2102 DeviceManagement-v1.2-client-con-2501 DeviceManagement-v1.2-int-037
DMREPPRO-PCE-C-008	Support for receiving 'Get'	DeviceManagement-v1.2-client-con-0302 DeviceManagement-v1.2-client-con-0303 DeviceManagement-v1.2-client-con-0304 DeviceManagement-v1.2-client-con-1302 DeviceManagement-v1.2-client-con-1303 DeviceManagement-v1.2-client-con-1304 DeviceManagement-v1.2-client-con-1305 DeviceManagement-v1.2-client-con-1306 DeviceManagement-v1.2-client-con-1401 DeviceManagement-v1.2-client-con-1701 DeviceManagement-v1.2-client-con-1703 DeviceManagement-v1.2-client-con-1704 DeviceManagement-v1.2-client-con-1801 DeviceManagement-v1.2-client-con-1901 DeviceManagement-v1.2-int-003 DeviceManagement-v1.2-int-004 DeviceManagement-v1.2-int-005 DeviceManagement-v1.2-int-006 DeviceManagement-v1.2-int-008 DeviceManagement-v1.2-int-026 DeviceManagement-v1.2-int-027
DMREPPRO-PCE-C-009	Support for receiving 'Sequence'	DeviceManagement-v1.2-client-con-0701
DMREPPRO-PCE-C-010	Support for sending 'Results'	DeviceManagement-v1.2-client-con-0302 DeviceManagement-v1.2-client-con-0303 DeviceManagement-v1.2-client-con-1703 DeviceManagement-v1.2-int-012

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Event Alert

Item	Function	Test Case
DMREPPRO-Alert-C-001	Sending Client Event Alert	N/A

DM Client Security requirements

The following specifies the ICS proforma tables for DM security for client devices that conform to [DMSEC].

Item	Function	Test Case
DM-SEC-C-001	Client must authenticate itself to a server	DeviceManagement-v1.2-client-con-0201 DeviceManagement-v1.2-client-con-0301 DeviceManagement-v1.2-int-001
DM-SEC-C-002	Client must authenticate a server	DeviceManagement-v1.2-int-002
DM-SEC-C-003	Support for transport layer authentication	DeviceManagement-v1.2-client-con-2601 DeviceManagement-v1.2-int-039 DeviceManagement-v1.2-int-040
DM-SEC-C-004	Support for HTTP transport	DeviceManagement-v1.2-client-con-2601
DM-SEC-C-005	Send credentials to server	DeviceManagement-v1.2-client-con-0201 DeviceManagement-v1.2-client-con-0301 DeviceManagement-v1.2-int-001
DM-SEC-C-006	Challenge Server	DeviceManagement-v1.2-int-002
DM-SEC-C-007	Support for application layer authentication	DeviceManagement-v1.2-client-con-0201
DM-SEC-C-008	Support for OMA DM syncml:auth-md5 type authentication	DeviceManagement-v1.2-client-con-0201 DeviceManagement-v1.2-client-con-0301 DeviceManagement-v1.2-int-001 DeviceManagement-v1.2-int-002
DM-SEC-C-009	Accept challenges from server that has not yet been successfully authenticated	N/A
DM-SEC-C-010	Integrity checking using HMAC-MD5	DeviceManagement-v1.2-client-con-0401 DeviceManagement-v1.2-int-010 DeviceManagement-v1.2-int-011
DM-SEC-C-011	Inserting HMAC in transport	DeviceManagement-v1.2-client-con-0401 DeviceManagement-v1.2-int-010 DeviceManagement-v1.2-int-011

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM-SEC-C-012	Using HMAC for all subsequent messages	DeviceManagement-v1.2-client-con-0401 DeviceManagement-v1.2-int-010 DeviceManagement-v1.2-int-011
DM-SEC-C-013	Identifying that the server is using TLS1.0 or SSL3.0	DeviceManagement-v1.2-client-con-2601 DeviceManagement-v1.2-int-039
DM-SEC-C-014	Support for TLS	DeviceManagement-v1.2-client-con-2601 DeviceManagement-v1.2-int-039 DeviceManagement-v1.2-int-040
DM-SEC-C-015	Support for SSL 3.0	DeviceManagement-v1.2-client-con-2601
DM-SEC-C-016	Supporting at least one of the cipher suites TLS_RSA_WITH_AES_128_CBC_SHA-1, TLS_RSA_WITH_3DES_EDE_CBC_SHA and TLS_RSA_WITH_RC4_128_SHA	DeviceManagement-v1.2-client-con-2601 DeviceManagement-v1.2-int-039
DM-SEC-C-017	Support for at least one of SSL_RSA_WITH_RC4_128_SHA and SSL_RSA_WITH_3DES_EDE_CBC_SHA	DeviceManagement-v1.2-int-040
DM-SEC-C-018	Bootstrap Security for Bootstrap via DM Profile	DM-1.2-int-018
DM-SEC-C-019	Transport neutral security for Bootstrap via DM Profile	DM-1.2-int-018
DM-SEC-C-020	Transport layer security for Bootstrap via DM Profile	DM-1.2-int-018
DM-SEC-C-021	Use of NETWORKID and USERPIN when Bootstrapping via DM Profile	DM-1.2-int-020
DM-SEC-C-022	Support of NETWORKID in Bootstrap via DM Profile	DM-1.2-int-018
DM-SEC-C-023	Support of USERPIN in Bootstrap via DM Profile	DM-1.2-int-019

DM Tree and Description

Item	Function	Test Case
DMTND-Prop-C-001	Support for the ACL property	DeviceManagement-v1.2-client-con-1301 DeviceManagement-v1.2-client-con-1401 DeviceManagement-v1.2-client-con-1501 DeviceManagement-v1.2-client-con-2102 DeviceManagement-v1.2-int-008
DMTND-Prop-C-002	Support for the Format property	DeviceManagement-v1.2-client-con-1302
DMTND-Prop-C-003	Support for the Name property	DeviceManagement-v1.2-client-con-1305 DeviceManagement-v1.2-client-con-1307

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DMTND-Prop-C-004	Support for the Size property in leaf nodes	DeviceManagement-v1.2-client-con-1306
DMTND-Prop-C-005	No support for the Size property in interior nodes	DeviceManagement-v1.2-client-con-1304
DMTND-Prop-C-006	Support for the Title property	N/A
DMTND-Prop-C-007	Support for the TStamp property	N/A
DMTND-Prop-C-008	Support for the Type property	DeviceManagement-v1.2-client-con-1303
DMTND-Prop-C-009	Support for the VerNo property	N/A
DMTND-Prop-C-010	Support Get?list=Struct	DeviceManagement-v1.2-int-026
DMTND-Prop-C-011	Support Get?list=StructData	DeviceManagement-v1.2-int-027
DMTND-Prop-C-012	Support Get?list=TNDS	DeviceManagement-v1.2-client-con-2301 DeviceManagement-v1.2-int-035

DM Standardized Objects

Item	Function	Test Case
SCR-DM-STDOBJ-C-001	Support of DevInfo object	DeviceManagement-v1.2-client-con-1201
SCR-DM-STDOBJ-C-002	Support of DevDetail Object	DeviceManagement-v1.2-client-con-1202 DeviceManagement-v1.2-client-con-1701
SCR-DM-STDOBJ-C-003	Support of DM Account Object	DeviceManagement-v1.2-client-con-1203
SCR-DM-STDOBJ-C-004	Support of Inbox Object	DeviceManagement-v 1.2-int-031

DM Bootstrap Client

Item	Function	Test Case
DM-BOOT-C-001	Support for OMA Client Provisioning Profile	DeviceManagement-v1.2-int-014 DeviceManagement-v1.2-int-015 DeviceManagement-v1.2-int-015a DeviceManagement-v1.2-int-015b DeviceManagement-v1.2-int-015c DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v1.2-int-017a
DM-BOOT-C-002	Support for OMA Device Management Profile	DeviceManagement-v1.2-int-016 DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v1.2-int-017

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

		DM-1.2-int-018 DM-1.2-int-019 DM-1.2-int-020
DM-BOOT-C-003	Provisioning Content granted Get, Replace and Delete ACL rights to ServerID in w7.	DeviceManagement-v1.2-int-015 DeviceManagement-v1.2-int-015a DeviceManagement-v1.2-int-015b DeviceManagement-v1.2-int-015c DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v1.2-int-017a
DM-BOOT-C-004	Support for OMA Client Provisioning Profile AND OMA Device Management	DeviceManagement-v1.2-int-015 DeviceManagement-v1.2-int-015a DeviceManagement-v1.2-int-015b DeviceManagement-v1.2-int-015c DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v1.2-int-017a
DM-BOOT-C-005	Map w7, NAPDEF (if supported) and PROXY (if supported) to management tree.	DeviceManagement-v1.2-int-015 DeviceManagement-v1.2-int-015a DeviceManagement-v1.2-int-015b DeviceManagement-v1.2-int-015c DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v1.2-int-017a
DM-BOOT-C-006	Device supports a Smartcard.	DeviceManagement-v1.2-int-015 DeviceManagement-v1.2-int-015a DeviceManagement-v1.2-int-015b DeviceManagement-v1.2-int-015c DeviceManagement-v1.2-int-016 DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v 1.2-int-017 DeviceManagement-v1.2-int-017a
DM-BOOT-C-007	DM Client is capable of detecting, retrieving, and processing DM Profile bootstrap data from the Smartcard.	DeviceManagement-v1.2-int-016 DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v 1.2-int-017
DM-BOOT-C-008	Smartcard bootstrap function is enabled by DM client and the smartcard has not been rejected by the device.	DeviceManagement-v1.2-int-015 DeviceManagement-v1.2-int-015a DeviceManagement-v1.2-int-015b DeviceManagement-v1.2-int-015c DeviceManagement-v1.2-int-016 DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v 1.2-int-017 DeviceManagement-v1.2-int-017a
DM-BOOT-C-009	Device retrieves bootstrap data from the Smartcard and applies it to the device configuration.	DeviceManagement-v1.2-int-015 DeviceManagement-v1.2-int-015a DeviceManagement-v1.2-int-015b DeviceManagement-v1.2-int-015c DeviceManagement-v1.2-int-016

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

		DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v 1.2-int-017 DeviceManagement-v1.2-int-017a
DM-BOOT-C-010	Support for embedded WBXML encoded TNDS objects.	DeviceManagement-v1.2-int-016 DeviceManagement-v1.2-int-016b DeviceManagement-v1.2-int-016c DeviceManagement-v 1.2-int-017 DM-1.2-int-018 DM-1.2-int-019 DM-1.2-int-020
DM-BOOT-C-011	Support for Inbox.	DM-1.2-int-018 DM-1.2-int-019 DM-1.2-int-020 DeviceManagement-v 1.2-int-032

DM Notification Initiated Session

Item	Function	Test Case
SCR-DM-NOTI-C-001	Support of Server-Alerted Management Session	DeviceManagement-v1.2-client-con-2001 DeviceManagement-v1.2-int-013
SCR-DM-NOTI-C-002	Receiving Notification message	DeviceManagement-v1.2-client-con-2001

DM Tree and Description Serialization

Item	Function	Test Case
DM-TNDS-C-001	Support of Encoding a TNDS object	DeviceManagement-v1.2-client-con-2301
DM-TNDS-C-002	Support of Decoding a TNDS object	DeviceManagement-v1.2-int-038

F.2 SCR for DM Server

DM Protocol Requirements

Item	Function	Test Case
DM-PRO-S-001	Support of Session Setup Phase	ALL (Implicit)
DM-PRO-S-002	Support of Session Abort	DeviceManagement-v1.2-server-con-1201
DM-PRO-S-003	Support of Multiple Messages	DeviceManagement-v1.2-server-con-0701
DM-PRO-S-004	Support of Large Object Handling	DeviceManagement-v1.2-int-012
DM-PRO-S-005	Support of Management Phase	ALL (Implicit)
DM-PRO-S-006	Support for sending Management Commands	ALL (Implicit)

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM-PRO-S-007	Sending User Interaction Commands	DeviceManagement-v1.2-server-con-1501 DeviceManagement-v1.2-server-con-1601
DM-PRO-S-008	Support for sending Status and Results on Client Commands and Alerts	ALL (Implicit)
DM-PRO-S-009	Support of Generic Alert	DeviceManagement-v1.2-server-con-2001 DeviceManagement-v1.2-int-037
DM-PRO-S-010	Support application layer authentication	DeviceManagement-v1.2-int-001

DM Session Setup Phase

Item	Function	Test Case
DM-PRO-Session-S-001	Support Server Notification	DeviceManagement-v1.2-server-con-1801 DeviceManagement-v1.2-int-013
DM-PRO-Session-S-002	Support of receiving initiation message from client (Package #1), perform authentication and send initiation (Package #2)	Implicit

Session Abort

Item	Function	Test Case
DM-PRO-Abort-S-001	Sending Session Abort Alert	N/A
DM-PRO-Abort-S-002	Receiving Session Abort Alert	DeviceManagement-v1.2-server-con-1201
DM-PRO-Abort-S-003	Session Abort message includes Status and Results of executed commands	N/A
DM-PRO-Abort-S-004	Include Final in Message	N/A
DM-PRO-Abort-S-005	Sender of Abort must discard the response if response is received	N/A

Multiple Messages

Item	Function	Test Case
DM-PRO-Mul-S-001	Last message within multiple messages must contain Final	DeviceManagement-v1.2-server-con-0701
DM-PRO-Mul-S-002	If message that is not the last one within Multiple Messages then the Next Message or Abort Alert must be sent	DeviceManagement-v1.2-server-con-0701

Large Object

Item	Function	Test Case
DM-PRO-LO-S-001	Response with Status 213 if data chunk that is not the last one is received	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM-PRO-LO-S-002	Management Commands inside Large Object is handled as Atomic	DeviceManagement-v1.2-server-con-1002 DeviceManagement-v1.2-server-con-1102
DM-PRO-LO-S-003	While sending data chunks all chunks except the last one must include “MoreData”	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101
DM-PRO-LO-S-004	Data chunks must be sent in continues order without any new ccommands	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101
DM-PRO-LO-S-005	Data that fits into a single message must be sent in a single message	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101
DM-PRO-LO-S-006	Sending MaxObjSize to indicate size limitations for Package	DeviceManagement-v1.2-server-con-1002 DeviceManagement-v1.2-server-con-1102
DM-PRO-LO-S-007	Never encode packages bigger than the server indicated in MaxObjSize	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101
DM-PRO-LO-S-008	Include Size in first data chunk	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101
DM-PRO-LO-S-009	Validation of actual size and the Size value and report if not equal	DeviceManagement-v1.2-server-con-1002 DeviceManagement-v1.2-server-con-1102 (Implicit)

User Interaction Commands

Item	Function	Test Case
DM-PRO-UI-S-001	Sending Display Alert	DeviceManagement-v1.2-server-con-1501 DeviceManagement-v1.2-int-021
DM-PRO-UI-S-002	Sending Confirm or Reject Alert	DeviceManagement-v1.2-server-con-1601 DeviceManagement-v1.2-int-022
DM-PRO-UI-S-003	Sending Text Input Alert	DeviceManagement-v1.2-server-con-1701 DeviceManagement-v1.2-int-023
DM-PRO-UI-S-004	Sending Single Choice Alert	DeviceManagement-v1.2-int-024
DM-PRO-UI-S-005	Sending Multiple Choice Alert	DeviceManagement-v1.2-int-025
DM-PRO-UI-S-006	Order of the Items MUST be followed in the DM message	DeviceManagement-v1.2-int-025:

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Generic Alert

Item	Function	Test Case
DM-PRO-GAlert-S-001	Support for receiving, parsing and send Status back to client	DM-1.2-server-con-1901 DeviceManagement-v1.2-server-con-2001 DeviceManagement-v1.2-int-037
DM-PRO-GAlert-S-002	Perform action from the data content in the Generic Alert	DM-1.2-server-con-1901

DM Representation Protocol

Common use elements

Item	Function	Test Case
DMREPPRO-CUE-S-001	Support for 'Chal'	Implicit
DMREPPRO-CUE-S-002	Support for 'Cmd'	Implicit
DMREPPRO-CUE-S-003	Support for 'CmdId'	Implicit
DMREPPRO-CUE-S-004	Support for 'CmdRef'	Implicit
DMREPPRO-CUE-S-005	Support for 'Cred'	Implicit
DMREPPRO-CUE-S-006	Support for 'Final'	DeviceManagement-v1.2-server-con-0701
DMREPPRO-CUE-S-007	Support for 'LocName'	Implicit
DMREPPRO-CUE-S-008	Support for 'LocURI'	Implicit
DMREPPRO-CUE-S-009	Support for 'MoreData'	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101
DMREPPRO-CUE-S-010	Support for 'MsgID'	Implicit
DMREPPRO-CUE-S-011	Support for 'MsgRef'	Implicit
DMREPPRO-CUE-S-012	Support for sending 'RespURI'	Implicit
DMREPPRO-CUE-S-013	Support for receiving 'RespURI'	Implicit
DMREPPRO-CUE-S-014	Support for 'SessionID'	Implicit
DMREPPRO-CUE-S-015	Support for 'Source'	Implicit
DMREPPRO-CUE-S-016	Support for 'SourceRef'	Implicit
DMREPPRO-CUE-S-017	Support for 'Target'	Implicit
DMREPPRO-CUE-S-018	Support for 'TargetRef'	Implicit

Data description elements

Item	Function	Test Case
DMREPPRO-DDE-S-001	Support for sending 'Correlator'	DeviceManagement-v1.2-server-con-2001 DeviceManagement-v1.2-int-037
DMREPPRO-DDE-S-002	Support for receiving 'Correlator'	DeviceManagement-v1.2-server-con-2001 DeviceManagement-v1.2-int-037

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Meta Information elements

Item	Function	Test Case
DMREPPRO-MIE-S-001	Support for 'EMI'	N/A
DMREPPRO-MIE-S-002	Support for 'Format'	Implicit
DMREPPRO-MIE-S-003	Support for sending 'MaxMsgSize'	DeviceManagement-v1.2-server-con-0701 DeviceManagement-v1.2-server-con-1001 DeviceManagement-v1.2-server-con-1002 DeviceManagement-v1.2-server-con-1101 DeviceManagement-v1.2-server-con-1102
DMREPPRO-MIE-S-004	Support for receiving 'MaxMsgSize'	DeviceManagement-v1.2-server-con-0701 DeviceManagement-v1.2-server-con-1001 DeviceManagement-v1.2-server-con-1002 DeviceManagement-v1.2-server-con-1101 DeviceManagement-v1.2-server-con-1102
DMREPPRO-MIE-S-005	Support for 'MaxObjSize'	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101 DeviceManagement-v1.2-server-con-1002. DeviceManagement-v1.2-server-con-1102
DMREPPRO-MIE-S-006	Support for 'MetInf'	DeviceManagement-v1.2-server-con-1001. DeviceManagement-v1.2-server-con-1101 DeviceManagement-v1.2-server-con-1002. DeviceManagement-v1.2-server-con-1102
DMREPPRO-MIE-S-007	Support for 'NextNonce'	Implicit
DMREPPRO-MIE-S-008	Support for 'Size'	DeviceManagement-v1.2-server-con-1001 DeviceManagement-v1.2-server-con-1101 DeviceManagement-v1.2-server-con-1102
DMREPPRO-MIE-S-009	Support for 'Type'	Implicit

Protocol command elements

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Test Case
DMREPPRO-PCE-S-001	Support for 'Alert'	DeviceManagement-v1.2-server-con-0204
DMREPPRO-PCE-S-002	Support for 'Replace'	DeviceManagement-v1.2-server-con-0203 DeviceManagement-v1.2-server-con-0501 DeviceManagement-v1.2-server-con-0801 DeviceManagement-v1.2-server-con-0901 DeviceManagement-v1.2-int-007 DeviceManagement-v1.2-int-008
DMREPPRO-PCE-S-003	Support for sending 'Add'	DeviceManagement-v1.2-server-con-0401 DeviceManagement-v1.2-server-con-1001 DeviceManagement-v1.2-server-con-1101 DeviceManagement-v1.2-int-012
DMREPPRO-PCE-S-004	Support for sending 'Atomic'	DeviceManagement-v1.2-server-con-0901
DMREPPRO-PCE-S-005	Support for sending 'Copy'	DeviceManagement-v1.2-int-036
DMREPPRO-PCE-S-006	Support for sending 'Delete'	DeviceManagement-v1.2-server-con-0601
DMREPPRO-PCE-S-007	Support for sending 'Exec'	DeviceManagement-v1.2-server-con-2001 DeviceManagement-v1.2-int-037
DMREPPRO-PCE-S-008	Support for sending 'Get'	DeviceManagement-v1.2-server-con-0301 DeviceManagement-v1.2-server-con-1002 DeviceManagement-v1.2-server-con-1102 DeviceManagement-v1.2-server-con-1301 DeviceManagement-v1.2-server-con-1401 DeviceManagement-v1.2-int-003 DeviceManagement-v1.2-int-004 DeviceManagement-v1.2-int-005 DeviceManagement-v1.2-int-006 DeviceManagement-v1.2-int-008 DeviceManagement-v1.2-int-026 DeviceManagement-v1.2-int-027
DMREPPRO-PCE-S-009	Support for sending 'Sequence'	DeviceManagement-v1.2-server-con-0801
DMREPPRO-PCE-S-010	Support for receiving 'Results'	DeviceManagement-v1.2-int-012

Event Alert

Item	Function	Test Case
DMREPPRO-Alert-S-001	Receiving Client Event Alert	DeviceManagement-v1.2-server-con-0204

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM Server Security requirements

Item	Function	Test Case
DM-SEC-S-001	Different password for each client	N/A
DM-SEC-S-002	Support for client authentication at the transport layer	DeviceManagement-v1.2-server-con-2101 DeviceManagement-v1.2-server-con-2102 DeviceManagement-v1.2-int-039 DeviceManagement-v1.2-int-040
DM-SEC-S-003	Send credentials to client	DeviceManagement-v1.2-int-002 DeviceManagement-v1.2-int-011
DM-SEC-S-004	Challenge Client	DeviceManagement-v1.2-int-001 DeviceManagement-v1.2-int-010
DM-SEC-S-005	Support for clients authentication at the application layer	DeviceManagement-v1.2-int-001 DeviceManagement-v1.2-int-010
DM-SEC-S-006	MD5 challenge to client	DeviceManagement-v1.2-int-002
DM-SEC-S-007	MD5 challenge to client in conjunction with transport layer security	N/A
DM-SEC-S-008	Supply of a new nonce with one more challenge if authentication fails	DeviceManagement-v1.2-int-001
DM-SEC-S-009	Using new nonce for each new session	N/A
DM-SEC-S-010	Accept challenges from clients that have not yet been successfully authenticated	Implicit
DM-SEC-S-011	Integrity checking using HMAC-MD5	DeviceManagement-v1.2-int-010 DeviceManagement-v1.2-int-011
DM-SEC-S-012	Inserting HMAC in transport	DeviceManagement-v1.2-int-010 DeviceManagement-v1.2-int-011
DM-SEC-S-013	Using HMAC for all subsequent messages	DeviceManagement-v1.2-int-010 DeviceManagement-v1.2-int-011
DM-SEC-S-014	Support for HTTP transport	DeviceManagement-v1.2-server-con-2101 DeviceManagement-v1.2-server-con-2102
DM-SEC-S-015	Support for TLS 1.0 [TLS]	DeviceManagement-v1.2-server-con-2101 DeviceManagement-v1.2-int-039
DM-SEC-S-016	Support for SSL3.0 [SSL3.0]	DeviceManagement-v1.2-server-con-2102 DeviceManagement-v1.2-int-040
DM-SEC-S-017	Using OMA DM over HTTP	DeviceManagement-v1.2-server-con-2101
DM-SEC-S-018	Using TLS	DeviceManagement-v1.2-server-con-2101 DeviceManagement-v1.2-int-039

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM-SEC-S-019	Using SSL3.0	DeviceManagement-v1.2-server-con-2102 DeviceManagement-v1.2-int-040
DM-SEC-S-020	Supporting all three cipher suites TLS_RSA_WITH_AES_128_CBC_SHA-1, TLS_RSA_WITH_3DES_EDE_CBC_SHA and TLS_RSA_WITH_RC4_128_SHA	DeviceManagement-v1.2-server-con-2101 DeviceManagement-v1.2-int-039
DM-SEC-S-021	Support for both of SSL_RSA_WITH_RC4_128_SHA and SSL_RSA_WITH_3DES_EDE_CBC_SHA	DeviceManagement-v1.2-server-con-2102 DeviceManagement-v1.2-int-040
DM-SEC-S-022	Bootstrap Security for Bootstrap via DM Profile	DeviceManagement-v 1.2-int-018 DeviceManagement-v 1.2-int-019 DeviceManagement-v 1.2-int-020
DM-SEC-S-023	Transport neutral security for Bootstrap via DM Profile	DeviceManagement-v 1.2-int-018 DeviceManagement-v 1.2-int-019 DeviceManagement-v 1.2-int-020
DM-SEC-S-024	Transport layer security for Bootstrap via DM Profile	N/A
DM-SEC-S-025	Use of NETWORKID and USERPIN when Bootstrapping via DM Profile	DM-1.2-int-020
DM-SEC-S-026	Support of NETWORKID in Bootstrap via DM Profile	DM-1.2-int-018
DM-SEC-S-027	Support of USERPIN in Bootstrap via DM Profile	DM-1.2-int-019

DM Tree and Description

Item	Function	Test Case
DMTND-Prop-S-001	Support for the ACL property	DeviceManagement-v1.2-int-008
DMTND-Prop-S-002	Support for the Format property	Implicit
DMTND-Prop-S-003	Support for the Name property	Implicit
DMTND-Prop-S-004	Support for the Size property in leaf nodes	DeviceManagement-v1.2-server-con-1001 DeviceManagement-v1.2-server-con-1101 DeviceManagement-v1.2-server-con-1102
DMTND-Prop-S-005	No support for the Size property in interior nodes	N/A
DMTND-Prop-S-006	Support for the Title property	N/A
DMTND-Prop-S-007	Support for the TStamp property	N/A
DMTND-Prop-S-008	Support for the Type property	Implicit
DMTND-Prop-S-009	Support for the VerNo property	N/A
DMTND-Prop-S-010	Support Get?list=Struct	DeviceManagement-v1.2-server-con-1301 DeviceManagement-v1.2-int-026
DMTND-Prop-S-011	Support Get?list=StructData	DeviceManagement-v1.2-server-con-1401 DeviceManagement-v1.2-int-027
DMTND-Prop-S-012	Support Get?list=TNDS	DeviceManagement-v1.2-int-035

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DM Standardized Objects

Item	Function	Test Case
SCR-DM-STDOBJ-S-001	Support of DevInfo object	DeviceManagement-v1.2-server-con-0203
SCR-DM-STDOBJ-S-002	Support of DevDetail Object	Implicit
SCR-DM-STDOBJ-S-003	Support of DM Account Object	Implicit
SCR-DM-STDOBJ-S-004	Support of Inbox Object	DeviceManagement-v 1.2-int-031 DeviceManagement-v 1.2-int-032

DM Bootstrap Server

Item	Function	Test Case
DM-BOOT-S-001	Support for OMA Client Provisioning Profile	DeviceManagement-v1.2-int-014
DM-BOOT-S-002	Support for OMA Device Management Profile	DM-1.2-int-018 DM-1.2-int-019 DM-1.2-int-020
DM-BOOT-S-003	Encode DM message into WBXML.	DM-1.2-int-018 DM-1.2-int-019 DM-1.2-int-020

DM Notification Initiated Session

Item	Function	Test Case
SCR-DM-NOTI-S-001	Support of Server-Alerted Management Session	DeviceManagement-v1.2-server-con-1801 DeviceManagement-v1.2-int-013
SCR-DM-NOTI-S-002	Sending of Notification message	DeviceManagement-v1.2-server-con-1801
SCR-DM-NOTI-S-003	Notification message <version> field value is the binary value ‘0000001011’	DeviceManagement-v1.2-server-con-1801 DeviceManagement-v1.2-int-013

DM Tree and Description Serialization

Item	Function	Test Case
DM-TNDS-S-001	Support of Encoding a TNDS object	DeviceManagement-v1.2-int-038
DM-TNDS-S-002	Support of Decoding a TNDS object	DeviceManagement-v1.2-int-035

© 2010 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. OMA-Template-EnablerTestSpec-20070101-]