Enabler Test Specification for Device Management Interoperability

Candidate Version 1.2 – 24 May 2006

Open Mobile Alliance

OMA-ETS-DM_INT-V1_2-20060524-C

Contents

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.

2. References

2.1 Normative References

[ERELD] ”Enabler Release Definition for Device Management”, 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_0. URL:http://www.openmobilealliance.org

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

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

[SYNCHTTP] “SyncML HTTP Binding Specification”, Open Mobile Alliance™, OMA-SyncML-HTTPBinding- V1_2_0, URL:http://www.openmobilealliance.org

[SYNCMETA] “SyncML Meta Information, version 1.2”. Open Mobile Alliance™. OMA-SyncML-MetaInfo- V1_2_0 URL:http://www.openmobilealliance.org

[SYNCOBEX] “SyncML OBEX Binding Specification”, Open Mobile Alliance™, OMA-SyncML-OBEXBinding- V1_2_0, URL:http://www.openmobilealliance.org

[SYNCWSP] “SyncML WSP Binding Specification”, Open Mobile Alliance™, OMA-SyncML-WSPBinding- V1_2_0, URL:http://www.openmobilealliance.org

[IOPPROC] “OMA Interoperability Policy and Process”, Open Mobile Alliance™, OMA-IOP-Process-V1_4, 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

2.2 Informative References

[OMADICT] “Dictionary for OMA specifications”. Open Mobile Alliance™. OMA-Dictionary-v1_0. http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope”, 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

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.

5. Device Management Client Conformance Test Cases

5.1 Device Management Client Conformance Test Group #1

5.1.1 Sending of valid Alert command

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
SCR Reference	DMREPPRO-PCE-C-001 Support for sending 'Alert'
Test Tool	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	The Test Object MUST send valid Client Initiated Alert.

5.1.2 Sending of Device Information

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
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace'
Test Tool	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	The Test Object MUST send its Device Information in a Replace command

5.1.3 Checking of Source LocURI

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
Specification Reference	[DMREPU] Chapter 6.1.10

SCR Reference	DMREPPRO-CUE-C-008 Support for 'LocURI'
Test Tool	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	The value of Source LocURI in the SyncHdr sent by the client MUST be equal to the value sent in ./DevInfo/DevId

5.2 Device Management Client Conformance Test Group #2

5.2.1 Switching of the authentication scheme based on the challenge (MD5)

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).
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
Test Tool	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	The Test Object MUST update its authentication scheme and send credentials using MD5 in the next session.

5.3 Device Management Client Conformance Test Group #3

5.3.1 Support of the MD5 Digest authentication scheme

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
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	SCTS DM 1.2 as a server
Preconditions	Incoming userid & password configured on SCTS should match those of the clients outgoing userid & password.
Pass-Criteria	The Test Object MUST send valid credentials encoded using the MD5 Digest authentication scheme.

5.3.2 Respond with a Results for a Get on the Root node

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	SCTS DM 1.2 as a server
Preconditions	SCTS should have ACL access rights for Get on the Root node.
Pass-Criteria	The Test Object MUST respond with a Results containing at least the following element: DevInfo, DevDetail, SyncML.

5.3.3 Respond with a Results for a Get on a leaf node

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	SCTS DM 1.2 as a server
Preconditions	SCTS should have ACL access rights for Get on the leaf node.
Pass-Criteria	The Test Object MUST respond with a Results.

5.3.4 Respond for a Get on a non-existant node

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 404 status code on the Get.

5.4 Device Management Client Conformance Test Group #4

5.4.1 Support of HMAC and use of insecure transport

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 DM-SEC-C-012 Using HMAC for all subsequent messages
Test Tool	SCTS DM 1.2 as a server
Preconditions	The client should support HMAC and use a insecure transport.
Pass-Criteria	The Test Object MUST send valid HMAC.

5.5 Device Management Client Conformance Test Group #5

5.5.1 Adding of interior node to a client

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’
Test Tool	SCTS DM 1.2 as a server
Preconditions	The node MUST not exist on the device
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.

5.5.2 Adding of leaf node to a client

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 200 status code and the new leaf node MUST exist.

5.5.3 Add on an existing leaf node

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	SCTS DM 1.2 as a server
Preconditions	Test case 501 should have passed with a 200 status code.
Pass-Criteria	The Test Object MUST return a 418 status code.

5.6 Device Management Client Conformance Test Group #6

5.6.1 Handling of Replace

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.
Specification Reference	[DMREPU] Chapter 6.6.11
SCR Reference	DMREPPRO-PCE-C-002 Support for 'Replace'
Test Tool	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 200 status code.

5.6.2 Rejecting a Replace on a non-existant node

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'
Test Tool	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 404 status code.

5.7 Device Management Client Conformance Test Group #7

5.7.1 Handling of Sequence command

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST behave according to the rules specified for Sequence.

5.8 Device Management Client Conformance Test Group #8

5.8.1 Accept on Confirmation User Interaction Alert command

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	[DMPRO] Chapter 10.2
SCR Reference	DM-PRO-UI-C-002 Executing Confirm or Reject Alert
Test Tool	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	The test object must accept the change and send a status of 200 status code on the Alert.

5.8.2 Reject on Confirmation User Interaction Alert command

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
Test Tool	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	The test object must reject the change and send a status of 304 status code on the Alert.

5.9 Device Management Client Conformance Test Group #9

5.9.1 Deletion of interior node

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	SCTS DM 1.2 as a server
Preconditions	There should be interior node configured under test node.
Pass-Criteria	The Test Object MUST return either a 200/405 status code.

5.9.2 Delete on an non-existant node

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 404 status code.

5.9.3 Delete on a Permanent node

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 a Permanent node
Specification Reference	[DMREPU] Chapter 6.6.5
SCR Reference	DMREPPRO-PCE-C-006 Support for receiving 'Delete'
Test Tool	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 405 status code.

5.10 Device Management Client Conformance Test Group #10

5.10.1 Handling of multiple messages

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The session MUST complete successfully.

5.11 Device Management Client Conformance Test Group #11

5.11.1 Handling of Atomic command

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	SCTS DM 1.2 as a server

Preconditions	None.
Pass-Criteria	The Test Object MUST return a 200 status code on the Atomic.

5.12 Device Management Client Conformance Test Group #12

5.12.1 Structure of the ./DevInfo standard object

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 [DMREPU] Chapter 6.6.12
SCR Reference	SCR-DM-STDOBJ-C-001 Support of DevInfo object
Test Tool	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Results returned by the Test Object MUST contain the names of all the mandatory nodes under /DevInfo seperated by /

5.12.2 Structure of the./DevDetail standard object

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Results returned by the Test Object MUST contain the names of all the mandatory nodes under ./DevDetail seperated by /

5.12.3 Structure of the DMAcc MO

Test Case Id	DeviceManagement-v1.2-client-con-1203

Test Object	Client device
Test Case Description	To check if the structure of the DMAcc MO is correct.
Specification Reference	[DMSTDOBJ] Chapter 5.3.1
SCR Reference	[DMSTDOBJ] SCR-DM-STDOBJ-C-003, SCR-DM-STDOBJ-S-003
Test Tool	SCTS DM 1.2 as a server
Preconditions	1. Client must submit to server a DDF or XML schema description of the expected node structure 2. Client must enter <Interior Node> location into the test tool 3. Server address, port number, authentication settings, and connectivity definitions provisioned into the client as applicable.
Procedure	1. Server will perform a GET on <Interior Node> as defined by submitted DDF or XML Schema 2. Server will perform a GET on <Interior Node> / AppAddr/<x> 3. If supported Server will perform a GET on <Interior Node>/AppAddr/<x>/Port/<x> 4. Server will perform a GET on <Interior Node>/ AppAuth /<x>/ 5. If supported Server will perform a GET on <Interior Node>/ ToConRef /<x>
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: 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

5.13 Device Management Client Conformance Test Group #13

5.13.1 Checking that Root node has ACL

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	DMTND-Prop-C-001 Support for the ACL property
Test Tool	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	Response to Get on '.?prop=ACL' MUST be 200 and the Results must contain valid ACL .

5.13.2 Support of Get on Format property on an Interior node ('.')

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 a Interior 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	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	Status to Get on '.?prop=Format' MUST be 200 and the Results must contain the data 'node'.

5.13.3 Support of Get on Type property on an Interior node ('.')

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 a Interior 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	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	Response to Get on '.?prop=Type' MUST be 200 and the Results must be null or point to DDF document.

5.13.4 Support of Get on the Size property on an Interior node ('.')

Test Case Id	DeviceManagement-v1.2-client-con-1304
Test Object	Client device
Test Case Description	To check if the Test Object supports Get on the Size property on a Interior 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-005 No support for the Size property in interior nodes
Test Tool	SCTS DM 1.2 as a server
Preconditions	None

Pass-Criteria	Status to Get on '.?prop=Size' MUST be a 406.

5.13.5 Support of Get on the Name property on an Interior node ('./DevDetail/URI')

Test Case Id	DeviceManagement-v1.2-client-con-1305	
Test Object	Client device	
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	SCTS DM 1.2 as a server	
Preconditions	None	
Pass-Criteria	Status to Get on '.DevDetail/URI?prop=Name' MUST be 200 and the Results must have the data as 'URI'.	

5.13.6 Support of Get on the Size property on a leaf node('./DevDetail/URI/MaxTotLen')

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	SCTS DM 1.2 as a server	
Preconditions	None	
Pass-Criteria	Status to Get on '.DevDetail/URI?prop=Size' MUST be 200.	

5.13.7 Replace on the Name property on a permanent node ('./DevDetail')

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	SCTS DM 1.2 as a server
Preconditions	None
Pass-Criteria	Status to Replace on '.DevDetail?prop=Name' MUST be 405.

5.13.8 Support of Replace on the ACL property on the interior test node

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	SCTS DM 1.2 as a server
Preconditions	An interior test node must be set prior to the execution of this test and there should be replace access rights for SCTS.
Pass-Criteria	Status to Replace on the interior test node MUST be 200.

5.14 Device Management Client Conformance Test Group #14

5.14.1 Checking of ACL enforcement

Test Case Id	DeviceManagement-v1.2-client-con-1401
Test Object	Client device
Test Case Description	To check if the Test Object enforces ACL. SCTS replaces 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	SCTS DM 1.2 as a server
Preconditions	SCTS should have replace access rights on the test interior node.
Pass-Criteria	SCTS should get a 200 status code for Get and 425 status code for Delete.

5.15 Device Management Client Conformance Test Group #15

5.15.1 ACL setting

Test Case Id	DeviceManagement-v1.2-client-con-1501
Test Object	Client device
Test Case Description	SCTS removes the Replace access right for the test interior node and tries to replace the ACL of a 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	SCTS DM 1.2 as a server
Preconditions	SCTS should have replace access right on the test interior node and this node should have atleast one leaf node as its child.
Pass-Criteria	SCTS should receive a 425 status code for the Replace.

5.16 Device Management Client Conformance Test Group #16

5.16.1 Deletion of a leaf node

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
SCR Reference	DMREPPRO-PCE-C-006 Support for receiving 'Delete'
Test Tool	SCTS DM 1.2 as a server
Preconditions	The device should allow Adding and Deleting of nodes. There should be a leaf node under the test node.

Pass-Criteria	The Test Object MUST return a 200/405 status code.

5.17 Device Management Client Conformance Test Group #17

5.17.1 Support of Large Object Delivery Mechanism

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.SCTS 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.
Test Tool	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	SCTS should receive a status of 200 on Get and the result should contain a value of either 'true' or 'false'.

5.17.2 Large Object Delivery rules

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.SCTS Adds 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.
Test Tool	SCTS DM 1.2 as a server
Preconditions	This test is executed only if the Test Object indicates support for Large Object.
Pass-Criteria	SCTS should receive a status code of 200.

5.17.3 Sending of Results with a Large Object

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.SCTS 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.
Test Tool	SCTS DM 1.2 as a server
Preconditions	This test is executed only if Test Case 1702 successfully added a leaf node with Large Object.
Pass-Criteria	SCTS should receive a status code of 200 on the Get and valid results.

5.17.4 MaxObjSize check

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.SCTS 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. DMREPPRO-MIE-C-005 Support for 'MaxObjSize'
Test Tool	SCTS DM 1.2 as a server
Preconditions	This test is executed only if Test Case 1702 successfully added a leaf node with Large Object.
Pass-Criteria	SCTS should receive a status code of 413.

5.18 Device Management Client Conformance Test Group #18

5.18.1 Handling of Get with 'list=Struct”

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'. SCTS 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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	SCTS should receive a status code of either 200/406. If status is 200, SCTS should receive valid results.

5.19 Device Management Client Conformance Test Group #19

5.19.1 Handling of Get with 'list=StructData”

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=Struct'. SCTS 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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	SCTS should receive a status code of either 200/406. If status is 200, SCTS should receive valid results.

5.20 Device Management Client Conformance Test Group #20

5.20.1 Support of Notification Initiated Session

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 DMREPPRO-PCE-C-001 Support for sending 'Alert'
Test Tool	SCTS DM 1.2 as a server
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.

5.21 Device Management Client Conformance Test Group #21

5.21.1 Exec on a node with AccessType restriction

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 405 status code.

5.21.2 Exec on an ACL protected node

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	SCTS DM 1.2 as a server
Preconditions	None.
Pass-Criteria	The Test Object MUST return a 425 status code.

5.22 Device Management Client Conformance Test Group #22

5.22.1 Checking Implicit Addition of Interior node/s

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	SCTS DM 1.2 as a server
Test Code	
Preconditions	Parent Node of the child node to be added does not exist. An stablished DM session between Test Tool and DM Client. SCTS Tool has sufficient rights to add a node on the DM tree.
Test Procedure	1- SCTS Server sends an add command including thecomplete URI of the child node. 2- DM Clients Returns a 200 Status code. 3- SCTS 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.

5.23 Device Management Client Conformance TestGroup #23

5.23.1 Handling of Get with ‘list=TNDS’

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'. SCTS 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	SCTS DM 1.2 as a server
Test Code	
Preconditions	An stablished DM session between Test Tool and DM Client. SCTS Tool has sufficient rights on /DevDetail. /DevDetail node exists on the DM Client Tree and contains some sub nodes.
Test-Procedure	1-SCTS 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	SCTS should receive valid results in TNDS format (including ACLs, Format and Value) .

5.24 Device Management Client Conformance Test Group #24

5.24.1 Handling of copy command

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.
SCR Reference	DMREPPRO-PCE-C-005 Support for receiving 'Copy' command
Test Tool	SCTS DM 1.2 as a server
Test Code	
Preconditions	An stablished DM session between Test Tool and DM Client. SCTS Tool has sufficient rights on target node. SCTS Tool has sufficient rights on sorce node.
Test-Procedure	1- SCTS 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- SCTS 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 Support for Correlator

Test Case Id	DeviceManagement-v1.2-client-con-2501
Test Object	Client Device
Test Case Description	To check if the DM Client can support receiving and processing a Correlator.
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 Support for Correlator
Test Tool	SCTS DM 1.2 as a Server
Test Code	
Preconditions	An established DM session between Test Tool and DM Client. A node capable of receiving an exec node exists in the DM Tree of the DM Client. SCTS Server has sufficient rights to exec a node on that node of the DM Client. Test object is capable of supporting Correlator.
Test procedure	1- SCTS Server need to be configured to send an Exec to a specified node 2- SCTS Server sends an Exec command to the node with a Correlator. # Typically a process is being started at client by the EXEC command. # When the process is terminated, then: 3- DM Client returns a Generic Alert including the same correlator.
Pass-Criteria	-Client returns 200 for the EXEC

	-Correlator of Exec command and Generic Alert are the same.

5.26 Device Management Client Conformance Test Group #26

5.26.1 TLS support over HTTP transport

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 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 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
Test Tool	SCTS DM 1.2 as a server
Preconditions	Client Device supports HTTP. Credentials / certificates necessary to perform authentication have been provisioned in Test Object and Tool.
Test Procedure	1. DM client initiates a session with the Test Tool requesting transport layer authentication using TLS and indicating cipher settings. 2. Test Tool authenticates the DM client and sends the information the DM client needs to authenticate it. 3. DM client authenticates the Test Tool and the TLS session is established. 4. DM session is established between DM client and Test Tool
Pass-Criteria	- Test Tool authenticates the DM client - DM client authenticates the Test Tool - DM session is correctly established

5.26.2 SSL 3.0 support over HTTP transport

Test Case Id	DeviceManagement-v1.2-client-con-2602
Test Object	Client 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-C-003 Support for transport layer authentication DM-SEC-C-004 Support for HTTP transport 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
Test Tool	SCTS DM 1.2 as a server
Preconditions	Client Device supports HTTP. Credentials / certificates necessary to perform authentication have been provisioned in Test Object and Tool.
Test Procedure	1. DM client initiates a session with the Test Tool requesting transport layer authentication using SSL 3.0 and indicating cipher settings. 2. Test Tool authenticates the DM client and sends the information the DM client needs to authenticate it. 3. DM client authenticates the Test Tool and the SSL session is established. 4. DM session is established between DM client and Test Tool
Pass-Criteria	- Test Tool authenticates the DM client - DM client authenticates the Test Tool - DM session is correctly established

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 Server Layer Authentication with wrong credentials

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	SCTS DM 1.2 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 Server Layer Authentication with no credentials

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

6.2.2 Accepting of credentials

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 Processing the Replace command

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 Processing of the Alert command

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'

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 Get command on an existing interior node (Root Node, '.')

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 Add command to add a leaf node (./SCTSValue)

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.

6.5 Device Management Server Conformance Test Group #5

6.5.1 Replace command to replace the contents of a leaf node (./SCTSValue)

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 Delete command to delete a leaf node (./SCTSValue)

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 Handling of Multiple Messages

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 [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 Sequence command

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 Atomic command

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
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 Add a text Large Object (./SCTSLrgObjText)

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 Get a text Large Object (./SCTSLrgObjText)

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.
Pass-Criteria	Successfully Get the text based large object.

6.11 Device Management Server Conformance Test Group #11

6.11.1 Add a binary Large Object (./SCTSLrgObjBin)

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 Get a binary Large Object (./SCTSLrgObjBin)

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.

6.12 Device Management Server Conformance Test Group #12

6.12.1 Session Abort Alert

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 Get with '.?list=Struct'

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 Get with '.?list=StructData'

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.

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 UI Display Alert

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 UI Confirmation Alert

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.

6.17 Device Management Server Conformance Test Group #17

6.17.1 UI User Input Alert

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 Notification Initiated Session

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 Support of Generic alert

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

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 Support for sending and receiving Correlator

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

	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 TLS support over HTTP transport

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	5. 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 6. Server authenticates the Test Tool and sends the information the Test Tool needs to authenticate it. 7. Test Tool authenticates the Server and the TLS session is established. 8. DM session is established between Server and Test Tool 9. 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

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

6.21.2 SSL 3.0 support over HTTP transport

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

7. Device Management Interoperability Test Cases

7.1 MD-5 client authentication

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 MD-5 server authentication

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

Preconditions	None.
Test Procedure	1. 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 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.

7.3 Get on Leaf Node

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.

	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 Get on Non-Existant Node

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 Get on Interior Node

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

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 Get on Inaccessible Leaf Node

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'

	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 Replace on Permanent Leaf Node

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

	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 ACL Property

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.

	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 Connection Failure during DM session

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 Client Authentication - HMAC

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

	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 Server Authentication - HMAC

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.)

	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 Large Object/Multiple Commands

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.

	2. Synchronisation runs through with a basic DM authentication.

7.13 Notification Initiated Session

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 Bootstrap

7.14.1 Server Initiated Bootstrap. CP Profile

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 [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

7.14.2 Bootstrap from Smart Card, Client Provisioning Profile support

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
Specification Reference	[DMBOOT] Section 5.3 [DMBOOT] Appendix D
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)
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

7.14.3 Bootstrap from Smart Card, Device Management Profile support with TNDS objects

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 the 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)
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 smart card 4. DM client is able to successfully establish a DM session with the DM server indicated in the bootstrap message

7.14.4 Removal of account information when removing the Smart Card

Test Case Id	DeviceManagement-v 1.2-int-017
Test Object	Client and Smart Card device
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 and 5.4.6.
SCR Reference	DM-BOOT-C-006 DM-BOOT-C-007 DM-BOOT-C-008 DM-BOOT-C-009
Preconditions	• A DM client supporting bootstrap from the Smart Card • A DM client provisioned from a Smart Card and with the Smart Card inside the terminal
Test Procedure	6. Introduce a Smart Card containing bootstrap information 7. Verify that the DM client is provisioned with the corresponding account information contained in the Smart Card 8. Remove the Smart Card from the terminal. 9. Introduce another Smart Card with different bootstrap information
Pass-criteria	The DM client should remove from the DM management tree the account information corresponding to the DM server bootstrapped from the first Smart Card

7.14.5 DM Profile with TNDS – Transport Neutral Security: NETWORKID

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.	
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	

7.14.6 DM Profile with TNDS – Transport Neutral Security: USERPIN

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.	
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	

7.14.7 DM Profile with TNDS – Transport Neutral Security: USERPIN_NETWORKID

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 [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	

7.15 UI Alert – Display

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 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.

7.16 UI Alert – Confirmation

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. ■ 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.

7.17 UI Alert – Text Input

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. 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.

7.18 UI Alert – Single Choice

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.

7.19 UI Alert – Multiple Choice

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

7.20 Get Subtree Structure Without Data

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' 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.

7.21 Get Subtree Structure With Data

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). ■ 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.

7.22 Create new Application Settings

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.

7.23 Modify the Application Settings

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.

7.24 Delete the Application Settings

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. 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.

7.25 Inbox

7.25.1 Correct mapping of information in ./Inbox node

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"

7.25.2 Rejection of GET operations from a DM server on the ./Inbox node

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 *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".

7.26 DM Session set up between a DM 1.2 server and a DM 1.1.2 terminal

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. 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

7.27 Checking Implicit Addition of Interior node/s

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.

7.28 Handling of Get with 'list=TNDS'

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 [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) .

7.29 Handling of Copy command

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.
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.

7.30 Correlator

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

7.31 Support for adding a serialized management object

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

7.32 TLS 1.0 support

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 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	5. DM client initiates a session with the DM Server requesting transport layer authentication using TLS and indicating cipher settings. 6. DM Server authenticates the DM client and sends the information the DM client needs to authenticate it. 7. DM client authenticates the DM Server and the TLS session is established. 8. DM session is established between DM client and DM Server
Pass-Criteria	10. DM Server authenticates the DM client 11. DM client authenticates the DM Server 12. DM session is correctly established.

7.33 SSL 3.0 support

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.
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.

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
OMA-ETS-DM_INT-V1_2-20060524-C	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

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[
<SyncML xmlns='syncml:dmddf1.2'>
<MgmtTree>
        <VerDTD>1.2</VerDTD>
        <!-- <Man>The device manufacturer</Man> -->
        <!-- <Mod>The device model</Mod> -->
        <Node>
                <NodeName>DMAcc</NodeName>
                <RTProperties>
                        <Format>
```

```
                        <node/>
                </Format>
        </RTProperties>

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

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


        <Node>
                <NodeName>Name</NodeName>
                <RTProperties>
                        <Format>
                                <chr/>
                        </Format>
                        <Type> text/plain </Type>
                </RTProperties>
                <Value>Optional DM Server Displayable Name</Value>
        </Node>

        <Node>
                <NodeName>PrefConRef</NodeName>
                <RTProperties>
                        <Format>
                                <chr/>
                        </Format>
                        <Type>text/plain</Type>
                </RTProperties>
                <Value>Optional Reference to preferred connectivity information </Value>
        </Node>
```

```
<Node>
        <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> text/plain </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> text/plain </Type>
                </RTProperties>
```

```
                        <Value>Management Server Address</Value>
                    </Node>
                    <Node>
                        <NodeName>AddrType</NodeName>
                        <RTProperties>
                            <Format>
                                <chr/>
                            </Format>
                            <Type> text/plain </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> text/plain </Type>
                                </RTProperties>
                                <Value>Port Number</Value>
                            </Node>
                        </Node>
                    </Node>
                </Node>
            </Node>
        <Node>
            <NodeName>AauthPref</NodeName>
            <RTProperties>
                <Format>
                    <chr/>
                </Format>
                <Type>text/plain</Type>
            </RTProperties>
            <Value>Optional Preferred auth mechanism (see section 5.3.1.20 in [DMSTDOBJ]) </Value>
```

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

```
                                                <Format>
                                                        <chr/>
                                                </Format>
                                                <Type> text/plain </Type>
                                        </RTProperties>
                                        <Value>Auth Secret</Value>
                                </Node>
                                <Node>
                                        <NodeName>AAuthData</NodeName>
                                        <RTProperties>
                                                <Format>
                                                        <chr/>
                                                </Format>
                                                <Type> text/plain </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"/>
    <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>
```