Enabler Test Report Device Management V1.1.2

OMA TestFest (October 2004)
Version 29-Oct-2004

Open Mobile Alliance
OMA-Enabler_Test_Report-DM-112-20041029

This document is available online in PDF format at http://www.openmobilealliance.org/.

Known problems associated with this document are published at http://www.openmobilealliance.org/.

Comments regarding this document can be submitted to the Open Mobile Alliance™ in the manner published at http://www.openmobilealliance.org/documents.html

Contents

1. Scope

This report describes the results from the testing carried out at OMA TestFest (October 2004) concerning the Device Management enabler version 1.1.2.

2. References

2.1 Normative References

[OMAIOPPROC]	OMA Interoperability Policy and Process, http://www.openmobilealliance.org/
[DM112EICS]	Device Management version 1.1.2 Enabler Implementation Conformance Statement (EICS), http://www.openmobilealliance.org/
[ERELD]	OMA Device Management version 1.1.2 Enabler Release Definition
[DM112_SPEC]	OMA Device Management version 1.1.2 specifications
[EPTR]	Enabler Product Test Report
[ETP]	Enabler Test Report
[ETS]	Device Management version 1.1.2 Test Cases, OMA-ETS-DeviceManagement-v1.1.2-20031015-A, 15-Oct-2003

2.2 Informative References

3. Terminology and Conventions

3.1 Conventions

This is an informative document, i.e. the document does not intend to contain normative statements.

3.2 Definitions

SCTS	SyncML Conformance Test Suite.
Test Object	The implementation under test is referred 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

DM	Device Management
DSDM	Data Synchronization Device Managment
EICS	Enabler Implementation Conformance Statement
EPTR	Enabler Product Test Report
ETP	Enabler Test Plan
ETS	Enabler Test Specification
OMA	Open Mobile Alliance
PR	Problem Report
SCTS	Synchronization Conformance Test Suite

4. Summary

This report gives details of the testing carried out during the OMA TestFest (October 2004) for enabler Device Management version 1.1.2.

The report is compiled on behalf of OMA by NCC Group.

The work and reporting has followed the OMA IOP processes and policies [OMAIOPPROC].

5. Test Details

5.1 Documentation

This chapter lists the details of the enabler and any documentation, tools or test suites used to prove the enabler.

Date:	October 2004
Location:	Beijing, China
Enabler:	Device Management v1.1.2
Process:	OMA Interoperability Policy and Process [OMAIOPPROC]
Type of Testing	Interoperability Testing
Products tested:	Client-to-Server
Test Plan:	Device Management Version 1.1.2 Enabler Test Plan [ETP]
Test Specification:	Device Management Enabler Test Specification [ETS]
Test Tool:	SCTS 3.1.2
Test Code:	None
Type of Test event:	TestFest
Participants:	Aepona, Bitfone, Extended Systems, Inc; IBM, Insignia, Intellisync Corporation, Mobile Leader, Inc, Openwave Systems Ltd
Number of Client Products:	4
Participating Technology Providers for clients:	Extended Systems, Inc; Mobile Leader, Inc *2 other client participants*
Number of Server Products:	6
Participating Technology Providers for servers:	Aepona, Bitfone, IBM, Insignia, Intellisync Corporation, Openwave Systems Ltd
Number of test sessions completed:	24 of 24

5.2 Test Case Statistics

5.2.1 Test Case Summary

This chapter gives an overview of the result for all test cases included in [ETS].

The following status is used in the tables below:

- **Total number of TCs**: Used in the summary to indicate how many test cases there are in total.
- **Number of passed:** Used in the summary to indicate how many of the total test cases successfully passed.
- **Number of failed:** Used in the summary to indicate how many of the total test cases failed.
- **Number of N/A:** Used in the summary to indicate how many of the total test cases have not been run due to one of the implementations not supporting the functionality required to run this test case.
- **Number of OT:** Used in the summary to indicate how many of the total test cases have not been run due to no time to run the test case.
- **Number of INC:** Used in the summary to indicate how many of the total test cases have not been run due to functionality not being tested due to an error in the implementation or other functionality that is required to run this test case.

Data Types:	**Total number of TCs:**	**Number of test session:**	**Number of Passed:**	**Number of Failed:**	**Number of N/A:**	**Number of OT:**	**Number of INC:**
Client to Server TCs	26	24	340	9	199	73	3
Total	**26**	**24**	**340**	**9**	**199**	**73**	**3**

5.2.2 Test Case List

This chapter lists the statistics for all test cases included in [ETS].

The following status is used in the tables below:

- **No. of runs(R):** Used to indicate how many times the test cases have been run in total.
- **No. of passed(P):** Used to indicate how many times the test case has been run with successful result.
- **No. of failed(F):** Used to indicate how many times the test case has been run with failed result
- **No. of OT(O):** Used to indicate how many times the test case has not been run due to no time available.
- **No. of INC(I):** Used to indicate how many times the test case has not been run due to errors being found in other functionality required for running this test case.
- **PR:** Used to indicate if any PRs (Problem Reports) have been issued during testing.
- **Note:** Used to indicate the cause of Inconclusive or Fail verdicts.

Test Case:	Test Case Description:	R	P	F	O	I	PR:	Note:
DeviceManagement-v1.1.2-int-001	Purpose of this verification is to show compliance with Basic client authentication.	24	21	0	3	0		
DeviceManagement-v1.1.2-int-002	Purpose of this verification is to show compliance with MD-5 client authentication.	24	20	0	3	1		
DeviceManagement-v1.1.2-int-003	Purpose of this verification is to show compliance with MD-5 server authentication.	20	16	1	3	0		
DeviceManagement-v1.1.2-int-004	Purpose of this verification is to show compliance with the GET command on a leaf node.	24	21	0	3	0		
DeviceManagement-v1.1.2-int-005	Purpose of this verification is to show compliance with the GET command on a node that doesn't exist.	24	21	0	3	0		
DeviceManagement-v1.1.2-int-006	Purpose of this verification is to show compliance with the GET command on an interior node.	24	21	0	3	0		
DeviceManagement-v1.1.2-int-007	Purpose of this verification is to show compliance with the GET on an inaccessible leaf node.	24	21	0	3	0		
DeviceManagement-v1.1.2-int-008	Purpose of this verification is to show compliance with REPLACE on permanent leaf node.	24	21	0	3	0		

DeviceManagement-v1.1.2-int-009	Purpose of this verification is to show compliance with management node ACL behaviour.	24	21	0	3	0		
DeviceManagement-v1.1.2-int-010	Purpose of this verification is to show compliance with the error handling when connection failure occurs during the SyncML DM session.	24	19	0	3	2		
DeviceManagement-v1.1.2-int-011	Purpose of this verification is to show compliance with Basic server authentication.	18	15	0	3	0		Observation 001
DeviceManagement-v1.1.2-int-012	Purpose of this verification is to show compliance with HMAC client authentication.	17	11	2	4	0		
DeviceManagement-v1.1.2-int-013	Purpose of this verification is to show compliance with HMAC server authentication.	13	6	3	4	0		
DeviceManagement-v1.1.2-int-014	Purpose of this verification is to show compliance with the large object/multiple commands.	13	9	0	4	0		
DeviceManagement-v1.1.2-int-015	Purpose of this verification is to show compliance with notification initiated session.	6	3	0	3	0		
DeviceManagement-v1.1.2-int-016	Purpose of this verification is to show compliance with bootstrap.	5	2	0	3	0		
DeviceManagement-v1.1.2-int-017	Purpose of this verification is to show compliance with UI Display Alert	15	14	1	0	0		
DeviceManagement-v1.1.2-int-018	Purpose of this verification is to show compliance with UI Confirmation Alert.	11	10	1	0	0		
DeviceManagement-v1.1.2-int-019	Purpose of this verification is to show compliance with UI Text Input Alert.	5	5	0	0	0		
DeviceManagement-v1.1.2-int-020	Purpose of this verification is to show compliance with UI Single Choice Alert.	4	4	0	0	0		
DeviceManagement-v1.1.2-int-021	Purpose of this verification is to show compliance with UI Multiple Choice Alert.	2	2	0	0	0		

DeviceManagement-v1.1.2-int-022	Purpose of this verification is to show compliance with the server reading subtree structure without data from part of the management tree.	17	13	1	3	0		
DeviceManagement-v1.1.2-int-023	Purpose of this verification is to show compliance with the server reading subtree structure and data from part of the management tree.	12	8	0	4	0		
DeviceManagement-v1.1.2-int-024	Purpose of this verification is to verify creation of new Application Setting in client using DM server	17	12	0	5	0		
DeviceManagement-v1.1.2-int-025	Purpose of this verification is to verify modification of Application Settings in client using DM server.	17	12	0	5	0		
DeviceManagement-v1.1.2-int-026	Purpose of this verification is to verify deletion of Application Settings in client using DM server.	17	12	0	5	0		

5.2.3 Observations

The following issues were captured by the Trusted Zone during the OMA TestFest.

5.2.3.1 EICS issues

This section details issues with the DM v1.1.2 Enabler Implementation Conformance Statement (EICS) [DMEICS].

None.

5.2.3.2 Enabler Test Suite (ETS) issues

This section details issues with the Enabler Test Specification for OMA DM v1.1.2.

Observation: 001	
Document:	ETS for Device management v1.1.2 Approved Version, 15-Oct-2003
Test Case:	DeviceManagement-v1.1.2-int-010
Comment:	There was some ambiguity about the Server functionality for this test case. In one instant on a Power off of a client the Server does not remember/queue the previous commands such that when the session is re-initiated, it has no memory of previous protocol exchanges. In other instance it was assumed that the Server would have memory of the previous test session by caching the operations and resending it once client reconnect.
Recommendation:	Re-review the test case pass criteria.

5.2.3.3 DM v1.1.2 Specification issues

Observation: 002	
Comment:	Currently a client supports basic authentication only when server requests and the server also will use basic authentication only when client challenges. It's not configurable for both to start using basic authentication immediately.
Recommendation:	It was suggested that the specification should provide a means to distinguish whether each side should support basic authentication directly or support basic authentication only when challenged to do so.

Observation: 003	
Comment:	The following question was raised regarding the sequence command: It is assumed the sequence means that each command inside should be executed one by one. This implies all commands should be executed even if some fail (i.e. different from atomic, where when one command failed, others will abort, and then rollback).
Recommendation:	It seems that the specification does not clarify this issue and clarification is

	sought.

Observation: 004	
Comment:	The following question was raised concerning 'DevInfo': Should a DM client be required to send a "DevInfo" along with Package 1, when the client does not yet know what authentication mode to employ, and is waiting for a sever to "Challenge" it? The DMProtocol_v1.1.2, section 8.3 specifies: '*To send the device information (like manufacturer, model etc) to a Device Management Server as specified [DMSTDOBJ]. Client MUST send device information in the first message of management session.'* This indicates that a DevInfo is to be sent in package 1 by a client, even if the client / server have not been authenticated yet.
Recommendation:	A clarification is required to determine if it is permissible to send DevInfo before any authentication is done, and a challenge is expected.

Observation: 005	
Comment:	The following question was raised concerning 'DevInfo': Should a DM client be required to send a "DevInfo" along with Package 1, when the client does not yet know what authentication mode to employ, and is waiting for a sever to "Challenge" it? The DMProtocol_v1.1.2, section 8.3 specifies: '*To send the device information (like manufacturer, model etc) to a Device Management Server as specified [DMSTDOBJ]. Client MUST send device information in the first message of management session.'* This indicates that a DevInfo is to be sent in package 1 by a client, even if the client / server have not been authenticated yet.
Recommendation:	A clarification is required to determine if it is permissible to send DevInfo before any authentication is done, and a challenge is expected.

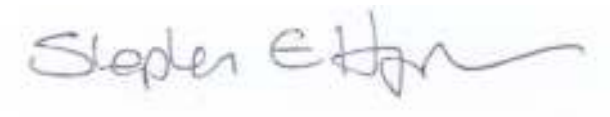
Observation: 006	
Comment:	The following question was raised concerning 'DevInfo': If a client sends a DevInfo in package 1, and the server sends a challenge in Package 2, should the client, in package 3, send back a DevInfo again (Replace on DevInfo)? If the client does not have to send (Replace) DevInfo again, it implies that the Server has to keep track of DevInfo for clients that are not "authenticated" yet.

Recommendation:	It is proposed that the specification is changed to indicate that the client has to resend DevInfo in Package 3.

Observation: 007	
Comment:	The following question was raised concerning a potential security issue: There is the Possibility of Denial of Service if client is not provisioned with Nonce and the server disseminates next nonce to clients that do not come with proper nonce. If a client has no nonce provisioned, and the MD5 check fails on server-side, the sever is required to ship a Next-nonce. The client is then expected to use this next nonce. Problem 1: Does this not open the door for Denial of service where a rogue client can constantly send a ping to the server, without any nonce, and force the server to send a next nonce. Problem 2: Does this not mean that a rogue client can force the server to send it a series of next nonces, and perhaps figure out the pattern of nonce generation, thereby compromising the man-in-the-middle attack prevention that nonces are expected to provide?
Recommendation:	Clarification sought.

6. Confirmation

This signature states that the included information is true and valid.



_____

Stephen Higgins – Device Management Trusted Zone

Appendix A. Change History (Informative)

Type of Change	Date	Section	Description