



Device Management Scheduling

Approved Version 1.0 – 19 Jul 2011

Open Mobile Alliance
OMA-TS-DM-Scheduling-V1_0-20110719-A

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

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

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

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

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

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

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE	4
2. REFERENCES.....	5
2.1 NORMATIVE REFERENCES.....	5
2.2 INFORMATIVE REFERENCES	5
3. TERMINOLOGY AND CONVENTIONS	6
3.1 CONVENTIONS	6
3.2 DEFINITIONS.....	6
3.3 ABBREVIATIONS	6
4. INTRODUCTION	8
5. SCHEDULING FRAMEWORK.....	9
5.1 SCHEDULE	9
5.1.1 Schedule Installation, Reconfiguration, and Removal	9
5.1.2 State Transitions of the Schedule	9
5.1.3 Types of the Schedule	9
5.2 SCHEDULING OPERATION.....	10
5.2.1 Initialization	11
5.2.2 Trigger Waiting.....	11
5.2.3 User Interaction.....	11
5.2.4 Task Execution.....	11
5.2.5 Gating.....	11
5.2.6 Status Reporting.....	12
5.3 SECURITY CONSIDERATIONS (INFORMATIVE).....	14
6. SCHEDULING MANAGEMENT OBJECTS	15
7. BEHAVIORS ASSOCIATED WITH THE MANAGEMENT OBJECT	25
7.1 STATE TRANSITION EXEC COMMAND SEMANTICS.....	25
7.2 TRIGGER FILTERING.....	25
7.3 USER INTERACTIONS.....	25
7.4 MISSED SCHEDULES AND OTHER RUN-TIME ERRORS	25
APPENDIX A. CHANGE HISTORY (INFORMATIVE).....	26
A.1 APPROVED VERSION HISTORY	26
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....	27
B.1 SCR FOR SCHEDULING MO.....	27
B.2 SCR FOR DM SCHEDULING ENABLER CLIENT AND AGENT	28
B.3 SCR FOR DM SCHEDULING ENABLER SERVER	29

Figures

Figure 1: Overall Scheduling Operation (informative)	10
Figure 2: Scheduling Management Objects.....	16

1. Scope

This document describes the Scheduling Framework and the Scheduling Management Objects, based on the requirements and the use cases included in [DMSCHED-RD] and the architecture described in [DMSCHED-AD].

As the rest of the OMA DM Management Object Enablers, the DM Scheduling Enabler builds and relies on the OMA Device Management v1.2 for the configuration, management, and execution of the Schedule.

2. References

2.1 Normative References

- [DiagMon-Function] “Diagnostics and Monitoring Functions Specification”, Open Mobile Alliance™, OMA-TS-DiagMon_Functions-V1_2,
URL: <http://www.openmobilealliance.org/>
- [DMPRO] “DM Protocol”, Version 1.2, Open Mobile Alliance™, OMA-TS-DM-Protocol -V1_0,
URL: <http://www.openmobilealliance.org/>
- [DMSCHED-AD] “DM Scheduling Architecture Document”, Version 1.0, Open Mobile Alliance™, OMA-AD-DM-Scheduling-V1_0,
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/>
- [iCal] “Internet Calendaring and Scheduling Core Object Specification”. F. Dawson, D. Stenerson, November 1998,
URL: <http://www.ietf.org/rfc/rfc2445.txt>
- [IOPPROC] “OMA Interoperability Policy and Process”, Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1,
URL: <http://www.openmobilealliance.org/>
- [ISO8601] Data elements and interchange formats – Information interchange – Representation of dates and times
- [META] “SyncML Meta Information Specification, version 1.2. Open Mobile Alliance™.
OMA-TS-SyncML_MetaInfo-V1_2.
URL: <http://www.openmobilealliance.org/>
- [OMA-DM] “OMA DM 1.2”, Version 1.2, Open Mobile Alliance™, OMA -DM -V1_2,
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>
- [RFC2234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997, URL:<http://www.ietf.org/rfc/rfc2234.txt>

2.2 Informative References

- [3GPP2] 3GPP2 C.S0002 Physical Layer Standard for cdma2000 Spread Spectrum Systems,
URL: <http://www.3gpp2.org/>
- [3GPP-NITZ] 3GPP TS 22.042 “3rd Generation Partnership Project; Network Identity and Timezone (NITZ); Service description, Stage 1 (Release 6)”, URL: <http://www.3gpp.org/>
- [OMA-Dic] “Dictionary for OMA Specifications V2.6”, Open Mobile Alliance™, OMA-Dictionary-V2_6,
URL: <http://www.openmobilealliance.org/>
- [RFC2030] IETF RFC 2030: “Simple Network Time Protocol (SNTP), Version 4 ”,
URL: <http://www.ietf.org/rfc/rfc2030.txt>
- [XMLENC] “XML Encryption Syntax and Processing”. W3C.
URL: <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>
- [XMLSIGN] “XML-Signature Syntax and Processing”. W3C.
URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>

3. Terminology and Conventions

3.1 Conventions

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

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Device	In this context, a Device is a voice and/or data terminal that uses a Wireless Bearer for data transfer. Device types may include (but are not limited to): mobile phones (GSM, CDMA, 3GSM, etc.), data-only terminals, PDAs, laptop computers, PCMCIA cards for data communication, and unattended data-only Devices (e.g., vending machines).
Device Description Framework	A markup language used to describe OMA DM object schema; may be used in a standardized specification to describe the characteristics of conformant implementations or published by a vendor to describe a particular device implementation.
Device Management	Management of the Device configuration and other managed objects of Devices from the point of view of the various Management Authorities. Device Management includes: <ul style="list-style-type: none"> Setting initial configuration information in Devices Subsequent updates of persistent information in Devices Retrieval of management information from Devices Processing events and alarms generated by Devices
Device Management Authority	Any legal entity authorized, either directly or through delegation, to perform management operations on a terminal using the OMA Device Management protocol through a set of management objects.
DM Client	An abstract software component in a Device implementation that conforms to the OMA Device Management Enabler static conformance requirements specified for DM Clients. It serves as an end-point of the DM Client-Server Protocols including the one described in this architecture document.
DM Server	An abstract software component in a deployed Device Management infrastructure that conforms to the OMA Device Management Enabler static conformance requirements specified for DM Servers. It serves as an end-point of the DM Client-Server Protocols including the one described in this architecture document.
DM System	A background system capable to interact with a (set of) Device(s) for the purpose of Device Management.
Management Object	A schema for configuration settings that an OMA DM client exposes to OMA DM servers for management operations defined in the OMA DM Enabler [OMA-DM].
Schedule	A collection of information which contains a list of scheduled management tasks, the condition specifying when those scheduled tasks are supposed to be executed, user interaction specifications, and status reporting specifications.
Status Reporting	It is the sending of Generic Alert to the DM server to inform the interested events occurred in the Device with regards to the Schedules under the control of the server.
Trap	The event that is generated by various sources in the Device which the Device Management System would be interested to know. The Trap source includes for example Radio Software, other OMA Enablers, or device drivers.

3.3 Abbreviations

DDF	Device Description Framework
DM	Device Management

DMS	Device Management Server
MO	Management Object
OMA	Open Mobile Alliance
SNTP	Simple Network Time Protocol
TCE	Threshold Crossing Event
UI	User Interface
UTC	Universal Time Coordinated

4. Introduction

The DM Scheduling Enabler v1.0 specifies the Scheduling Framework as well as its Management Objects that can be layered on top of the OMA DM v1.2 to seamlessly add the common scheduling capability to the OMA DM based management infrastructure.

With this capability, the OMA DM system is able to schedule management operations on the device, and have them executed offline when schedule – time-based or event-based – matches

The scheduling capability comes in handy when network operator or service provider needs to cause management operations to take place at the same time on large basis of the devices.

This capability also aims at reducing the management sessions and transactions with the built-in offline processing mechanisms, thereby decreasing management costs even further.

Finally, it is recommended that other management object enablers or general applications on the device take the advantage of this common scheduling capability as a building block rather than define their own scheduling mechanisms, only to create duplication and inconsistency among each other.

5. Scheduling Framework

In this section, the Scheduling Framework is described. If not specified otherwise, the terms 'Client' and 'Agent' are alternately used in the following sections to respectively refer to the Device Management Scheduling Enabler Client and the Device Management Scheduling Agent as defined in [DMSCHED-AD].

5.1 Schedule

A Schedule for the management operations is installed on the Device as an instance of the Scheduling Management Objects, and run by the Agent conforming to this specification. Each Schedule SHALL be owned by a single Management Authority. It means that the Scheduling Management Object MUST have value assigned or inherited to the ACL property including only one server identifier that owns the Schedule during normal operation. However, it is not prevented that the server with sufficient access to its parent node take control of the Schedule in abnormal situations, e.g. when the original server cannot properly handle it any more.

5.1.1 Schedule Installation, Reconfiguration, and Removal

The Schedules SHALL be installed only through the direct management operations over the DM sessions [DMPRO]. That is, it is not possible for one Schedule to install another Schedule. The same is true for the reconfiguration and removal of the Schedule that already exists on the Device. However, one Schedule MAY be activated or deactivated as a result of executing another Schedule. Any request from the server to install or reconfigure the Schedule MAY be verified by the Client as described in the upcoming sections. If the validity date is specified, the Schedule MUST be silently removed when it is expired.

5.1.2 State Transitions of the Schedule

A Schedule MUST always reside in one of the following two states:

- **Stopped State** - In this state, the Schedule is inactive and all the resources needed to run the Schedule might have been freed. The Schedule just installed MUST be in this state. Or the Schedule transitions into this state when the Scheduling Operation is finished or stopped.
- **Running State** - In this state, the Schedule is active and the Scheduling Operation is successfully initialized with all the necessary resources assigned to run the Schedule. The Agent MUST run the Schedule in this state, even after the Device recycles power.

The state transition is triggered by the use of the Exec command. The state toggles between Stopped and Running. See section 7.1 for more details.

The state transitions MAY also be triggered by the User, depending on the permission set by server, or automatically as the Schedule finishes or encounters any error. See section 5.2.3 and the other upcoming sections for more description.

5.1.3 Types of the Schedule

Depending on the type of the trigger, a Schedule may fall into one of the three types: time-based, event-based, and combined type.

5.1.3.1 Time-based

Time-based Schedule MUST be supported. The date and time with the OPTIONAL recurrence rule specify when the Schedule match occurs.

For correct operation, the local device time and the server time needs to be synchronized within a reasonable tolerance. It is assumed that the time synchronization is already achieved by out-of-band mechanisms and any other mechanism will not be specified in this specification. In practice, it would be sufficient for the server to know that the device supports reliable time clock synchronized by well-known network time sources such as CDMA system time [3GPP2], 3GPP Network Identity and

Timezone (NITZ) [3GPP-NITZ], Simple Network Time Protocol (SNTP) [RFC2030], etc. Or, in some other cases where network time source is not used, it is possible to use less reliable methods to achieve the time synchronization.

Server SHALL specify scheduling events using either a time relative to UTC or in local time zone. If specified in UTC, Client MUST interpret the schedule relative to UTC independent of time zone or daylight saving time adjustments. If specified in local time zone, Client MUST interpret the schedule as local time in the current time zone and subject to any daylight saving time adjustments.

5.1.3.2 Event-based

Event-based Schedule MAY be supported. Trap identifier specifies the trigger. The Agent SHALL be able to receive the Trap Notification [TRAP]. The detailed mechanism for interfacing with Trap source is outside the scope of this specification.

5.1.3.3 Combined type

The combined type Schedule is the one in which different types of triggers are specified at the same time. Note that whenever different triggers are specified, they are combined using OR operation. See the description for the <x>/Trigger node in section 6.

5.2 Scheduling Operation

The DM Scheduling Agent MUST be able to run the Schedule. Specifically, running a Schedule means performing the Scheduling Operation for it as described this section.

The Scheduling Operation consists of the following stages: Initialization, Trigger Waiting, User Interaction, Task Execution, Gating, and Status Reporting. Basically, after successful initialization, it is straight forward operation – receive the triggers, execute the scheduled tasks, and then report the results. The Agent SHALL be able to perform this basic Scheduling Operation.

Figure 1 summarizes the overall Scheduling Operation process.

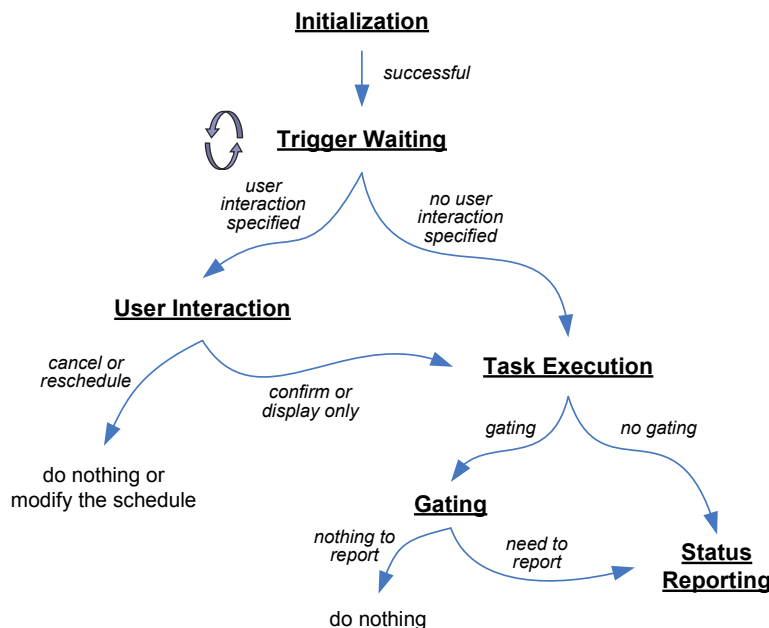


Figure 1: Overall Scheduling Operation (informative)

5.2.1 Initialization

Whenever the Schedule is set to Running, it always starts in this stage, where the DM Scheduling Agent uses the schedule informations to initialize the Scheduling Operation so that that the following stages will run correctly. For instance such tasks as checking the integrity of the Schedule, detecting wrong configuration, assigning resources, and configuring the trigger sources - timer or Traps - to send triggers are performed. The detailed mechanisms of the initialization are implementation specific.

5.2.2 Trigger Waiting

In this stage, the Agent waits for the triggers. When the trigger is received, the Agent first needs to verify the triggers for security reasons as outlined in section 5.3 before moving on to the next stage in the Scheduling Operation. The Trigger Filtering mechanism MAY be used to filter out the triggers based on the current value of other management objects. The filtered triggers are silently discarded. See section 7 for more details on this filtering mechanism.

5.2.3 User Interaction

Some optional localized user interactions are specified. Different from the OMA-DM User Interactions [DMPRO], these user interactions are done locally between the User and the Device, and do not directly involve DM sessions.

- Display User Notification – display notification message on the screen.
- Confirm – ask the User whether to confirm or cancel the execution of the scheduled task.
- Reschedule – allow the User to modify the time of the schedule.
- State Transition of the Schedule – allow the User to stop the operation of Schedule.

When schedule matches and if the User cancels or reschedules the execution of the scheduled tasks, the Task Execution SHALL be skipped. Note that explicit confirmation is required. Therefore, no response from the User for a certain time period SHALL NOT be considered confirmation, and the length of such time period is implementation specific. When the rescheduling is allowed, it implies that the User is given greater flexibility in setting up her or his own schedule in as many different ways as the device allows. In this case, for example, the User can schedule the task to run automatically regardless of the content of `<x>/UI/Confirm` node, because the explicit confirmation can be given as she or he set up the schedule.

5.2.4 Task Execution

The OMA DM message represented in clear-text XML or in binary WBXML MAY be specified [OMA-DM]. If the OMA DM message is specified, it MUST be processed in this stage by the underlying OMA DM Client as specified in [OMA-DM]; therefore, the DM Scheduling Agent simply forwards the OMA DM message to the OMA DM Client. Note that the OMA DM message specified in the Schedule is processed offline, not as a part of session. Therefore, some OMA DM features, eg Large Object, that require multiple messages exchange such as Large Object will not be useable. Similarly, the use of credentials in `SyncHdr` is also discouraged because it is prone to failure from staled nonce.

In addition to the OMA DM message, a connection to a DM server MAY be scheduled. If this particular action is specified, the Status Reporting message MUST be sent indicating the alert type `Connect` as specified in section 7. Unsuccessful completion of this action, after a certain number of retry, MAY be silently ignored.

5.2.5 Gating

As a result of processing the scheduled DM message in the Task Execution stage, a response message is generated. By default, when no gating rule is specified, it MUST be sent to the server in the Status Reporting message of 'Result' type. However, if any gating rule is specified in the Schedule, all or part of the response message may be gated-off. This OPTIONAL Gating function would be useful in the wireless environment where millions of end-user devices are being managed because it allows the server to control the bandwidth by reducing the number of the client initiated management sessions.

The gating rule specifies the references and code values for the responses to be gated-off. Currently, only the `Status` command MAY be gated-off. When the `Status` command is returned from the DM Client corresponding to each processed DM command, the DM Scheduling Agent MUST compare the `CmdRef` value and the Status Response Code in `Data` element in the `Status` command against the gating rules searching for matches, and the matched Status command SHALL be gated-off and SHALL NOT be included in the Status Reporting message. If there is no response to be included in the Status Reporting message, the message SHALL NOT be sent.

5.2.6 Status Reporting

While the Schedule is running, the server may need to know the events, errors, or status changes that happened to the Schedule. On the other hand, the responses or results from the Task Execution need to be sent to the server unless they are gated-off and the Client needs to initiate a session to the server if that particular action is scheduled.

The Status Reporting is designed, based on the Generic Alert mechanism and message template [OMA-DM], to provide a reporting mechanism in the above three cases.

If `<x>/Server`, which is used to explicitly designate the owner of the schedule, is specified, the Status Reporting SHALL be sent to the specified server identifier.

5.2.6.1 Alert Types

The alert types MUST be specified in `Type` in `Meta` element of the Generic Alert message. The following alert types MUST be used to identify the type of the Status Reporting messages.

Type Name	Contents	Usage	Status
Event	"um:oma:at:scheduling:1.0:event"	The alert type MUST be used if the message contains the Status Reporting code in the <code>Data</code> field.	OPTIONAL
Result	"um:oma:at:scheduling:1.0:result"	The alert type MUST be used if the message contains the responses to the management commands in the <code>Data</code> field.	REQUIRED
Connect	"um:oma:at:scheduling:1.0:connect"	The alert type MUST be used if the message is used for the scheduled connection to the server initiated from the client.	OPTIONAL

5.2.6.2 Source

The source of the Status Reporting message MUST be specified in `LocURI` in `Source` element of the Generic Alert message, using the URI of the root placeholder node of the corresponding Schedule. Using this source information, the server is able to identify to which Schedule this Status Reporting message is linked.

5.2.6.3 Data

For the messages of 'Event' type, the Status Reporting Code MUST be specified in `Data` in `Item` element of the Generic Alert message, using an integer code among the list below:

Status Reporting Code	Meaning

(500) Run-Time Error	The Run-Time errors are encountered while running the Schedule, which is stopped as a result of the errors.
(501) Schedule Missed	The Schedule was missed.
(502) User Stopped	The Schedule was forced to stop by the User or for some other reasons.
(503) User Cancelled	User cancelled (or deferred) the task execution.
(504) User Rescheduled	User rescheduled.

By default, these events listed in the table above are not reported. But if the event reporting rule specifies one or more of these events, such events SHALL be reported. Every time the event occurs, the <x>/SR/Code node MUST be reflected with the latest event code among those shown in the table, whether it is reported or not.

For the message of 'Result' type, the content of the Data element in Item element MUST specify the OMA DM message, either in XML format or WBXML format, if no gating rule is specified. Otherwise, it MUST specify in any order only the Status and Result commands that are not gated-off among those generated from all the commands in scheduled OMA DM message including the SyncHdr.

For the message of 'Connect' type, the content of the Data element in Item element has no meaning and SHOULD be left blank.

Following is an example of the Status Reporting message of 'Result' type when gating rule is specified:

```
<Alert>
  <CmdID>2</CmdID>
  <Data>1226</Data> <!-- Generic Alert -->
  <Item>
    <Source><LocURI>./Schedule/ClientUpdate</LocURI></Source>
    <Meta>
      <Type xmlns="syncml:metinf">
        urn:oma:at:scheduling:1.0:result<!-- Result Type -->
      </Type>
      <Format xmlns="syncml:metinf">chr</Format>
    </Meta>
    <Data> <!-- Status, Results here -->
      <Status>
        <MsgRef>1</MsgRef>
        <CmdID>1</CmdID>
        <CmdRef>0</CmdRef>
        <Cmd>SyncHdr</Cmd>
        <Data>200</Data> <!-- SyncHdr OK -->
      </Status>
      <Status>
        <MsgRef>1</MsgRef>
        <CmdID>2</CmdID>
        <CmdRef>1</CmdRef>
        <Cmd>Execute</Cmd>
        <Data>500</Data> <!-- Command Failed -->
      </Status>
    </Data>
  </Item>
</Alert>
```

5.2.6.4 Usage of Other Fields

Correlator field MUST NOT be used. The use of Mark in Meta element is up to implementation.

5.3 Security Considerations (Informative)

Since the Server can be authenticated, as specified in [DMSEC], during the DM Session to install or reconfigure a Schedule, and the content of the `<x>/Task/XML` or `<x>/Task/Binary`, the OMA DM message to be executed during the Task Execution stage, is being delivered just as an opaque content as a part of the DM Session, the OMA DM message doesn't need its own credentials for authentication. While in storage, however, the OMA DM message might be vulnerable to undesirable exposure or modification. To combat this problem, the OMA DM message may be protected by content signing or encryption method such as [XMLSIGN] or [XMLENC] as described in Appendix C of [DMSEC].

Also consider that, depending on the device platform, it might be safer to implement any mechanism to make sure that the received triggers, especially Trap Notifications, are not modified en route and originated from the trusted sources. The `<x>/Trap/<x>/Ext` node might be used for that purpose.

6. Scheduling Management Objects

The Scheduling management objects associated with each Schedule are assembled under the placeholder node x (dynamically or statically created) as shown in figure 2.

Management Object identifier: urn:oma:dm:scheduling:1.0

Protocol Compatibility: This object is compatible with OMA Device Management protocol specifications, version 1.2

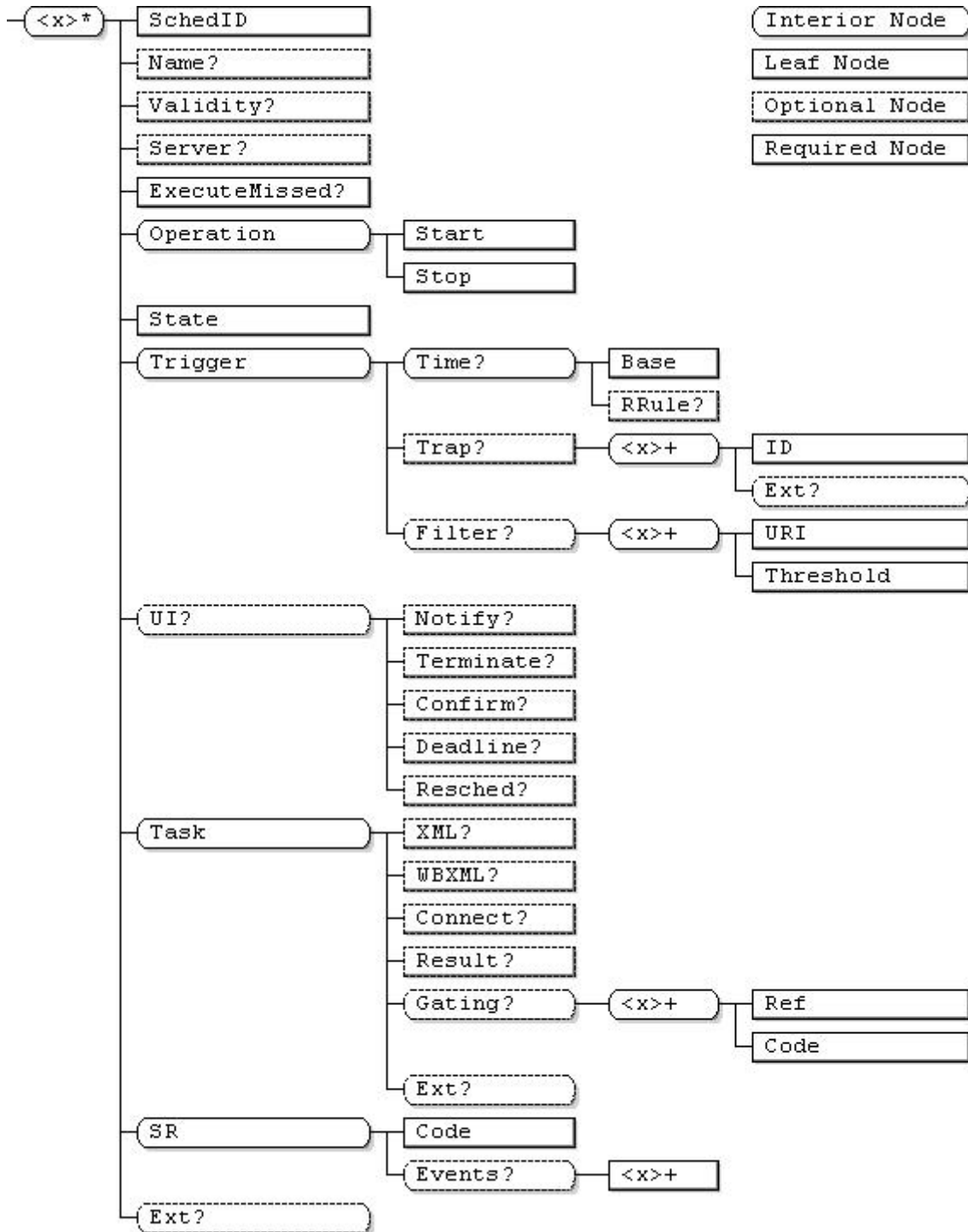


Figure 2: Scheduling Management Objects

<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for a Scheduling management objects. The ancestor elements of this node define the position in the management tree of this management object. The name of this node will be assigned when it is created at run-time except when this node is statically created.

<x>/SchedID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the Schedule ID of the Schedule.

<x>/Name

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the human readable name of the Schedule.

<x>/Validity

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the expiration time and date for the Schedule. The Schedule is valid only before the time and date specified in this node. The content is expressed in complete representation, basic format as specified in ISO 8601. The time MUST be in UTC based time. If time is not specified along with the date, it implies the end of the day.

<x>/Server

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node contains the server identifier of the DM Server that owns the Schedule.

<x>/ExecuteMissed

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

The ExecuteMissed node specifies handling of schedules that could not be executed at the specified time. If the node is missing or its value is 'false', the schedule MUST be ignored. If the value is 'true', the schedule MUST be retroactively executed as soon as practical.

<x>/Operation

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This node is a placeholder node for the state transition operations and informations associated with the Schedule.

<x>/Operation/Start

Status	Occurrence	Format	Min. Access Types
Required	One	null	Exec

This leaf node is used as a target of the Exec command to start the running of Schedule.

<x>/Operation/Stop

Status	Occurrence	Format	Min. Access Types
Required	One	null	Exec

This leaf node is used as a target of the Exec command to stop the running of Schedule.

<x>/State

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node reflects the operational state of the Schedule. The possible values are Stopped and Running.

<x>/Trigger

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node specifies the configurations for the triggers and filters. Two different types of trigger MAY be specified: Time or Event. Under this placeholder, more than one trigger configurations MAY be specified. If multiple configurations are specified, it implies that they are combined using the OR operation.

<x>/Trigger/Time

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node serves as a placeholder for the Time-based trigger, which consists of base time and recurrence rule.

<x>/Trigger/Time/Base

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the base value of time trigger expressed in the [ISO 8601] complete representation, basic format for the calendar date and time. Either the date with UTC time form or date with local time form MAY be used. This base value specifies at what point in time the timer trigger MUST be generated.

Example:

20070820T080000Z ; August 20, 2007, at 0800 UTC

20070820T230000 ; August 20, 2007, at 2300 local time

<x>/Trigger/Time/RRule

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the recurrence rule specification. The recurrence rule is used, together with the base value of the time trigger, to generate the recurrence set, using the methods specified in [iCal]. Each element of the recurrence set indicates the points in time at which the time trigger MUST be generated with the base value of the time trigger being the first instant of the set. The content of this node MUST be the string that conforms to the format and the recurrence grammar defined for the RECUR data type in [iCal].

Example:

FREQ=DAILY;COUNT=10 ; daily for 10 occurrences

<x>/Trigger/Trap

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This interior node acts as a placeholder for the Event-based triggers that use Traps as source [TRAP].

<x>/Trigger/Trap/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node specifies the registration information for a Trap [TRAP]. If multiple triggers are specified, they are combined using OR operation.

<x>/Trigger/Trap/<x>/ID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the Trap Identifier [TRAP].

<x>/Trigger/Trap/<x>/Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is a placeholder for additional parameters for Trap configuration.

<x>/Trigger/Filter

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node acts as a placeholder to contain the Trigger Filters configurations.

<x>/Trigger/Filter/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node is a placeholder for a particular Trigger Filter configuration.

<x>/Trigger/Filter/<x>/URI

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the URI of the reference node in the referenced management object to be sampled for comparison with the given threshold, e.g. this node specifies the URI of *<x>/DiagMonData/<x>/BatteryLevel* node defined in [DiagMon-Function].

<x>/Trigger/Filter/<x>/Threshold

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the threshold. The threshold is compared with the value of the referenced node specified by *<x>/Trigger/Filter/<x>/URI*, e.g. the threshold is a percentage when the reference node is *<x>/DiagMonData/<x>/BatteryLevel* defined in [DiagMon-Function]. The content of this node MUST be a text string encoded for threshold formats such as integer, boolean, float, character, date and time, and the transfer-encoding method described in section 5.2.4 of SyncML Common Meta-Information [META] MUST be used. Other threshold formats are not supported. This string SHALL be transformed into the format equal to that of the reference management object before it is compared with the sample value of the reference.

<x>/UI

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is the place holder to include the user interaction rules.

<x>/UI/Notify

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the user notification message to be displayed before or while executing the scheduled management tasks.

<x>/UI/Terminate

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get

This leaf node specifies whether the User is allowed to stop (terminate) the Schedule. If this node is not present, it implies “false” for this node.

<x>/UI/Confirm

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get

This leaf node specifies whether the user confirmation is required to execute the scheduled tasks. Failing to get confirmation, the Agent has to skip the execution. If this node is not present, it implies “false” for this node.

<x>/UI/Deadline

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the date and time until which the user is allowed to defer or reschedule the execution of the task. The value of this node MUST be a string for combinations of date and time expressed in complete representation, basic format as specified in ISO 8601.

<x>/UI/Resched

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get

This leaf node specifies whether the user is allowed to modify the Schedule in terms of time.

<x>/Task

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is a placeholder to contain the scheduled tasks.

<x>/Task/XML

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	xml	Get

This leaf node specifies the OMA DM message represented in clear-text XML. See [REPPRO]. The content of this node may also be an OMA DM message protected by content signalling or encryption method. This node MUST NOT be specified when the <x>/Task/WBXML node is present.

<x>/Task/WBXML

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bin	Get

This leaf node specifies the OMA DM message represented in binary WBXML. See [REPPRO]. This node MUST NOT be specified when the <x>/Task/XML node is present.

<x>/Task/Connect

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get

This leaf node specifies whether the Client initiate a connection to the owning server when the schedule matches. See section 5.2.6. If this node is not present, it implies “false” for this node.

<x>/Task/Result

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	xml	Get

This leaf node specifies the OMA DM message resulting from executing the scheduled OMA DM message.

<x>/Task/Gating

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is the placeholder for the gating rules. See section 5.2.5 and 5.2.6.

<x>/Task/Gating/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node specifies a gating rule.

<x>/Task/Gating/<x>/Ref

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the reference to the response command to be gated-off. The content of this node is compared with the value of the `CmdRef` element in the `Status` commands. It is possible to use the wildcard character `*` to imply any value used for `CmdRef` element.

<x>/Task/Gating/<x>/Code

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the code of the response command to be gated off. The content of this node is compared with the Status Response Code in the `Status` command. It is possible to use the wildcard character `*` or `?` to imply all and any digit from 0 to 9 respectively, eg. `5??` means any code from 500 to 599.

<x>/Task/Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node is a placeholder for the platform or vendor specific extensions.

<x>/SR

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the placeholder for the event reporting rules and the gating rules. See section 5.2.5, 5.2.6, and 7.

<x>/SR/Code

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains the Status Reporting code for the latest events that have occurred. The server will be able to poll this node to see what happened to the Schedule in case the event reporting was not done or failed.

<x>/SR/Events

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is the placeholder for the event reporting rules. See section 5.2.6 and section 7.

<x>/SR/Events/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	chr	Get

This leaf node specifies a 3 digit Status Reporting Code. It is possible to use the wildcard character “*” to imply all or “?” to imply any digit from “0” to “9”. Any events, from the table in section 5.2.6.3, that has a corresponding code specified in this node SHALL be reported to server.

<x>/Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This optional interior node is a placeholder for the platform or vendor specific extensions.

7. Behaviors associated with the Management Object

7.1 State Transition Exec Command Semantics

Two state transition nodes, `<x>/Operation/Start` and `<x>/Operation/Stop`, are provided, respectively, to start and stop running the Schedule through the management operations using Exec command. Therefore, the server compliant to DM Scheduling Enabler MUST support OMA DM Exec command.

When the Exec command is sent to `<x>/Operation/Start` node, the Client MUST attempt to start the Scheduling Operation and then send status command for the Exec command back to the server after the attempt is done. If the attempt is failed because the Schedule is already running, the status code (403) `Forbidden` SHALL be returned. In all other cases, other status codes valid for the Exec command can be used as specified in [DMPRO].

When the Exec command is sent to `<x>/Operation/Stop` node, the Client MUST attempt to stop the Scheduling Operation and then send status command for the Exec command back to the server after the attempt is done. If the attempt is failed because the Schedule is already stopped, the status code (403) `Forbidden` SHALL be returned. In all other cases, other status codes valid for the Exec command can be used as specified in [DMPRO].

After successful state transitions the new state value MUST be reflected into `<x>/State` node.

7.2 Trigger Filtering

The Trigger Filter is based on the current value the reference node in of the management objects sampled and compared with the thresholds given along with the URI of the referred management objects. If the value of the reference node doesn't match, triggers SHALL be silently discarded.

Note that when the management objects are referred, the access rights of the server are checked. If it turns out that the server doesn't have sufficient access right to refer to the value of the management objects, (500) `Run-Time Error` SHALL be generated.

In addition, if the management object doesn't exist at the given URI or the URI cannot be resolved successfully to refer to existing management object, (500) `Run-Time Error` SHALL be generated.

7.3 User Interactions

The result of the User Interactions affects the state of the Scheduling Operation as mentioned in section 5. The new state MUST be reflected into the `<x>/State` node.

If the User does not act within a predefined time after being asked to confirm the execution of the scheduled tasks, it is regarded as he canceled the offer as mentioned earlier in section 5.2.3. The minimum and maximum times for which the User action will be waited and the user notification message will be display are implementation specific.

7.4 Missed Schedules and Other Run-Time Errors

It is possible that the schedules may be missed for some reasons. Here is an example: if the Device has a Time-based Schedule, a schedule match might occur while the Device is switched off, and might be detected as the Device is switched back on. The missed schedule SHALL NOT affect the Scheduling Operation. That is, the Schedule SHALL continue to run even though such missed schedule is detected. The Device SHOULD be able to detect the missed schedule matches, but detailed mechanism of detecting the them is implementation dependent.

Other various errors encountered during each stage of Scheduling Operation, such as the one described in section 7.2, SHALL generate (500) `Run-Time Error`, and the Schedule SHALL be stopped. Reportings of these run-time errors are described in section 5.2.6.

Appendix A. Change History

(Informative)

A.1 Approved Version 1.0 History

Reference	Date	Description
OMA-TS-DM-Scheduling-V1_0-20110719-A	19 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0259-INP_Scheduling_V1_0_ERP_for_final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

B.1 SCR for Scheduling MO

Item	Function	Reference	Requirement
DMSched-MO-T-001-M	Support for the correct MO Identifier for the Scheduling MO	Section 6	
DMSched-MO-T-002-M	Support for the SchedID node	Section 6	
DMSched-MO-T-003-O	Support for the Name node	Section 6	
DMSched-MO-T-004-O	Support for the Validity node	Section 6	
DMSched-MO-T-005-O	Support for the Server node	Section 6	
DMSched-MO-T-006-M	Support for the ExecuteMissed node	Section 6	
DMSched-MO-T-007-M	Support for the Operation node	Section 6	
DMSched-MO-T-008-M	Support for the State node	Section 6	
DMSched-MO-T-009-M	Support for the Trigger node	Section 6	
DMSched-MO-T-010-O	Support for the UI node	Section 6	DMSched-MO-T-025-O OR DMSched-MO-T-026-O OR DMSched-MO-T-027-O OR DMSched-MO-T-028-O OR DMSched-MO-T-029-O
DMSched-MO-T-011-M	Support for the Task node	Section 6	DMSched-MO-T-030-O OR DMSched-MO-T-031-O
DMSched-MO-T-012-M	Support for the SR node	Section 6	
DMSched-MO-T-013-M	Support for the Ext node	Section 6	
DMSched-MO-T-014-M	Support for the Operation/Start node	Section 6	
DMSched-MO-T-015-M	Support for the Operation/Stop node	Section 6	
DMSched-MO-T-016-M	Support for the Trigger/Time node	Section 6	
DMSched-MO-T-017-M	Support for the Trigger/Time/Base node	Section 6	
DMSched-MO-T-018-O	Support for the Trigger/Time/RRule node	Section 6	
DMSched-MO-T-019-O	Support for the Trigger/Trap node	Section 6	DMSched-MO-T-020-O
DMSched-MO-T-020-O	Support for the Trigger/Trap/<x>/ID node	Section 6	
DMSched-MO-T-021-O	Support for the Trigger/Trap/<x>/Ext node	Section 6	
DMSched-MO-T-022-O	Support for the Trigger/Filter node	Section 6	DMSched-MO-T-023-O AND DMSched-MO-T-024-O
DMSched-MO-T-023-O	Support for the Trigger/Filter/<x>/URI node	Section 6	
DMSched-MO-T-024-O	Support for the Trigger/Filter/<x>/Threshold node	Section 6	
DMSched-MO-T-025-O	Support for the UI/Notify node	Section 6	
DMSched-MO-T-026-O	Support for the UI/Confirm node	Section 6	
DMSched-MO-T-027-O	Support for the UI/Deadline node	Section 6	

Item	Function	Reference	Requirement
DMSched-MO-T-028-O	Support for the UI/Reschedule node	Section 6	
DMSched-MO-T-029-O	Support for the UI/Terminate node	Section 6	
DMSched-MO-T-030-O	Support for the Task/XML node	Section 6	
DMSched-MO-T-031-O	Support for the Task/WBXML node	Section 6	
DMSched-MO-T-032-O	Support for the Task/Connect node	Section 6	
DMSched-MO-T-033-O	Support for the Task/Result node	Section 6	
DMSched-MO-T-034-O	Support for the Task/Ext node	Section 6	
DMSched-MO-T-035-O	Support for the Task/Gating nodes	Section 6	DMSched-MO-T-036-O AND DMSched-MO-T-037-O
DMSched-MO-T-036-O	Support for the Task/Gating/<x>/Ref node	Section 6	
DMSched-MO-T-037-O	Support for the Task/Gating/<x>/Code node	Section 6	
DMSched-MO-T-038-M	Support for the SR/Code node	Section 6	
DMSched-MO-T-039-O	Support for the SR/Events node	Section 6	
DMSched-MO-T-040-O	Support for the SR/Events/<x> node	Section 6	

B.2 SCR for DM Scheduling Enabler Client and Agent

Item	Function	Reference	Requirement
Schedule SCR			
DMSched-SCH-C-001-M	Install/Reconfig/Remove operations	Section 5.1.1	
DMSched-SCH-C-002-M	Support for Running and Stopped states	Section 5.1.2	
DMSched-SCH-C-003-O	Support for Schedule Validity	Section 5.1.1 Section 6	
DMSched-SCH-C-004-M	Support for the single-shot Time-based Schedule	Section 5.1.3.1 Section 6	
DMSched-SCH-C-005-O	Support for Recurrence Rule (iCalendar)	Section 5.1.3.1 Section 6	
DMSched-SCH-C-006-O	Support for Event-based Schedule	Section 5.1.3.2	
Operation SCR			
DMSched-OPR-C-001-M	Capable of performing basic Scheduling Operation	Section 5.2 Section 6	
DMSched-OPR-C-002-O	Support for Trigger Filtering	Section 5.2.2 Section 6 Section 7.2	DMSched-BHV-C-002-O
DMSched-OPR-C-003-M	Support for Task Execution	Section 5.2.4	DMSched-OPR-C-005-O OR DMSched-OPR-C-006-O
DMSched-OPR-C-004-O	Support for User Interactions	Section 5.2.3 Section 6	
DMSched-OPR-C-005-O	Support for handling OMA DM message in XML format	Section 5.2.4 Section 6	
DMSched-OPR-C-006-O	Support for handling OMA DM message in WBXML format	Section 5.2.4 Section 6	

Item	Function	Reference	Requirement
DMSched-OPR-C-007-O	Support for Connection to Server	Section 5.2.4	DMSched-OPR-C-010-O
DMSched-OPR-C-008-O	Support for Gating	Section 5.2.5 Section 6	
DMSched-OPR-C-009-M	Support for Status Reporting – Result type	Section 5.2.6	
DMSched-OPR-C-010-O	Support for Status Reporting – Event type	Section 5.2.6 Section 6	
DMSched-OPR-C-011-O	Support for Status Reporting – Connect type	Section 5.2.6	
DMSched-OPR-C-012-M	Capable of storing the latest event code	Section 5.2.6	
DMSched-OPR-C-013-O	Capable of storing the OMA DM message response resulting from the latest execution.	Section 6	
Behavior SCR			
DMSched-BHV-C-001-M	Support for the State Transition Exec Commands	Section 7.1	
DMSched-BHV-C-002-O	ACL rule compliance for Trigger Filtering operation	Section 7.2	
DMSched-BHV-C-003-M	Support for missed Schedule and Run-Time error handling	Section 7.4	
DMSched-BHV-C-004-M	Support for Missed Schedules execution	Section 6 Section 7.4	

B.3 SCR for DM Scheduling Enabler Server

Item	Function	Reference	Requirement
Schedule SCR			
DMSched-SCH-S-001-M	Install/Reconfig/Remove operations	Section 5.1	
DMSched-SCH-S-002-M	Support for the single-shot Time-based Schedule	Section 5.1.3.1 Section 6	
DMSched-SCH-S-003-O	Support for Recurrence Rule (iCalendar)	Section 5.1.3.1 Section 6	
Operation SCR			
DMSched-OPR-S-001-M	Receiving Status Reporting message – Result type	Section 5.2.6	
DMSched-OPR-S-002-O	Receiving Status Reporting message – Event type	Section 5.2.6	
DMSched-SCH-S-003-O	Receiving Status Reporting message – Connect type	Section 5.2.6	
Behavior SCR			
DMSched-BHV-S-001-M	Support for the State Transition Exec commands	Section 7.1	