



# **DM Client Side API Framework (DMClientAPIfw)**

Candidate Version 1.0 – 04 Oct 2011

---

**Open Mobile Alliance**  
OMA-ER-DMClientAPIfw-V1\_0-20111004-C

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.....	5
2.	REFERENCES .....	6
2.1	NORMATIVE REFERENCES .....	6
2.2	INFORMATIVE REFERENCES .....	6
3.	TERMINOLOGY AND CONVENTIONS.....	7
3.1	CONVENTIONS .....	7
3.2	DEFINITIONS.....	7
3.3	ABBREVIATIONS .....	7
4.	INTRODUCTION .....	8
5.	REQUIREMENTS (NORMATIVE).....	9
5.1	HIGH-LEVEL FUNCTIONAL REQUIREMENTS .....	9
5.2	DM-7 (REGISTRATION) API REQUIREMENTS.....	9
5.3	DM-8 (NOTIFICATION) API REQUIREMENTS .....	9
5.4	DM-9 (INTERACTION) API REQUIREMENTS.....	10
6.	ARCHITECTURAL MODEL .....	11
6.1	DEPENDENCIES.....	11
6.2	ARCHITECTURAL DIAGRAM .....	11
6.3	FUNCTIONAL COMPONENTS AND INTERFACES/REFERENCE POINTS DEFINITION.....	11
6.3.1	Protocol Endpoint .....	11
6.3.2	Interfaces.....	11
6.3.2.1	DM-7 Client-App Registration .....	11
6.3.2.2	DM-8 Client-App Notification.....	11
6.3.2.3	DM-9 Client-App Interaction.....	11
7.	DMCLIENTAPI INTERFACES.....	12
7.1	DMAPI.....	12
7.2	OMAAPIOBJECT .....	12
7.3	DMCLIENT.....	12
7.4	DMALERTOBJECT.....	16
7.5	DMOBJECT .....	17
7.6	DMUPDATELISTENER .....	20
7.7	DMUPDATETARGET .....	21
7.8	MOUPDATEINFO.....	22
7.9	OMAAPIEXCEPTION.....	23
7.10	MOINSTANCECB .....	23
7.11	MOINSTANCELISTCB .....	24
7.12	FLOW(S) (INFORMATIVE).....	26
8.	RELEASE INFORMATION .....	27
8.1	SUPPORTING FILE DOCUMENT LISTING.....	27
APPENDIX A. CHANGE HISTORY (INFORMATIVE).....		28
A.1	APPROVED VERSION HISTORY .....	28
A.2	DRAFT/CANDIDATE VERSION 1.0 HISTORY .....	28
APPENDIX B. USE CASES (INFORMATIVE) .....		29
B.1	CONFIGURATION OF INSTALLED SOFTWARE: INITIAL SETTINGS ARE ALREADY PRESENT .....	29
B.1.1	Short Description .....	29
B.1.2	Scenario .....	29
B.1.3	Market benefits .....	29
B.2	CONFIGURATION OF INSTALLED SOFTWARE: INITIAL SETTINGS ARE PROVIDED BY APPLICATION .....	29
B.2.1	Short Description .....	29
B.2.2	Scenario .....	29
B.2.3	Market benefits .....	30

**APPENDIX C. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....31**  
    **C.1 ERDEF FOR DMCLIENTAPIFW - CLIENT REQUIREMENTS .....31**  
    **C.2 SCR FOR DM CLIENT .....32**  
**APPENDIX D. DMCLIENTAPI FULL WIDL .....33**

# 1. Scope

The document includes RD, AD, and TS for DMClientAPIfw 1.0. It defines the Application Programming Interfaces (API) as part of the OMA DM Client API framework enabler work item.

The APIs to be defined by DMClientAPIfw are specifically scoped to enable access to the services provided by OMA DM enablers as made available on the host device.

## 2. References

### 2.1 Normative References

- [DMPRO] "OMA Device Management Protocol", Version 1.2, Open Mobile Alliance™, OMA-TS-DM\_Protocol-V1\_2,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [OMA-API-Module] "OMA API Common Module (omaapi)", OMA-SUP-IDL-WRAPI-Common,  
[URL:http://www.openmobilealliance.org/](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](http://www.ietf.org/rfc/rfc2119.txt)
- [RFC4627] "The application/json Media Type for JavaScript Object Notation (JSON)", D. Crockford, July 2006,  
[URL:http://www.ietf.org/rfc/rfc4627.txt](http://www.ietf.org/rfc/rfc4627.txt)
- [SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR\_Rules\_and\_Procedures,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)

### 2.2 Informative References

- [DMDICT] "OMA Device Management Dictionary", Draft Version 1.0, , Open Mobile Alliance™,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [OMADICT] "Dictionary for OMA Specifications", Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2\_8,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [WebIDL] "Web IDL", Worldwide Web Consortium (W3C),  
[URL:http://www.w3.org/TR/WebIDL/](http://www.w3.org/TR/WebIDL/)

## 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

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

**Local Application**                      A Local Application on the device is an agent (e.g. installed software) that operates according to an MO that is accessible by the DM Client and managed by the DM Server.

### 3.3 Abbreviations

Kindly consult [DMDICT] and [OMADICT] for all abbreviations used in this document.

## 4. Introduction

The OMA DM protocol is used for remote management of devices. The DM protocol is a direct communication between a DM Server and a DM Client. It does not specify the communication between device's Local Applications and a DM Client. This document provides a Device Management Client API framework to allow device's Local Applications and the DM Client to communicate.



## 5. Requirements (Normative)

### 5.1 High-Level Functional Requirements

Label	Description	Release
DMClientAPIfw1_0-HLF-1	The DM Client SHALL allow device's Local Applications to interact with registered Device Management Objects	1.0
DMClientAPIfw1_0-HLF-2	The DMClientAPIfw enabler SHALL provide a mechanism for device applications to request notification of changes to a registered MO	1.0
DMClientAPIfw1_0-HLF-3	The DMClientAPIfw enabler SHALL provide a mechanism to notify a device application of changes to a registered MO.	1.0
DMClientAPIfw1_0-HLF-4	The DMClientAPIfw enabler SHALL support API's to access OMA DM enabler.	1.0
DMClientAPIfw1_0-HLF-5	The DMClientAPIfw API's support SHALL be discoverable by device's Local Applications.	1.0
DMClientAPIfw1_0-HLF-6	The DMClientAPIfw API SHALL be defined using consistent API design patterns, e.g. error handling, namespaces, and interface structure.	1.0
DMClientAPIfw1_0-HLF-7	The DMClientAPIfw SHALL support scripting language callable API	1.0
DMClientAPIfw1_0-HLF-8	The DMClientAPIfw API SHALL be defined in Web IDL [WebIDL]	1.0
DMClientAPIfw1_0-HLF-9	The DMClientAPIfw SHALL be platform independent.	1.0
DMClientAPIfw1_0-HLF-10	The DMClientAPIfw API SHALL be compatible with OMA DM enabler version 1.2 and any later backward compatible versions.	1.0

### 5.2 DM-7 (Registration) API Requirements

Label	Description	Release
DM7-API-001	The DMClientAPIfw enabler SHALL have API to register/deregister the standardized MO implementation.	1.0
DM7-API-002	The DMClientAPIfw enabler SHALL have API to register/deregister the non-standardized MO implementation.	1.0
DM7-API-003	The DMClientAPIfw enabler SHALL have API to register/deregister the DDF of the non-standardized MO implementation.	1.0
DM7-API-004	The DMClientAPIfw enabler SHALL have API to register/deregister the MO as the set of cached node values.	1.0
DM7-API-005	The DMClientAPIfw enabler SHALL have API to update the cached node value of the MO.	1.0

### 5.3 DM-8 (Notification) API Requirements

Label	Description	Release
DM8-API-001	The DMClientAPIfw enabler SHALL have API to add/remove callback function for the Get operation on the registered MO.	1.0
DM8-API-002	The DMClientAPIfw enabler SHALL have API to add/remove callback function for the Replace operation on the registered MO.	1.0
DM8-API-003	The DMClientAPIfw enabler SHALL have API to add/remove callback function for the Add operation on the registered MO.	1.0
DM8-API-004	The DMClientAPIfw enabler SHALL have API to add/remove callback function for the Exec operation on the registered MO.	1.0

DM8-API-005	The DMClientAPIfw enabler SHALL have API to add/remove callback function for the Delete operation on the registered MO.	1.0
DM8-API-006	The DMClientAPIfw enabler SHALL have API to add/remove callback function for getting/setting value of multiple nodes of the registered MO.	1.0
DM8-API-007	The DMClientAPIfw enabler SHALL have API to add/remove callback function to be invoked when a change occurs in a specific MO instance. The callback function SHALL convey the set of modified data values.	1.0

## 5.4 DM-9 (Interaction) API Requirements

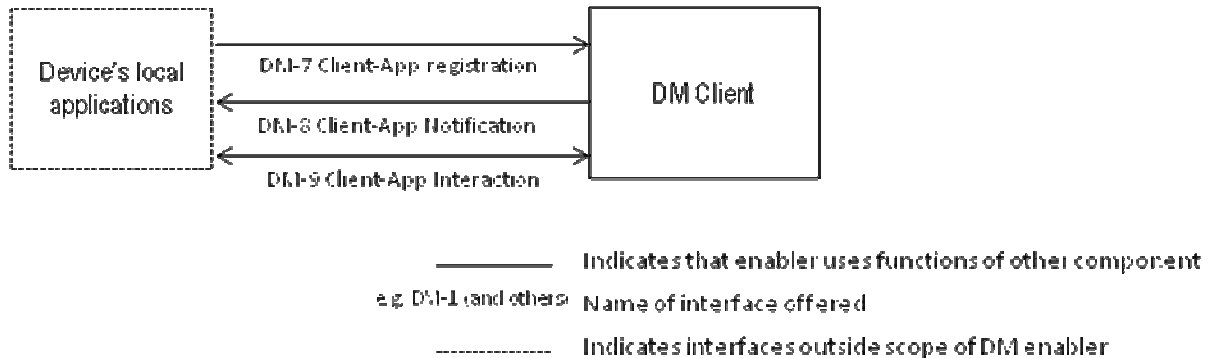
Label	Description	Release
DM9-API-001	The DMClientAPIfw enabler SHALL have API to start Client Initiated Session. Server Identifier SHALL be supported as a parameter.	1.0
DM9-API-002	The DMClientAPIfw enabler SHALL have API to abort ongoing DM session.	1.0
DM9-API-003	The DMClientAPIfw enabler SHALL have API to send Generic Alert.	1.0
DM9-API-004	The DMClientAPIfw enabler SHALL have API to get status of DM Client.	1.0
DM9-API-005	The DMClientAPIfw enabler SHALL have API to get DM Client version.	1.0
DM9-API-006	The DMClientAPIfw enabler SHALL have API to instantiate a MO in the Management Tree with provided initial node values.	1.0
DM9-API-007	The DMClientAPIfw enabler SHALL have API to read the node value of the device's Management Tree from the Local Application.	1.0
DM9-API-008	The DMClientAPIfw enabler SHALL have API to configure callback API invocable by DM Client to send modified MO data when change occurs in specific MO instance.	1.0

## 6. Architectural Model

### 6.1 Dependencies

DMClientAPIfw 1.0 has the same dependencies as the Device Management v1.2 enabler.

### 6.2 Architectural Diagram



### 6.3 Functional Components and Interfaces/reference points definition

#### 6.3.1 Protocol Endpoint

##### 6.3.1.1 DM Client

The DM Client is the abstract software component that conforms to the requirements for DM Clients specified in the OMA Device Management Enabler.

#### 6.3.2 Interfaces

##### 6.3.2.1 DM-7 Client-App Registration

This provides an interface over which the device's Local Applications may send Management Object registration or unregistration commands to the DM Client. The device's Local Application may issue a Management Object retrieval request.

##### 6.3.2.2 DM-8 Client-App Notification

This provides an interface over which the DM Client may send Management Object update notification to the registered device's Local Application.

##### 6.3.2.3 DM-9 Client-App Interaction

This provides an interface over which the device's Local Applications may send Management Object manipulation and retrieval commands.

## 7. DMClientAPI Interfaces

### 7.1 DMAPI

This interface represents the instance of DMAPI through which the functionality of the DMClient can be accessed.

**Table 1: DMAPI interface**

```
Web IDL Specification
[NoInterfaceObject] interface DMAPI {
    readonly attribute DMClient dmClient;
};
```

#### Attributes

*readonly DMClient dmClient*

#### Methods

*n/a*

### 7.2 OmaapiObject

This interface represents the instance of OmaapiObject specified in [OMA-API-Module].

**Table 2: OmaapiObject interface**

```
Web IDL Specification
[NoInterfaceObject] interface OmaapiObject {
    readonly attribute Omaapi omaapi;
};
```

#### Parameters

*readonly Omaapi omaapi*

#### Methods

*n/a*

### 7.3 DMClient

This interface is the main entry point to DM API and its functions.

This interface SHALL be implemented by DM Client API framework.

**Table 3: DMClient interface**

```

Web IDL Specification

[NoInterfaceObject] interface DMClient {

    readonly attribute DOMString version;

    const unsigned short CONNECTING = 0;
    const unsigned short OPEN = 1;
    const unsigned short CLOSING = 2;
    const unsigned short CLOSED = 3;
    const unsigned short ABORTED = 4;

    readonly attribute unsigned short sessionStatus;

    PendingOperation startDMSession(in SuccessCallback successCB,in ErrorCallback errorCallback, in
DOMString serverId, in optional DMLAlertObject alert) raises (OmaAPIError);

    StringArray listURIByMOID(in DOMString moid) raises (OmaAPIError);

    PendingOperation getMOByURI(in DOMString uri, in MOInstanceCB successCB, in ErrorCallback
errorCB) raises (OmaAPIError);

    DOMString createMO(in String moid,in MOData data) raises (OmaAPIError);

};
    
```

**Attributes**

*readonly DOMString version*

Attribute containing protocol version supported by this DMClient interface.

*readonly unsigned short sessionStatus*

Attribute representing the session Status.

**Methods**

*startDMSession*

**Table 4: startDMSession method**

```

Web IDL Specification

PendingOperation startDMSession(in SuccessCallback successCB,in ErrorCallback errorCallback, in DOMString serverId, in
optional DMLAlertObject alert) raises (OmaAPIError);
    
```

This method allows DM Client to start a DM Session.

It is possible to pass a DMLAlertObject object to the DM Client in order to insert a Generic Alert in package#1.

Returned PendingOperation object allows to cancel ongoing operation.

**Parameters**

<i>successCB</i>	<b>Optional:</b> no <b>Type:</b> SuccessCallback
------------------	---

	<p><b>Description:</b> Function to be invoked when DM Session ends successfully</p>
<i>errorCB</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> ErrorCallback</p> <p><b>Description:</b> Function to be invoked when DM Session fails</p>
<i>serverId</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DOMString</p> <p><b>Description:</b> The ServerId of the server that the DM client MUST connect to. This value can be an empty DOMString when only one ServerID is available on the Management Tree. If it does not match with any bootstrapped server, it MUST be rejected</p>
<i>alert</i>	<p><b>Optional:</b> yes</p> <p><b>Type:</b> DMAAlertObject</p> <p><b>Description:</b> If defined, this object contains values which MUST be used to populate generic alert in package#1</p>

**Returned Value**

PendingOperation: PendingOperation object allowing to cancel ongoing operation

*listURIByMOID*

**Table 5: listURIByMOID method**

```
Web IDL Specification
StringArray listURIByMOID(in DOMString moid) raises (OmaAPIError);
```

This method allows retrieving the list of MO instances URIs for the given MOID.

Each returned URI SHALL specify the location of a MO instance in the DM Tree.

This method is synchronous.

**Parameters**

<i>moid</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DOMString</p> <p><b>Description:</b> The MO identifier used to retrieve the list of MO instance URIs: it MUST be a valid MO URN (e.g. to retrieve a CAB MO, the MOID SHOULD be "urn:oma:mo:oma-cab:1.0")</p>
-------------	---

**Returned Value**

StringArray: StringArray array of MO instance URIs

*getMOByURI*

**Table 6: getMOByURI method**

```
Web IDL Specification
PendingOperation getMOByURI(in DOMString uri, in MOInstanceCB successCB, in ErrorCallback errorCB) raises
(OmaAPIError);
```

This method allows retrieving the DMOBJECT (see sub clause 7.5) representing the MO instance identified by provided URI.

This method is asynchronous and therefore returns immediately.

The successCB and errorCB callback functions are used to handle the results.

**Parameters**

<i>uri</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DOMString</p> <p><b>Description:</b> The absolute URI indicating the location of the MO instance in the DM Tree</p>
<i>successCB</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> MOInstanceCB</p> <p><b>Description:</b> Function to be invoked when the asynchronous operation completes successfully; the DMOBJECT representing the MO instance is returned as parameter</p>
<i>errorCB</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> ErrorCallback</p> <p><b>Description:</b> Function to be invoked when the asynchronous operation fails</p>

**Returned Value**

PendingOperation: PendingOperation object allowing to cancel ongoing operation

*createMO*

**Table 7: createMO method**

```
Web IDL Specification
```

```
DOMString createMO(in String moid,in MOData data) raises (OmaAPIError);
```

This method allows DM Client to create the DMOBJECT (see sub clause 7.5) representing the MO instance identified by the provided URI.

**Parameters**

<i>moid</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> String</p> <p><b>Description:</b> The MOID of the provided URI identifies the Management Object to be created by the DM Client</p>
<i>data</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> MOData</p> <p><b>Description:</b> MO structured data</p>

**Returned Value**

DOMString: DOMString created MO root URI

## 7.4 DMAAlertObject

This interface represents the instance of DM Generic Alert Object, to be used to define a generic alert inside package#1 (as defined in [DMPRO]) when a DM Session start is triggered.

**Table 8: DMAAlertObject interface**

**Web IDL Specification**

```
[NoInterfaceObject] interface DMAAlertObject {
    attribute DOMString metaType;
    attribute DOMString alertType;
    attribute DOMString mark;
    attribute DOMString data;
};
```

**Attributes**

*DOMString metaType*

Indicates the type of the content information of Generic Alert object. Its value MUST be compliant to [DMPRO].



***DOMString alertType***

Indicates the media type of the content information within the Generic Alert.

***DOMString mark***

Indicates the value of mark element of Generic Alert. Its value MUST be compliant to [DMPRO].

***DOMString data***

Represents the container for data to be conveyed in Generic Alert.

**Methods**

*n/a*

## 7.5 DMOBJECT

This interface represents the instance of a DM MO identified by the MOID; the methods defined by this interface can be used to interact with an instance of this MO, e.g. to read the values of nodes.

This interface implements the MOUpdateTarget interface (see sub clause 7.7).

**Table 9: DMOBJECT interface**

Web IDL Specification

```
[NoInterfaceObject] interface DMOBJECT: MOUpdateTarget {
    attribute DOMString moid;
    attribute DOMString uri;
    MOData getNodeValue(in DOMString nodeURI) raises (OmaAPIError);
    void setNodeValues(in StringArray nodeURIs, in StringArray values) raises (OmaAPIError);
    void setNodeValues(in DOMString nodeURI, in MOData subtree) raises (OmaAPIError);
    void deleteNode(in DOMString nodeURI) raises (OmaAPIError);
};
```

**Attributes*****DOMString moid***

Management Object MOID URN.

***DOMString uri***

Management Object instance URI.

**Methods*****getNodeValue***

**Table 10: getNodeValue method**

```
Web IDL Specification
MOData getNodeValue(in DOMString nodeURI) raises (OmaAPIError);
```

Returns the value of the node, whose location, within the scope of the DMOBJECT, is specified by the nodeURI.

**Parameters**

<i>nodeURI</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DOMString</p> <p><b>Description:</b> The relative URI of the node within the DMOBJECT</p>
----------------	--

**Returned Value**

MOData: MOData

*setNodeValues*

**Table 11: setNodeValues method**

```
Web IDL Specification
void setNodeValues(in StringArray nodeURIs, in StringArray values) raises (OmaAPIError);
```

Sets values for the specific nodes identified by nodeURIs.

If node already exists, this method overwrites actual value; if node does not exist, a new node is created; if value is null, nodeURI identifies an interior node.

**Parameters**

<i>nodeURIs</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> StringArray</p> <p><b>Description:</b> A String Array containing relative URIs of the nodes within the DMOBJECT</p>
<i>values</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> StringArray</p> <p><b>Description:</b> A String Array containing node values. If an array element is null, related node is interior</p>

**Returned Value**

void

*setNodeValues*

**Table 12: setNodeValues method**

```
Web IDL Specification
void setNodeValues(in DOMString nodeURI,in MOData subtree) raises (OmaAPIError);
```

Sets subtree for the node identified by nodeURI.

If node already exists, this method overwrites actual subtree structure; if node does not exist, a new node is created; if value is null, nodeURI identifies an interior node.

**Parameters**

<i>nodeURI</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DOMString</p> <p><b>Description:</b> The relative URI of the node within the DMOBJECT</p>
<i>subtree</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> MOData</p> <p><b>Description:</b> Node subtree represented by MOData. If null, node is interior</p>

**Returned Value**

void

*deleteNode*

**Table 13: deleteNode method**

```
Web IDL Specification
void deleteNode(in DOMString nodeURI) raises (OmaAPIError);
```

Deletes node identified by nodeURI. If the target node is the interior node, then the subtree of this node will be deleted.

**Parameters**

<i>nodeURI</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DOMString</p> <p><b>Description:</b> The relative URI of the node within the</p>
----------------	---

	DMObject
--	----------

**Returned Value**

void

## 7.6 DMUpdateListener

This interface SHALL be implemented by an application in order to handle the update notifications.

The handleDMUpdate method is invoked by underlying implementation in order to notify an update.

**Table 14: DMUpdateListener interface**

```

Web IDL Specification
[NoInterfaceObject] interface DMUpdateListener {
    void handleDMUpdate(in MOUpdateInfo dmUpdate);
};
    
```

**Attributes**

*n/a*

**Methods**

*handleDMUpdate*

**Table 15: handleDMUpdate method**

```

Web IDL Specification
void handleDMUpdate(in MOUpdateInfo dmUpdate);
    
```

This method is invoked when the DMUpdateInfo occurs on the DMUpdateTarget for which the DMUpdateListener was registered.

**Parameters**

<i>dmUpdate</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> MOUpdateInfo</p> <p><b>Description:</b> The MOUpdateInfo object representing the DM update for which this method was called</p>
-----------------	--

**Returned Value**

void

## 7.7 DMUpdateTarget

This interface allows registering and deregistering DMUpdateListeners based on a specific MOUpdateInfo operation. This interface SHALL be implemented by all DMOjects which could be target of DMUpdate operations.

**Table 16: DMUpdateTarget interface**

```

Web IDL Specification
[NoInterfaceObject] interface DMUpdateTarget {
    void addDMUpdateListener (in unsigned short type, in DMUpdateListener listener) raises
(OmaAPIError);
    void removeDMUpdateListener (in unsigned short type, in DMUpdateListener listener) raises
(OmaAPIError);
};
    
```

### Attributes

n/a

### Methods

*addDMUpdateListener*

**Table 17: addDMUpdateListener method**

```

Web IDL Specification
void addDMUpdateListener (in unsigned short type, in DMUpdateListener listener) raises (OmaAPIError);
    
```

This method allows registering a DMUpdateListener for the specified DMUpdate operation.

### Parameters

<i>type</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> unsigned short</p> <p><b>Description:</b> The DMUpdate operation for which the application asks the registration</p>
<i>listener</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DMUpdateListener</p> <p><b>Description:</b> Object implementing the DMUpdateListener interface to be registered</p>

### Returned Value

void

*removeDMUpdateListener*

**Table 18: removeDMUpdateListener method**

```
Web IDL Specification
void removedMUpdateListener (in unsigned short type, in DMUpdateListener listener) raises (OmaAPIError);
```

This method allows deregistering a DMUpdateListener for the specified DMUpdate operation.

**Parameters**

<i>type</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> unsigned short</p> <p><b>Description:</b> The DMUpdate operation for which the application asks the unregistration</p>
<i>listener</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DMUpdateListener</p> <p><b>Description:</b> Object implementing the DMUpdateListener interface to be deregistered</p>

**Returned Value**

void

## 7.8 MOUpdateInfo

This interface contains the details regarding an update to Management Tree which are notified by the DM Client to registered applications.

**Table 19: MOUpdateInfo interface**

```
Web IDL Specification
[NoInterfaceObject] interface MOUpdateInfo {
    const unsigned short OP_CREATE = 0;
    const unsigned short OP_REPLACE = 1;
    const unsigned short OP_UPDATE = 2;
    const unsigned short OP_DELETE = 3;

    readonly attribute unsigned short type;
    readonly attribute unsigned long timestamp;
```

```
};    readonly attribute MOUpdateTarget MOUpdateTarget;
```

## Attributes

### *readonly unsigned short type*

Indicates the type of MOUpdate.

### *readonly unsigned long timestamp*

Indicates the time and date of update, expressed as a UTC based [ISO8601] basic format.

### *readonly MOUpdateTarget MOUpdateTarget*

Indicates the MOUpdateTarget associated with the update.

## Methods

*n/a*

## 7.9 OmaAPIException

This interface describes the object which is returned when an exception is raised by current operation.

Exception codes list extends OmaAPIError codes defined in [OMA-API-Module].

**Table 20: OmaAPIException interface**

```
Web IDL Specification
[Supplemental] interface OmaAPIException {
    const unsigned short    UNSUPPORTED_OPERATION    = 1001;
    const unsigned short    CLIENT_NOT_READY        = 1002;
    const unsigned short    NOT_ALLOWED_OPERATION    = 1003;
};
```

## Attributes

*n/a*

## Methods

*n/a*

## 7.10 MOInstanceCB

This interface SHALL be implemented by function used as success callback for DMClientInterface.getMOByURI method.

The onSuccess method is invoked on getMOByURI normal exit.

**Table 21: MOInstanceCB interface**

```

Web IDL Specification
[Callback=FunctionOnly,NoInterfaceObject] interface MOInstanceCB {
    void onSuccess (in DMOBJECT dmObject);
};
    
```

**Attributes**

*n/a*

**Methods**

*onSuccess*

**Table 22: onSuccess method**

```

Web IDL Specification
void onSuccess (in DMOBJECT dmObject);
    
```

This method is invoked on normal exit by getMOByURI.

**Parameters**

<i>dmObject</i>	<p><b>Optional:</b> no</p> <p><b>Type:</b> DMOBJECT</p> <p><b>Description:</b> DMOBJECT representing MO instance.</p>
-----------------	---

**Returned Value**

void

## 7.11 MOInstanceListCB

This interface SHALL be implemented by function used as success callback for DMClientInterface.getMOByMOID method.

The onSuccess method is invoked on getMOByMOID normal exit.

**Table 23: MOInstanceListCB interface**

```

Web IDL Specification
[Callback=FunctionOnly,NoInterfaceObject] interface MOInstanceListCB {
    void onSuccess (in DMOBJECTArray dmObjectArray);
};
    
```



## Attributes

*n/a*

## Methods

*onSuccess*

**Table 24: onSuccess method**

### Web IDL Specification

```
void onSuccess (in DMOBJECTARRAY dmObjectArray);
```

This method is invoked on normal exit by getMOByMOID.

### Parameters

<i>dmObjectArray</i>	<b>Optional:</b> no <b>Type:</b> DMOBJECTARRAY <b>Description:</b> Array of DMOBJECTS
----------------------	---

### Returned Value

void

## 7.12 Flow(s)

(Informative)

The following flow describes the interactions between the Local Application and the DMClient API, particularly the following operations:

- 1) Retrieving a DMOBJect representing an MO instance from the DM tree maintained by the DM Client
- 2) Local Application registration of a “listener/handler” for notification of updates (e.g. create, replace, update, delete) to the retrieved MO instance
- 3) Notification of updates to the registered MO instance to the Local Application
- 4) Execution of logic (e.g. Local Application’s interaction with the DMOBJect)

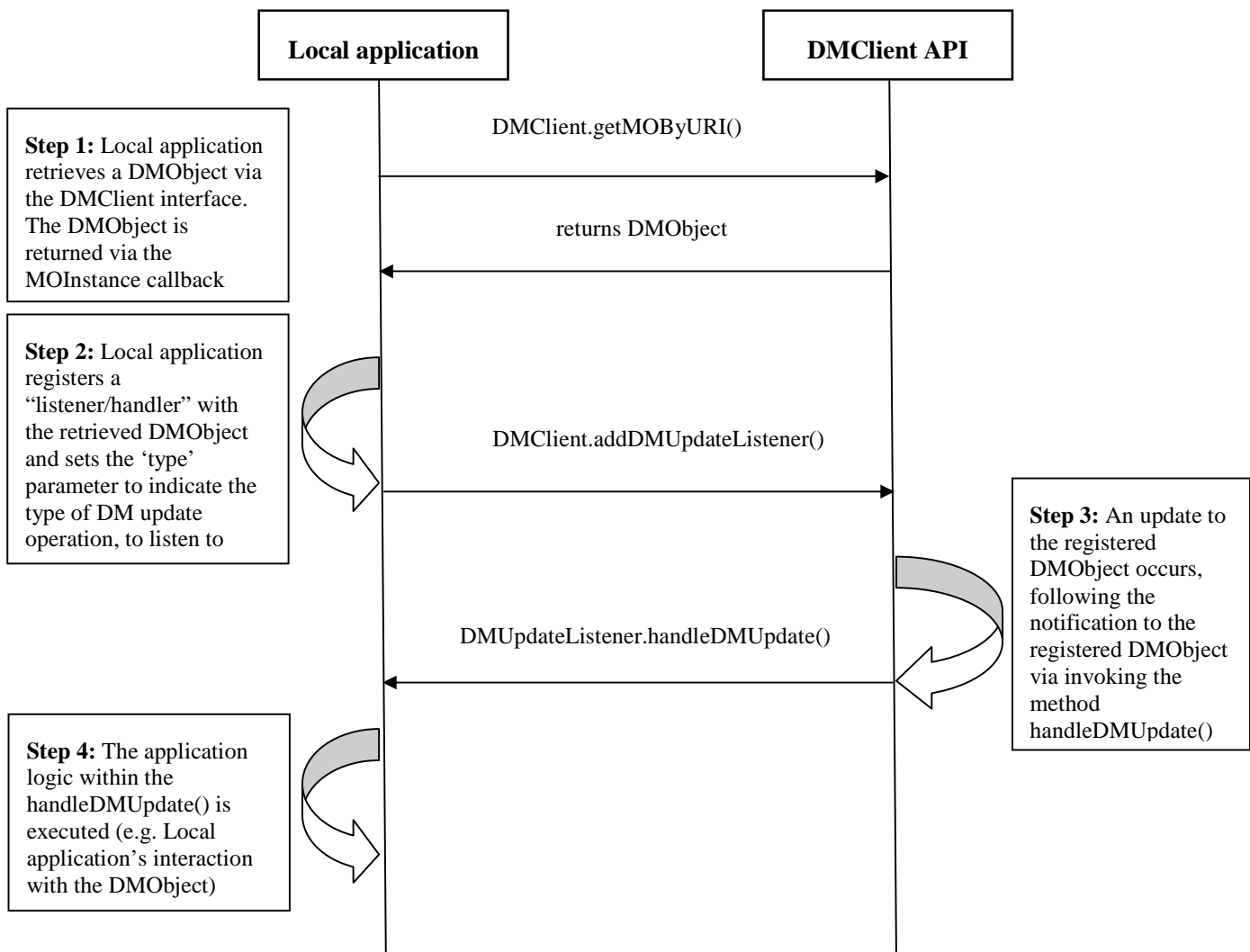


Figure 1: Flow between a device’s Local Application and the DM Client

## 8. Release Information

### 8.1 Supporting File Document Listing

Doc Ref	Permanent Document Reference	Description
<b>Supporting Files</b>		
[DMClientAPIFW_WIDL]	OMA-SUP-WIDL_DMClientAPIfw-V1_0-20110829-D	This WIDL describes the DMClientAPIfw API Working file in WIDL directory: file: dmclientapifw-v1_0.widl path: <a href="http://www.openmobilealliance.org/tech/profiles/">http://www.openmobilealliance.org/tech/profiles/</a>

**Table 25: Listing of Supporting Documents in DMClientAPIfw Release**

## Appendix A. Change History (Informative)

### A.1 Approved Version History

Reference	Date	Description
N/A	N/A	No prior version

### A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description	
Draft Versions OMA-ER-DMClientAPIFw-V1_0	30 Aug 2010	All	First draft baseline as agreed in “OMA-DM-DMClientAPIFw-2010-0001R01-INP_DMClientAPIFw_1_0_baseline”	
	08 Sep 2010	5.1	Modified according to “OMA-DM-DMClientAPIFw-2010-0002R01-INP_Requirements”	
	15 Oct 2010	1/ 2.2 / 4 / 5.1 / 6.1 / 6.2 / 6.3	Modified according to “OMA-DM-DMClientAPIFw-2010-0003-CR_Update” and “OMA-DM-DMClientAPIFw-2010-0004-CR_Additional_Reqs”	
	16 Mar 2011	5.1	Applied: OMA-DM-DMClientAPIFw-2011-0003R01-CR_Req_modifications MA-DM-DMClientAPIFw-2011-0004R04-CR_REQ_Versions OMA-DM-DMClientAPIFw-2011-0005R01-CR_API_Requirements	
	08 Jun 2011	7	Applied: OMA-DM-DMClientAPIFw-2011-0001R04-CR_first_APIS	
	13 July 2011	all	Applied: OMA-DM-DMClientAPIFw-2011-0008R02-CR_New_API OMA-DM-DMClientAPIFw-2011-0006R05-CR_API_UseCase	
	19 July 2011	all	Applied: OMA-DM-DMClientAPIFw-2011-0009R02- CR_API_Description_Cleanup	
	17 Aug. 2011	8/ 5.5 / 5.4/ 7.2	Applied: OMA-DM-DMClientAPIFw-2011-0015-CR_Addition_Support_File OMA-DM-DMClientAPIFw-2011-0012R01-CR_ACL_Requirement OMA-DM-DMClientAPIFw-2011-0011R01- CR_DM_Session_Requirement	
	18 Aug. 2011	all	Applied: OMA-DM-DMClientAPIFw-2011-0014-CR_Editor_Note_Cleanup	
	07 Sep. 2011	all	Applied: OMA-DM-DMClientAPIFw-2011-0016-CR_Remove_useless_sections OMA-DM-DMClientAPIFw-2011-0017R02- CR_Closure_Review_Editorials OMA-DM-DMClientAPIFw-2011-0024-CR_WRAPAPI_Style_Format OMA-DM-DMClientAPIFw-2011-0018- CR_Populating_the_SCR_Tables OMA-DM-DMClientAPIFw-2011-0019R02- CR_DM_version_consistency OMA-DM-DMClientAPIFw-2011-0021R01- CR_sessionId_startDMSessionAPI OMA-DM-DMClientAPIFw-2011-0022R01- CR_ACL_Requirement_part_2 OMA-DM-DMClientAPIFw-2011-0025-CR_Revised_Flow OMA-DM-DMClientAPIFw-2011-0026- CR_Sessionid_in_DM_Session_Start	
	08 Sep 2011	all	General editorial clean-up by DSO prior to initiation of CONR Update of copyright year in footers Sorting of listings in alphabetical order	
	Candidate Version OMA-ER-DMClientAPIFw-V1_0	04 Oct 2011	n/a	Status changed to Candidate by TP Ref #OMA-TP-2011-0344- INP_DMClientAPIFw_V1_0_ERP_for_Candidate_approval

## Appendix B. Use Cases (Informative)

### B.1 Configuration of Installed Software: initial settings are already present

The application is configured with settings already present in the Management Tree, interacting with the DM Client through the DM Client API Framework.

#### B.1.1 Short Description

A user installs an application on her mobile device, instead of having to manually enter configuration settings to make the application function properly, the application itself would automatically register with the OMA DM Client to access the application specific Management Object. If authorized, the application would then retrieve the parameters from the MO, to configure itself.

#### B.1.2 Scenario

1. The User installs an application on her mobile device.
2. The application interacts with the DM Client to get the corresponding MO instance identified by MOID.
3. The application obtained the MO instance and retrieves the parameters of the MO instance.
4. The application registers itself to be notified in case of MO updates.
5. If MO content is modified, i.e. by DM Server after a DM Session, DM Client notifies the application with new settings.

#### B.1.3 Market benefits

The User does not have to configure the application manually.

### B.2 Configuration of Installed Software: initial settings are provided by application

The application provides the initial set of parameters which can be updated by the DM Server later.

#### B.2.1 Short Description

A user installs an application on her mobile device, instead of having to manually enter configuration settings to make the application function properly, the application would create, if authorized, a MO in the Management Tree and asks the DMClient to perform a DM Session with the DM Server in order to retrieve latest settings. Since the application already registered with the OMA DM Client to be notified of any update in the MO parameters, the new settings will be sent to the application after the DM session is finished.

#### B.2.2 Scenario

1. The User installs an application on her mobile device.
2. The application interacts with the DM Client to get the MO instance identified by MOID.
3. The application does not find any instances of the requested MO.
4. The application asks to the DM Client to create the new MO with initial values provided by the application. The DM Client will return the URI of the created MO instance.
5. The application registers itself to be notified in case of MO updates.
6. The application triggers the DM Client to start a DM session.

7. The DM Client performs a DM session with the DM Server, which handles the new MO update if necessary.
8. If updates have been applied, the DM Client notifies it to the application with new settings.

### **B.2.3 Market benefits**

The User does not have to configure the application manually.

## Appendix C. Static Conformance Requirements (Normative)

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

### C.1 ERDEF for DMClientAPIfw - Client Requirements

This section is normative.

Item	Feature / Application	Requirement
OMA-ERDEF-DMClientAPIfw-C-001-M	DM Client	DMClientAPIfw1_0-HLF-2
OMA-ERDEF-DMClientAPIfw-C-002-M	DM Client	DMClientAPIfw1_0-HLF-4
OMA-ERDEF-DMClientAPIfw-C-003-M	DM Client	DMClientAPIfw1_0-HLF-5
OMA-ERDEF-DMClientAPIfw-C-004-M	DM Client	DMClientAPIfw1_0-HLF-6
OMA-ERDEF-DMClientAPIfw-C-005-M	DM Client	DMClientAPIfw1_0-HLF-7
OMA-ERDEF-DMClientAPIfw-C-006-M	DM Client	DMClientAPIfw1_0-HLF-8
OMA-ERDEF-DMClientAPIfw-C-007-M	DM Client	DMClientAPIfw1_0-HLF-9
OMA-ERDEF-DMClientAPIfw-C-008-M	DM Client	DMClientAPIfw1_0-HLF-10
<b>DM-7 Client-App Registration</b>		
OMA-ERDEF-DMClientAPIfw-C-010-M	DM Client	DMClientAPIfw1_0-HLF-1
OMA-ERDEF-DMClientAPIfw-C-011-M	DM Client	DM7-API-001
OMA-ERDEF-DMClientAPIfw-C-012-M	DM Client	DM7-API-002
OMA-ERDEF-DMClientAPIfw-C-013-M	DM Client	DM7-API-003
OMA-ERDEF-DMClientAPIfw-C-014-M	DM Client	DM7-API-004
OMA-ERDEF-DMClientAPIfw-C-015-M	DM Client	DM7-API-005
<b>DM-8 Client-App Notification</b>		
OMA-ERDEF-DMClientAPIfw-C-016-M	DM Client	DMClientAPIfw1_0-HLF-3
OMA-ERDEF-DMClientAPIfw-C-017-M	DM Client	DM8-API-001
OMA-ERDEF-DMClientAPIfw-C-018-M	DM Client	DM8-API-002
OMA-ERDEF-DMClientAPIfw-C-019-M	DM Client	DM8-API-003
OMA-ERDEF-DMClientAPIfw-C-020-M	DM Client	DM8-API-004
OMA-ERDEF-DMClientAPIfw-C-021-M	DM Client	DM8-API-005
OMA-ERDEF-DMClientAPIfw-C-022-M	DM Client	DM8-API-006
OMA-ERDEF-DMClientAPIfw-C-023-M	DM Client	DM8-API-007
<b>DM-9 Client-App Interaction</b>		
OMA-ERDEF-DMClientAPIfw-C-024-M	DM Client	DM9-API-001
OMA-ERDEF-DMClientAPIfw-C-025-M	DM Client	DM9-API-002
OMA-ERDEF-DMClientAPIfw-C-026-M	DM Client	DM9-API-003
OMA-ERDEF-DMClientAPIfw-C-027-M	DM Client	DM9-API-004
OMA-ERDEF-DMClientAPIfw-C-028-M	DM Client	DM9-API-005
OMA-ERDEF-DMClientAPIfw-C-029-M	DM Client	DM9-API-006
OMA-ERDEF-DMClientAPIfw-C-030-M	DM Client	DM9-API-007
OMA-ERDEF-DMClientAPIfw-C-031-M	DM Client	DM9-API-008

Table 26: ERDEF for DMClientAPIfw Client-side Requirements

## C.2 SCR for DM Client

Item	Function	Reference	Requirement
DMClientAPIfw-C-001-M	This interface is the main entry point to DMAPI and its functions	Section 7.3	DMClientAPIfw-C-001-M
DMClientAPIfw-C-002-M	This interface allows registering and deregistering DMUpdateListeners based on a specific MOUpdateInfo operation	Section 7.6	DMClientAPIfw-C-002-M
DMClientAPIfw-C-003-M	This interface allows an application to handle the update notifications	Section 7.7	DMClientAPIfw-C-003-M
DMClientAPIfw-C-004-M	This interface is implemented by the function used as success callback for DMClientInterface.getByURI method	Section 7.10	DMClientAPIfw-C-004-M
DMClientAPIfw-C-005-M	This interface is implemented by the function used as success callback for DMClientInterface.getByMOID method	Section 7.11	DMClientAPIfw-C-005-M



## Appendix D. DMClientAPI Full WIDL

Table 27: DMClientAPI module

### Web IDL Specification

```

typedef sequence<DMObject> DMObjectArray;

typedef Object MOData;

[NoInterfaceObject] interface DMAPI {
    readonly attribute DMClient dmClient;
};
Window implements DMAPI;

[NoInterfaceObject] interface OmaapiObject {
    readonly attribute Omaapi omaapi;
};
Window implements OmaapiObject;

[NoInterfaceObject] interface DMClient {

    readonly attribute DOMString version;

    const unsigned short CONNECTING = 0;
    const unsigned short OPEN = 1;
    const unsigned short CLOSING = 2;
    const unsigned short CLOSED = 3;
    const unsigned short ABORTED = 4;

    readonly attribute unsigned short sessionStatus;

    PendingOperation startDMSession(in SuccessCallback successCB, in ErrorCallback errorCB, in
DOMString serverId, in optional DMAlertObject alert) raises (OmaAPIError);

    StringArray listURIByMOID(in DOMString moid) raises (OmaAPIError);

    PendingOperation getMOByURI(in DOMString uri, in MOInstanceCB successCB, in ErrorCallback
errorCB) raises (OmaAPIError);

    DOMString createMO(in String moid, in MOData data) raises (OmaAPIError);
};

[NoInterfaceObject] interface DMAlertObject {
    attribute DOMString metaType;

    attribute DOMString alertType;

    attribute DOMString mark;

    attribute DOMString data;
};

[NoInterfaceObject] interface DMObject: MOUpdateTarget {
    attribute DOMString moid;

    attribute DOMString uri;

    MOData getNodeValue(in DOMString nodeURI) raises (OmaAPIError);

    void setNodeValues(in StringArray nodeURIs, in StringArray values) raises (OmaAPIError);

    void setNodeValues(in DOMString nodeURI, in MOData subtree) raises (OmaAPIError);

    void deleteNode(in DOMString nodeURI) raises (OmaAPIError);
};

[NoInterfaceObject] interface DMUpdateListener {
    void handleDMUpdate(in MOUpdateInfo dmUpdate);
};

```

```
};

[NoInterfaceObject] interface DMUpdateTarget {
    void addDMUpdateListener (in unsigned short type, in DMUpdateListener listener) raises
(OmaAPIError);

    void removeDMUpdateListener (in unsigned short type, in DMUpdateListener listener) raises
(OmaAPIError);
};

[NoInterfaceObject] interface MOUpdateInfo {

    const unsigned short    OP_CREATE        = 0;
    const unsigned short    OP_REPLACE      = 1;
    const unsigned short    OP_UPDATE       = 2;
    const unsigned short    OP_DELETE       = 3;

    readonly attribute unsigned short type;

    readonly attribute unsigned long timestamp;

    readonly attribute MOUpdateTarget MOUpdateTarget;
};

[Supplemental] interface OmaAPIException {

    const unsigned short    UNSUPPORTED_OPERATION    = 1001;
    const unsigned short    CLIENT_NOT_READY        = 1002;
    const unsigned short    NOT_ALLOWED_OPERATION    = 1003;
};

[Callback=FunctionOnly, NoInterfaceObject] interface MOInstanceCB {
    void onSuccess (in DMObject dmObject);
};

[Callback=FunctionOnly, NoInterfaceObject] interface MOInstanceListCB {
    void onSuccess (in DMObjectArray dmObjectArray);
};
```