OMA Device Management Standardized Objects

Candidate Version 1.3 – 25 May 2010

Open Mobile Alliance

OMA-TS-DM_StdObj-V1_3-20100525-C

Contents

Figures

Tables

1. Scope

This document defines a set of management objects. Some of these are mandatory for all OMA DM compliant devices and others are optional. The objects are defined using the OMA DM Device Description Framework.

2. References

2.1 Normative References

[DevDetailDDF] "OMA Device Management Detail Information Management Object DDF, Version 1.3". Open Mobile Alliance™. OMA-SUP-MO_DM_DevDetail-V1_3.
URL:http://www.openmobilealliance.org

[DevInfoDDF] "OMA Device Management Information Management Object DDF, Version 1.3". Open Mobile Alliance™. OMA-SUP-MO_DM_DevInfo-V1_3.
URL:http://www.openmobilealliance.org

[DMAccDDF] "OMA Device Management Account Management Object DDF, Version 1.3". Open Mobile Alliance™. OMA-SUP-MO_DM_DMAcc-V1_3.
URL:http://www.openmobilealliance.org

[DMBOOT] "OMA Device Management Bootstrap, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_Bootstrap-V1_3.
URL:http://www.openmobilealliance.org

[DMDDFDTD] "OMA DM Device Description Framework DTD, Version 1.3". Open Mobile Alliance™.
URL:http://www.openmobilealliance.org/tech/DTD/dm_ddf-v1_3.dtd

[DMNOTI] "OMA Device Management Notification Initiated Session, Version 1.3". Open Mobile Alliance™.OMA-TS-DM_Notification-V1_3.
URL:http://www.openmobilealliance.org

[DMPRO] "OMA Device Management Protocol, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_Protocol-V1_3.
URL:http://www.openmobilealliance.org

[DMREPPRO] "OMA Device Management Representation Protocol, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_RepPro-V1_3,
URL:http://www.openmobilealliance.org

[DMSEC] "OMA Device Management Security, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_Security-V1_3.
URL:http://www.openmobilealliance.org

[DMTND] "OMA Device Management Tree and Description, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_TND-V1_3.
URL:http://www.openmobilealliance.org

[DMTNDS] "OMA Device Management Tree and Description Serialization, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_TNDS-V1_3.
URL:http://www.openmobilealliance.org

[ERELDCP] "Enabler Release Definition for OMA Client Provisioning Specifications, version 1.1". Open Mobile Alliance™. OMA-ERELD-ClientProvisioning-V1_1.
URL:http://www.openmobilealliance.org

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1,
URL:http://www.openmobilealliance.org/

[ISO639-2] "ISO 639-2 Language Codes",
URL:http://www.loc.gov/standards/iso639-2/langhome.html

[RFC1766] "Tags for the Identification of Languages". H. Alvestrand. March 1995.
URL:http://www.ietf.org/rfc/rfc1766.txt

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

[RFC2141] "URN Syntax", R. Moats, May 1997,
URL:http://www.ietf.org/rfc/rfc2141.txt

[RFC2617] "HTTP Authentication: Basic and Digest Access Authentication", J. Frnaks et. Al.

URL:http://www.ietf.org/rfc/rfc2617.txt

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt

[RFC2373] "Internet Protocol Version 6 Addressing Architecture". R. Hinden and S. Deering. July 1998. URL:http://www.ietf.org/rfc/rfc2373.txt

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax". T. Berners-Lee, R. Fielding, L. Masinter. August 1998. URL:http://www.ietf.org/rfc/rfc2396.txt

[RFC791] "Internet Protocol: Darpa Internet Protocol Program Specification". September 1981. URL:http://www.ietf.org/rfc/rfc791.txt

[w7] "OMA w7 Application Characteristic for DM Version 1.0". Open Mobile Alliance™. OMA-w7-Application-Characteristic-for-DM-V1_0. URL:http://www.openmobilealliance.org

2.2 Informative References

None.

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.

Any reference to components of the DTD's or XML snippets are specified in this `typeface`.

3.2 Definitions

See the DM Tree and Description [DMTND] document for definitions of terms related to the management tree.

3.3 Abbreviations

None.

4. Introduction

Other OMA DM specifications define the syntax and semantics of the OMA DM protocol. However, the usefulness of such a protocol would be limited if the managed entities in devices required different data formats and displayed different behaviors. To avoid this situation this specification defines a number of mandatory management objects for various uses in devices.

5. Standardized Objects

5.1 Management Objects

Management objects are logical collections of related nodes that enable the targeting of management operations, using OMA DM protocol commands. Each node in a management object can be as small as an integer or large and complex like a background picture or screen saver. The OMA DM protocol is agnostic about the contents, or values, of the management objects and treats the node values as opaque data

5.1.1 Definition and description of management objects

OMA DM management objects are defined using the OMA DM Device Description Framework [DMTND], or DDF. The use of this description framework produces detailed information about the device in question. However, due to the high level of detail in these descriptions, they are sometimes hard for humans to digest and it can be a time consuming task to get an overview of a particular objects structure.

In order to make it easier to quickly get an overview of how a management object is organized and its intended use, a simplified graphical notation in the shape of a block diagram is used in this document. Even though the notation is graphical, it still uses some printable characters, e.g. to denote the number of occurrences of a node. These are mainly borrowed from the syntax of DTDs for XML. The characters and their meaning are defined in the following table.

Character	Meaning
+	one or many occurrences
*	zero or more occurrences
?	zero or one occurrences

If none of these characters is used the default occurrence is exactly once.

There is one more feature of the DDF that needs to have a corresponding graphical notation, the un-named block. These are blocks that act as placeholders in the description and are instantiated with information when the nodes are used at run-time. Un-named blocks in the description are represented by a lower case character in italics, e.g. *x*.

Each block in the graphical notation corresponds to a described node, and the text is the name of the node. If a block contains an *x*, it means that the name is not known in the description and that it will be assigned at run-time. The names of all ancestral nodes are used to construct the URI for each node in the management object. It is not possible to see the actual parameters, or data, stored in the nodes by looking at the graphical notation of a management object.

The following is an example of what a management object can look like when it is expressed using the graphical notation. This particular object is the OMA DM Device Information management object.

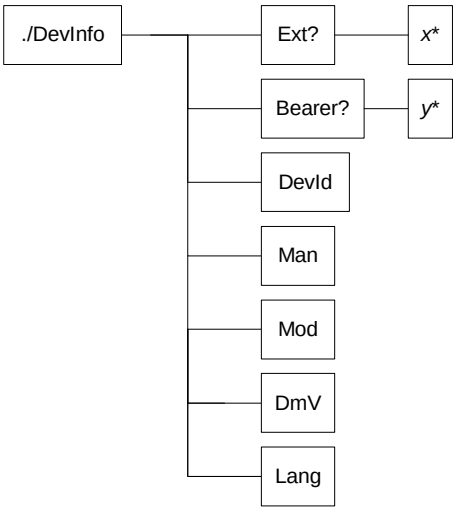


Figure 1: Example of a management object pictured using the graphical notation

Naturally, this graphical overview does not show all the details of the full description, but it provides a good map of the description so that it is easier to find the individual nodes. Although the figure only provides an elevated view of the description, there are still some things worth noticing. All the blocks with names in place occur exactly once, except Ext and Bearer that are optional and may not be present at all. One of the named nodes, DevInfo, has child nodes; it is an interior node. With the exception of Ext and Bearer, none of the other named nodes can have any children of their own; they are leaf nodes. The un-named leaf nodes are marked with *. This means that although the description only contains one node description at this position in the tree, there can be any number of instantiated nodes at run-time, including none. The only limit is that the node names MUST be unique and memory MUST be available to store the nodes.

The next figure shows an example of what the device information management object could look like at run-time.

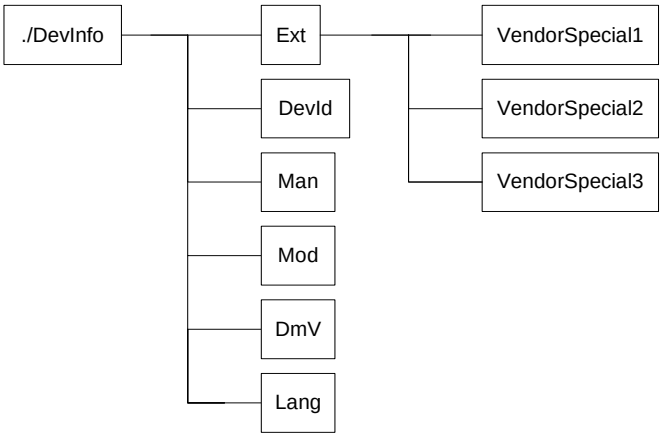


Figure 2: Example of an instantiated ./DevInfo object

The difference between this and the previous figure is that now the un-named blocks have been instantiated. It is also shown that the * character means that a node can occur zero or more times. Note that none of the stored data in the leaf nodes is shown in the figure, what are visible are only the node names.

5.1.2 DDF compliance

The management object descriptions in this document are normative. However, the descriptions also contain are a number of informative aspects that could be included to enhance readability or serve as examples. Other informative aspects are, for instance, the `ZeroOrMore` and `OneOreMore` elements, where implementations MAY introduce restrictions. All these exceptions are listed here:

- All XML comments, e.g. "`<!-- some text -->`", are informative.
- The descriptions do not contain an `RTProperties` element, or any of its child elements, but a description of an actual implementation of this object MAY include these.
- If a default value for a leaf node is specified in a description, by the `DefaultValue` element, an implementation MUST supply its own appropriate value for this element. If the `DefaultValue` element is present in the description of a node, it MUST be present in the implementation, but MAY have a different value.
- The value of all `Man, Mod, Description` and `DFTitle` elements are informative and included only as examples.
- Below the interior nodes `Ext` and `Bearer`, an implementation MAY add further nodes at will.
- The contents of the `AccessType` element MAY be extended by an implementation.
- If the any of the following `AccessType` values are specified, they MUST NOT be removed in an implementation: `Copy, Delete, Exec, Get,` and `Replace`.
- If the `AccessType` value `Add` is specified it MAY be removed in an implementation if the implementation only supports a fixed number of child nodes.
- An implementation MAY replace the `ZeroOrMore` or `OneOreMore` elements with `ZeroOrN` or `OneOrN` respectively. An appropriate value for `N` MUST also be given with the …`OrN` elements.

5.2 Management objects standardized by other organizations

OMA DM has been designed so that existing management objects can be managed. These existing management objects have typically already been standardized by other standards organizations.

5.3 The OMA DM management objects

Clients implementing OMA DM MUST support the OMA DM Account management object, DevInfo management object and the DevDetail management object, and Inbox management object. OMA DM servers MUST support all three management objects as well.

Management Object	Client Support	Server Support	Description
DMAcc	MUST	MUST	Settings for the DM client in a managed device.
DevInfo	MUST	MUST	Device information for the OMA DM server. Sent from the client to the server.

DevDetail	MUST	MUST	General device information that benefits from standardization.
Inbox	MUST	MUST	Reserved URI where the device SHOULD use the management object identifier to identify the absolute URI.

The difference between DevInfo and DevDetail is that the DevInfo parameters are needed by the management server for problem free operation of the OMA DM protocol. The DevInfo object is sent from client to server in the beginning of every session.

DevDetail contains other device specific parameters that benefits from being standardized and mandatory. The only difference is that these parameters are not sent from client to server automatically. Instead, these parameters are managed by servers as any other parameters and can be manipulated using OMA DM commands.

5.3.1 The DM Account management object

The management object is used to manage settings for OMA DM protocol.

Management object identifier: urn:oma:mo:oma-dm-dmacc:1.1

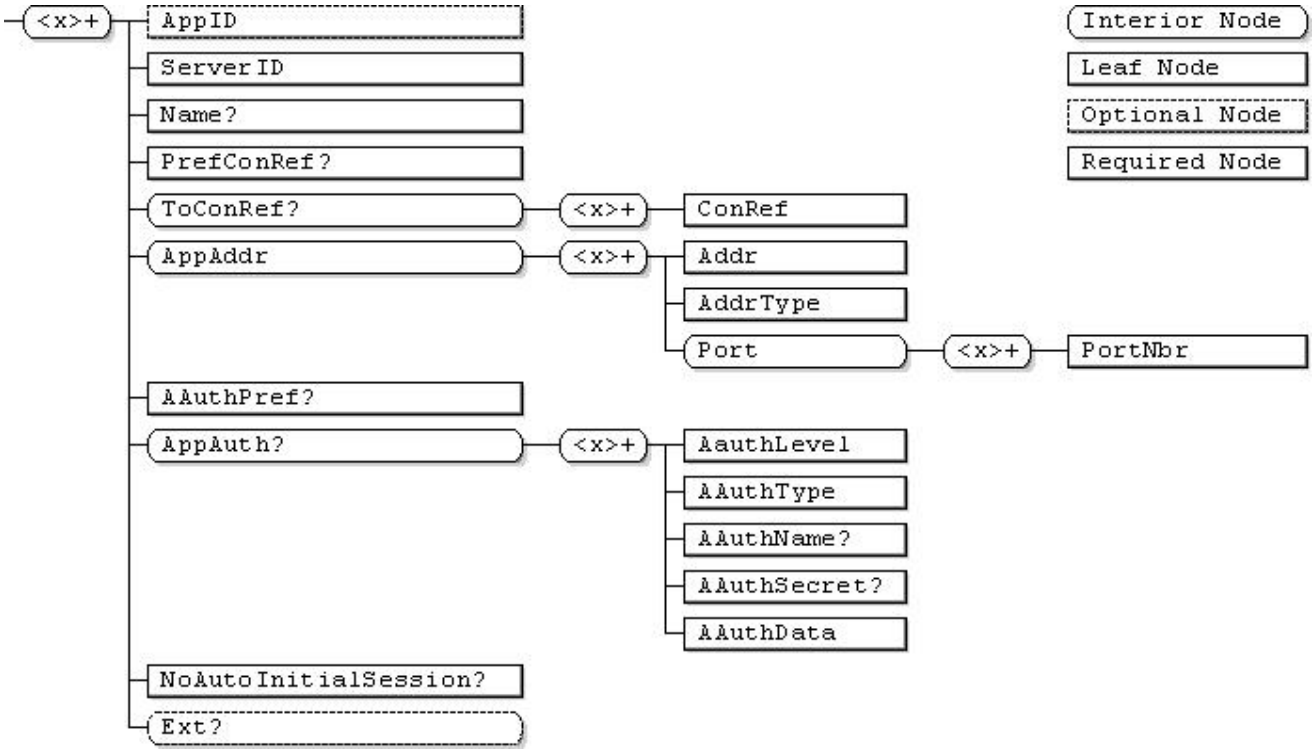


Figure 3: The DM Account Management Object

Parameters are also described in Device Management Application Characteristic registration document [w7] which is used as a part of OMA Client Provisioning specifications [ERELDCP]. General mapping rules of OMA Client Provisioning parameters are described in [DMBOOT]. When the DM Account parameters are derived from OMA Client Provisioning w7 document, see more information about parameter mapping in Appendix C.

The complete DDF description of this management object can be found in [DMAccDDF].

The DM Account Managed Object shown in Figure 3 may be located anywhere in the DM Tree. While there may be business cases for locating the DM Account Managed Object in a location other than at the root of the DM Tree, for most device management scenarios locating the DM Account Managed Object at the root of the DM Tree will meet the needs of the industry. Therefore it is recommended the DM Account Managed Object be located in the DM Tree as the URI ‘./DMAcc’.

An example for DM account URI under a fixed location is ‘./DMAcc/<x>/…..’.

The optional DMAcc node is an interior node acts as a placeholder for one or more DM Accounts as defined below:

- Occurrence: ZeroOrOne
- Format: Node
- Access Types: Get
- Values: N/A

…/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node acts as a placeholder for one or more accounts or for a fixed node. Management Object Identifier for the DMAcc MO MUST be: “urn:oma:mo:oma-dm-dmacc:1.1”.

AppID

Status	Occurrence	Format	Min. Access Types
Optional	One	chr	Get

This optional node specifies the application ID for device management account object. The value of this node, if present, MUST be ‘w7’.

ServerID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies a server identifier for management server used in the management session.

Name

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node specifies user displayable name for the management server.

PrefConRef

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node specifies a reference to preferred connectivity. It is expected that either a URI to proxy or NAP MO is specified, but other, implementation-specific connectoids MAY be referenced.

ToConRef

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node is used to allow application to refer to a collection of connectivity definitions. Several connectoids MAY be listed for a given application under this interior node.

ToConRef/<X>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node acts as a placeholder for one or more connectivity parameters.

ToConRef/<X>/ConRef

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node indicates the linkage to connectivity parameters, specified either as an URI to an MO or as an implementation-specific identifier.

AppAddr

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is used to specify multiple Management Server addresses.

AppAddr/<X>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node acts as a placeholder for separating one or more Server Addresses.

AppAddr/<X>/Addr

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies a Management Server address dependent upon AddrType.

AppAddr/<X>/AddrType

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies a Management Server address type. Valid values are: “URI”, “IPv4” or “IPv6”.

AppAddr/<X>/Port

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node acts as a placeholder for aggregating one or more Port settings.

AppAddr/<X>/Port/<X>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node acts as a placeholder for aggregating one or more Port settings.

AppAddr/<X>/Port/<X>/PortNbr

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies port number. The port number MUST be a decimal number and must fit within the range of a 16 bit unsigned integer.

AAuthPref

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This is a string-valued node whose possible value is exactly one of the names of the various possible authentication types (AAuthType values). E.g. "DIGEST". If this node is present, the client SHOULD use this authentication type when connecting to the server. The use of this node is intended to reduce the number of round trips between client and server that would be caused by authentication challenges. If the value is empty, the default behaviour is to indicate the authentication mechanism negotiated in the previous session if one exists.

See <x>/AppAuth/<x>/AAuthTypes in section 5.3.1.20 for possible values of this node.

AppAuth

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This node specifies authentication information.

AppAuth/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node acts as a placeholder for one or more authentication settings.

To ensure against misalignment of credentials with their correct Authentication Level (<X>/AppAuth/<X>/AAuthLevel) and avoid unnecessary processing within Device, the node name used for this node SHOULD be the value of AAuthLevel node under it,

For example:

<NodeName>AppAuth</NodeName>

<NodeName>CLCRED</NodeName>

<NodeName>AAuthLevel</NodeName>

<Value>CLCRED</Value>

AppAuth/<x>/AauthLevel

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the authentication level.

Values:

	Status	Interpretation
CLCRED	Optional	Credentials DM Client uses to authenticate itself to the OMA DM Server at the DM protocol level.
SRVCRED	Optional	Credentials DM Server uses to authenticate itself to the OMA DM Client at the DM protocol level.
OBEX	Optional	Credentials for OBEX authentication. NOTE: If this AAuthLevel is selected only HTTP-BASIC, HTTP-DIGEST and TRANSPORT are valid values for AAuthType.
HTTP	Optional	Credentials for HTTP (/WSP) authentication. NOTE: If this AAuthLevel is selected only HTTP-BASIC, HTTP-DIGEST and TRANSPORT are valid values for AAuthType.

Table 1: AAuthLevel Values

AppAuth/<x>/AAuthType

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This node specifies the authentication type.

Values:

	Status	Interpretation
HTTP-BASIC	Optional	HTTP basic authentication done according to RFC 2617.
HTTP-DIGEST	Optional	HTTP digest authentication done according to RFC 2617.
BASIC	Optional	DM 'syncml:auth-basic' authentication as specified in [DMSEC].
DIGEST	Optional	DM 'syncml:auth-md5' authentication as specified in [DMSEC].
HMAC	Optional	DM `'syncml:auth-MAC'` authentication as specified in [DMSEC].
X509	Optional	'syncml:auth-X509' authentication done according to [REPPRO].
SECURID	Optional	'syncml:auth-securid' authentication done according to [DMSEC].
SAFEWORD	Optional	'syncml:auth-safeword' authentication done according to [DMSEC].
DIGIPASS	Optional	'syncml:auth-digipass' authentication done according to [DMSEC].
TRANSPORT	Optional	Secure Transport authentication is used. Transport layer authentication is beyond the scope of OMA DM Security.

Table 2: AAuthType Values

AppAuth/<x>/AAuthName

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node specifies the authentication name.

AppAuth/<x>/AAuthSecret

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This node specifies the authentication secret.

AppAuth/<x>/AAuthData

Status	Occurrence	Format	Min. Access Types
Required	One	bin	No Get

This node specifies the authentication data relating to the AAuthType.

NoAutoInitialSession

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

If the NoAutoInitialSession leaf node is set to true, following completion of the DM Bootstrap operation, the DM Client MUST NOT attempt an untriggered connection to the DM Server. If set to false or omitted, the DM Client MUST conform to the normal bootstrap connection rule.

Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is where the vendor specific information about device management application is placed (vendor meaning application vendor, device vendor, OS vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include a non-standard sub-tree.

5.3.2 The DevInfo management object

Management object identifier: urn:oma:mo:oma-dm-devinfo:1.0

The following figure shows an overview of the DevInfo management object.

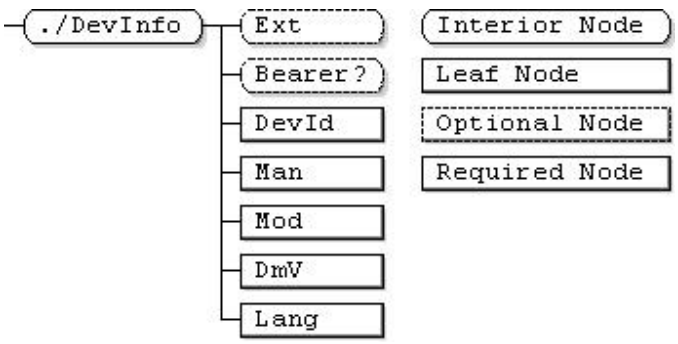


Figure 4: The DevInfo management object

The nodes making up DevInfo have the following meanings:

./DevInfo

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This required interior node specifies the unique object id of the DevInfo management object. Management Object Identifier for the DevInfo MO MUST be: “urn:oma:mo:oma-dm-devinfo:1.0”.

Ext

Status	Occurrence	Format	Min. Access Types
Optional	One	node	Get

An optional, interior node, designating the only branch of the DevInfo sub tree into which extensions can be added, permanently or dynamically.

Bearer

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

An optional, interior node in which items related to the bearer (CDMA, etc.) are stored. Use of this sub tree can be mandated by other standards.

DevId

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

A unique identifier for the device. SHOULD be globally unique and MUST be formatted as a URN as defined in [RFC2141].

Man

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The manufacturer identifier.

Mod

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

A model identifier (manufacturer specified string).

DmV

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

OMA device management client version identifier (manufacturer specified string).

Lang

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The current language setting of the device. The syntax of the language tags and their use are defined in [RFC1766]. Language codes are defined by ISO in the standard ISO639-2.

The complete DDF description of this management object can be found in [**Error! Reference source not found.**].

5.3.3 The DevDetail management object

Management object identifier: urn:oma:mo:oma-dm-devdetail:1.1

The following figure shows an overview of the DevDetail management object.

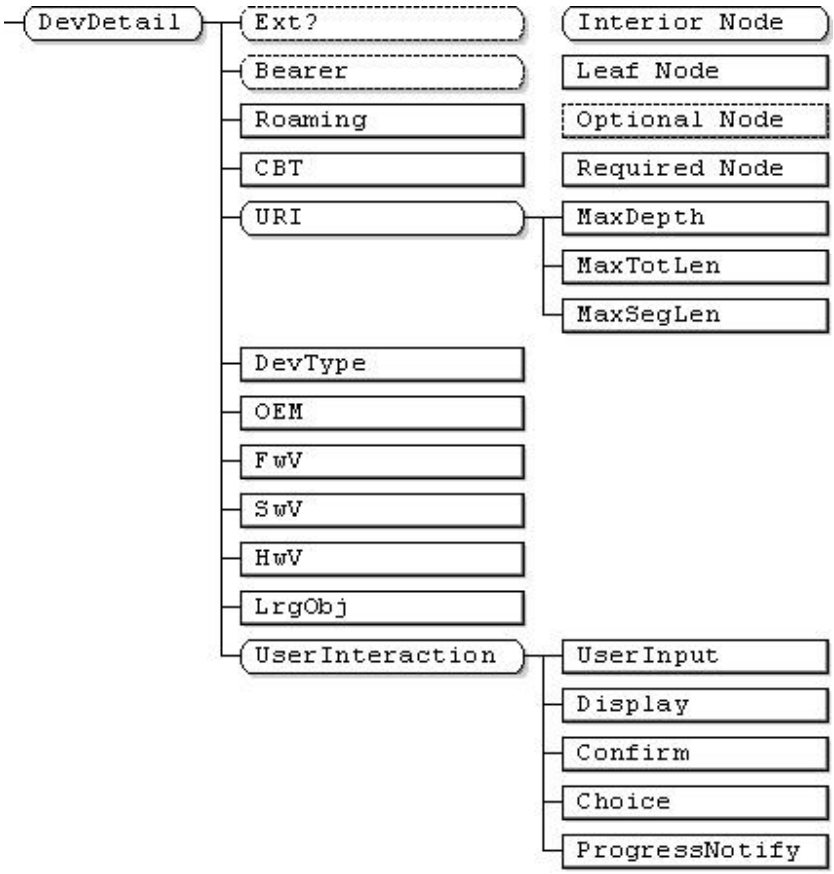


Figure 5: The DevDetail management object

The nodes making up DevDetail have the following meanings:

./DevDetail

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node specifies the unique object id of the DevDetail management object. Management Object Identifier for the DevDetail MO MUST be: “urn:oma:mo:oma-dm-devdetail:1.1”.

Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

An optional, interior node, designating the only branch of the DevDetail sub tree into which extensions can be added, permanently or dynamically.

Bearer

Status	Occurrence	Format	Min. Access Types
Optional	One	node	Get

An optional, interior node, designating a branch of the DevDetail sub tree into which items related to the bearer (CDMA, etc.) are stored. Use of this sub tree can be mandated by other standards.

Roaming

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This node indicated the current roaming status for the current DM session.

The following values are valid:

Value	**Description**
0	Current DM session is not over a roaming connection.
1	Current DM session is over a roaming connection
2	It is unknown if the current DM Session is over a roaming connection.

CBT

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This node provides bearer type information over which the DM session is currently being carried. The content of this node is Integer format with the range from 0 to 255, and currently the following values are allocated for different bearer types. For the bearer types not covered in this version of specification the value ‘0’ Other Bearer Type is to be used.

Bearer	**Value**
Other Bearer Type	0
3GPP Circuit Switched Bearer	1
3GPP Packet Switched Bearer	2
3GPP2 CDMA Packet Data Bearer	3
WLAN Bearer	4
LTE	5
WiMAX	6
OBEX	7
I-WLAN	8

URI

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node holds URI related information.

URI/MaxDepth

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Specifies the maximum depth of the management tree supported by the device. The maximum depth of the tree is defined as the maximum number of URI segments that the device supports. The value is a 16 bit, unsigned integer encoded as a numerical string. The value ‘0’ means that the device supports a tree of ‘unlimited’ depth.

URI/MaxTotLen

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Specifies the maximum total length of any URI used to address a node or node property. The maximum total length of a URI is defined as the largest total number of characters making up the URI which the device can support. Note that depending on the character set this might not be the same as the number of bytes. The value is a 16 bit, unsigned integer encoded as a numerical string. The value ‘0’ means that the device supports URI of ‘unlimited’ length.

URI/MaxSegLen

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Specifies the maximum total length of any URI segment in a URI used to address a node or node property.

The maximum total length of a URI segment is defined as the largest number of characters which the device can support in a single URI segment. Note that depending on the used character set this might not be the same as the number of bytes. The value is a 16 bit, unsigned integer encoded as a numerical string. The value ‘0’ means that the device supports URI segments of ‘unlimited’ length.

DevType

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Device type, for example. PDA, pager, or phone.

OEM

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Original Equipment Manufacturer of the device.

FwV

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Firmware version of the device.

SwV

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Software version of the device.

HwV

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

Hardware version of the device.

LrgObj

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

Indicates whether the device supports the OMA DM Large Object Handling specification, as defined in [DMPRO].

UserInteraction

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is a placeholder for the user interaction leaf nodes.

UserInteraction/UserInput

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

Indicates whether the device supports the User Input Alert, as defined in [DMPRO].

UserInteraction/Display

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

Indicates whether the device supports the Display Alert, as defined in [DMPRO].

UserInteraction/Confirm

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

Indicates whether the device supports the Confirmation Alert, as defined in [DMPRO].

UserInteraction/Choice

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

Indicates whether the device supports the Choice Alert, as defined in [DMPRO].

UserInteraction/ProgressNotify

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

Indicates whether the device supports the Progress Notification Alert, as defined in [DMPRO].

It is RECOMMENDED that the combination of HwV, SwV, FwV, Man, Mod, and OEM provide a unique signature identifying the specific version of software, thus providing a means for other implementations to make special provisions based on that identification.

The complete DDF description of this management object can be found in [**Error! Reference source not found.**].

5.3.4 Inbox

Management object identifier: urn:oma:mo:oma-dm-inbox:1.0

Inbox is designed to be used when the DM Server wants the DM Client to choose where to create a management object in the management tree. To have the DM Client perform this action, the DM Server will add a MO using TNDS to the ‘./Inbox’ URI, and the DM Client MUST add this MO into the DM tree using a location of DM Client’s choice.

For example, an operator would put a DMAcc and some connectivity MO into a single TNDS object, which in turn would be the data for an Add operation in a Bootstrap Message. This Bootstrap Message could then be provided via WAP Push or via a smartcard to a device for the DM Client to process. Since the DM Server may not know precisely where the DMAcc needs to reside, it will use the Inbox for the destination of the Add. How the DM Client will decide the location of the MO in the DM Tree is up to the DM Client, but might be chosen by the MOID of the new MO.

Inbox MUST be associated with the following fixed URI: "./Inbox"
Inbox MUST be supported by the DM Client.

DM Clients MUST only permit the *Add* operation on “./Inbox”. A DM Client MUST return the status code “Command not allowed” (405) in response to any command other than Add with targets “./Inbox”.

Inbox MUST support the ACL Runtime Property. The Inbox ACL property MUST be used to set access rights to DM Servers that are allowed to use this feature.

Inbox MUST only be used with TNDS objects [DMTNDS] - non-TNDS objects MUST be rejected with "Unsupported media type or format." (415).

Inbox MUST NOT support any child nodes. Commands that address child nodes of Inbox MUST be rejected with "Forbidden" (403).

The DM Client MUST copy the location of the new MO into the TargetRef element [DMREPPRO] in the returned Status command.

An example that illustrates the usage of the Inbox functionality is provided below

```
<Add>
  <CmdID>4</CmdID>
  <Item>
    <Target>
       <LocURI>./Inbox</LocURI>
    </Target>
    <Meta>
       <Format xmlns='syncml:metinf'>xml</Format>
       <Type xmlns='syncml:metinf'>
           application/vnd.syncml.dmtnds+xml
       </Type>
    </Meta>
    <Data>
        ... TNDS encoded Object ...
    </Data>
  </Item>
</Add>
```

Appendix A. (Informative)

A.1 Approved Version History

Reference	Date	Description
N/A	N/A	No prior 1.3 version

A.2 Draft/Candidate Version 1.3 History

Document Identifier	Date	Sections	Description
Draft versions OMA-TS-DM_StdObj-V1_3	15 Oct 2008	All	Baseline to v1.3., using OMA-TS-DM_StdObj-V1_2_1-20080617-A
	13 Apr 2009	5.3	Applied OMA-DM-DM13-2009-0005-CR_Bootstrap_Connect
	06 Jul 2009	5.3.1.8	Applied OMA-DM-DM13-2009-0036R02-CR_DMAcc_Authentication_Node_Naming OMA-DM-DM13-2009-0040-CR_DMAcc_Diagram
	09 Sep 2009	All	Applied OMA-DM-DM13-2009-0059R02-CR_Enhance_DevDetail
	28 Oct 2009	All	Applied OMA-DM-DM13-2009-0105R01-CR_Inbox_Only_Add OMA-DM-DM13-2009-0092-CR_Inbox_Not_Optional OMA-DM-DM13-2009-0074R02-CR_Add_StdObj_Tables
	17 Nov 2009	All	Applied OMA-DM-DM13-2009-0047R04-CR_DMAcc_Node_Name_Rooting OMA-DM-DM13-2009-0088R01-CR_StdObj_Bug_Fixes, OMA-DM-DM13-2009-0109R01-CR_StdObj_Cleanup
	02 Dec 2009	All	Corrected tables for node descriptions.
	28 Dec 2009	All	Applied OMA-DM-DM13-2009-0129-CR_StdObj_Missing_Nodes
	29 Dec 2009		Moved the URI text to the correct spot of DevDetail.
	13 Jan 2010	All	Applied OMA-DM-DM13-2009-0140R01-CR_StdObj_Diagrams DSO editorial clean-up
	14 Jan 2010	5.3.3	Applied OMA-DM-DM13-2010-0003-CR_DevDetail_Missing_Nodes
	04 Feb 2010	5.3.3	Applied OMA-DM-DM13-2010-0007R05-CR_Bearer_Type OMA-DM-DM13-2010-0005-CR_StdObj_SCR_Version_Fix OMA-DM-DM13-2009-0124R01-CR_RoamingInfo OMA-DM-DM13-2009-0120R02-CR_Inbox_Usage_Clarification.doc OMA-DM-DM13-2010-0032R01-CR_Inbox_Fixes.doc OMA-DM-DM13-2010-0019-CR_Correct_Description_For_Inbox_Mandatory.zip
	11 Feb 2010	All	Editorial clean-up of formatting by DSO
	15 Apr 2010	All	Applied OMA-DM-DM13-2010-0041-CR_StdObj_Cleanup OMA-DM-DM13-2010-0058-CR_Improve_Inbox_in_StdObj
	16 Apr 2010	All	Really applied OMA-DM-DM13-2010-0058-CR_Improve_Inbox_in_StdObj.
	26 Apr 2010	2.1	Formatting of one hyperlink
	04 May 2010	All	Applied OMA-DM-DM13-2010-0073-CR_Remove_Nonce_Resync Formatting of a few tables borders, double quotes changed to single quotes
Candidate version OMA-TS-DM_StdObj-V1_3	25 May 2010	N/A	Status changed to Candidate by TP Ref # OMA-TP-2010-0221-INP_DM_V1.3_ERP_and_ETR_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for DM Client

Item	Function	Reference	Status	Requirement
SCR-DM-STDOBJ-C-001	Support of DevInfo object	Section 5.3.2	M	
SCR-DM-STDOBJ-C-002	Support of DevDetail Object	Section 5.3.3	M	
SCR-DM-STDOBJ-C-003	Support of DM Account Object	Section 5.3.1	M	
SCR-DM-STDOBJ-C-004	Support of Inbox Object	Section 5.3.4	M	

B.2 SCR for DM Server

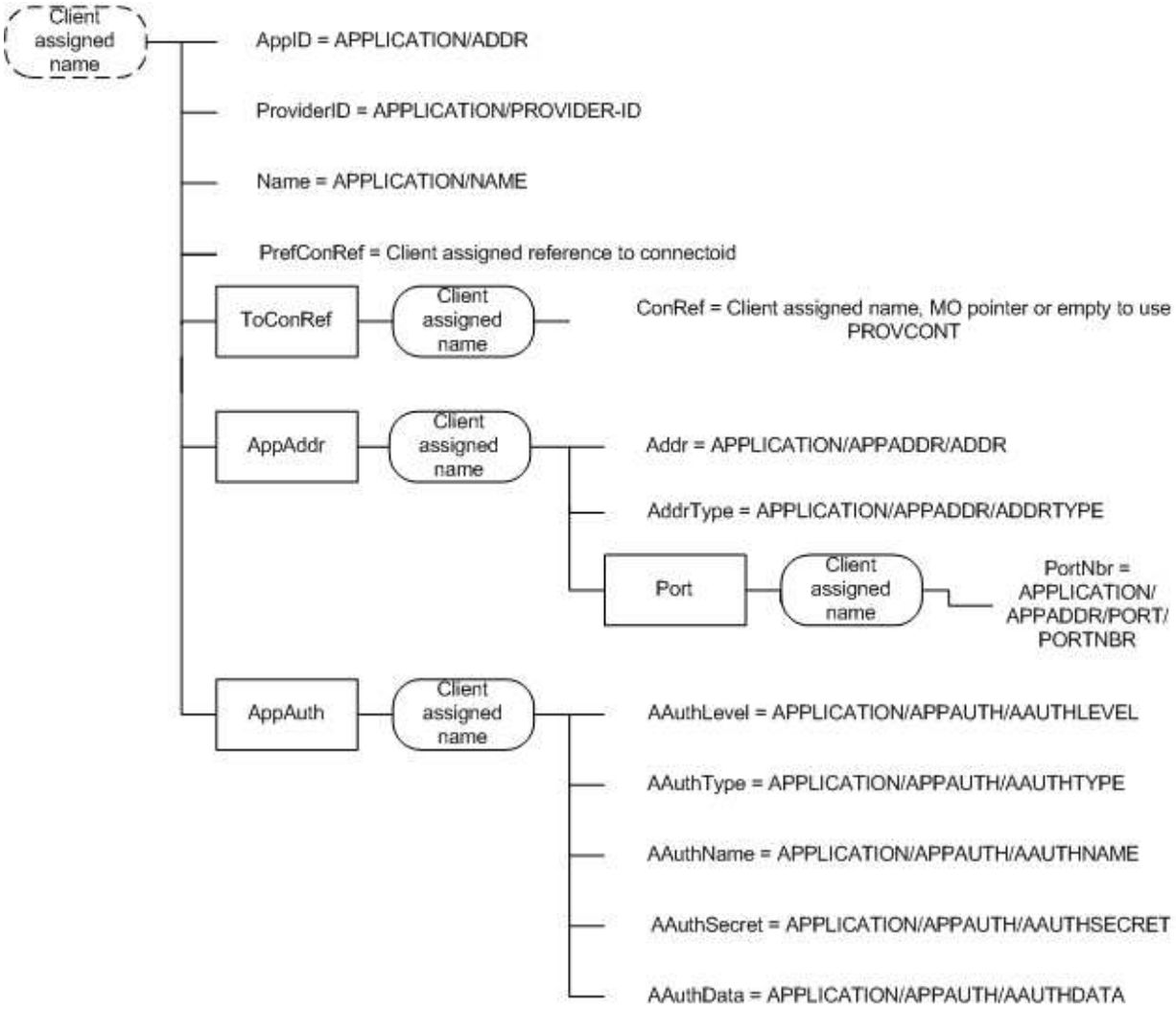
Item	Function	Reference	Status	Requirement
SCR-DM-STDOBJ-S-001	Support of DevInfo object	Section 5.3.2	M	
SCR-DM-STDOBJ-S-002	Support of DevDetail Object	Section 5.3.3	M	
SCR-DM-STDOBJ-S-003	Support of DM Account Object	Section 5.3.1	M	
SCR-DM-STDOBJ-S-004	Support of Inbox Object	Section 5.3.4	M	

Appendix C. Mapping of Device Management parameters

In the below table the Device Management Account Management Object and Device Management Provisioning Content Application characteristic (w7) parameter correspondence is shown.

DEVICE MANAGEMENT ACCOUNT MANAGEMENT OBJECT	DM PROVISIONING CONTENT APPLICATION CHARACTERISTIC (w7)
AppID	APPID
ServerID	PROVIDER-ID
Name	NAME
PrefConRef, if multiple ToConRef/<X>/ConRef	TO-PROXY
PrefConRef, if multiple ToConRef/<X>/ConRef	TO-NAPID
AppAddr/<X>/Addr	APPADDR/ADDR
AppAddr/<X>/AddrType	APPADDR/ADDRTYPE
AppAddr/<X>/Port/<X>/PortNbr	APPADDR/PORT/PORTNBR
AauthPref	N/A
AppAuth/<X>/AAuthLevel	APPAUTH/AAUTHLEVEL
AppAuth/<X>/AAuthType	APPAUTH/AAUTHTYPE
AppAuth/<X>/AAuthName	APPAUTH/AAUTHNAME
AppAuth/<X>/AAuthSecret	APPAUTH/AAUTHSECRET
AppAuth/<X>/AAuthData	APPAUTH/AAUTHDATA

The following diagram shows how information from the provisioning content and the w7 characteristic are mapped to the management tree.



Requirements for DM client when it converts the w7 APPLICATION characteristic to the management tree:

- DM Client MUST assign a unique name for the <X> (DMAcc Interior node) as specified in Section 5.3.2 in [DMBOOT]. Management server can modify this node name in some subsequent DM session.
- The DM Client MUST grant Get, Replace and Delete ACL rights to the specified ServerId for the <X> (DMAcc Interior node) as specified in Section 5.3.4 in [DMBOOT]. The provisioning server MAY modify this ACL to provide broader or narrower access in a subsequent DM session.

The values of each leaf in the DMAcc object is derived from a w7 APPLICATION characteristic as follows:

- **AppID** – takes the value of the APPLICATION/APPID = w7.
- **ServerID** – takes the value of APPLICATION/PROVIDER-ID
- **Name** – takes the value of APPLICATION/NAME

- **PrefConRef** – client assigned reference to connectoid, e.g. Connectivity MO or connection information maintained outside of the management tree, for example as specified within PXLOGICAL and NAPDEF.
- **ToConRef/<X>/ConRef** - client assigned name, MO pointer or may be left empty by the DM client to use connection information maintained outside of the management tree, for example as specified within PXLOGICAL and NAPDEF.
- **AppAddr/<X>/Addr** – takes the value of APPLICATION/APPADDR/ADDR
- **AppAddr/<X>/AddrType** - takes the value of APPLICATION/APPADDR/ADDRTYPE
- **AppAddr/<X>/Port/<X>/PortNbr** – takes the value of APPLICATION/APPADDR/PORT/PORTNBR
- **AppAuth/<X>/AAuthLevel** – correspondence to APPLICATION/APPAUTH/AAUTHLEVEL values is as follows:

w7 APPLICATION/APPAUTH/AAUTHLEVEL	DMAcc AppAuth/<x>/AAuthLevel
APPSRV	CLCRED
CLIENT	SRVCRED
OBEX	OBEX

- **AppAuth/<X>/AAuthType** – takes the value of APPLICATION/APPAUTH/AAUTHTYPE
- **AppAuth/<X>/AAuthName** – takes the value of APPLICATION/APPAUTH/AAUTHNAME
- **AppAuth/<X>/AAuthSecret** – takes the value of APPLICATION/APPAUTH/AAUTHSECRET
- **AppAuth/<X>/AAuthData** – takes the value of APPLICATION/APPAUTH/AAUTHDATA