OMA Device Management Standardized Objects

Approved Version 1.2.1 – 17 Jun 2008

Open Mobile Alliance

OMA-TS-DM_StdObj-V1_2_1-20080617-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.

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

[DMAccDDF] “OMA Device Management Account Management Object DDF, Version 1.2”. Open Mobile Alliance™. OMA-SUP-MO_DM_DMAcc-V1_2. URL:http://www.openmobilealliance.org

[DMBOOT] “OMA Device Management Bootstrap, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM_Bootstrap-V1_2. URL:http://www.openmobilealliance.org

[DMDDFDTD] “OMA DM Device Description Framework DTD, Version 1.2”. Open Mobile Alliance™. URL:http://www.openmobilealliance.org/tech/DTD/dm_ddf-v1_2.dtd

[DevDetailDDF] “OMA Device Management Detail Information Management Object DDF, Version 1.2”. Open Mobile Alliance™. OMA-SUP-MO_DM_DevDetail-V1_2. URL:http://www.openmobilealliance.org

[DevInfoDDF] “OMA Device Management Information Management Object DDF, Version 1.2”. Open Mobile Alliance™. OMA-SUP-MO_DM_DevInfo-V1_2. URL:http://www.openmobilealliance.org

[DMPRO] “OMA Device Management Protocol, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM_Protocol-V1_2. URL:http://www.openmobilealliance.org

[DMSEC] “OMA Device Management Security, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM_Security-V1_2. URL:http://www.openmobilealliance.org

[DMTND] “OMA Device Management Tree and Description, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM_TND-V1_2. URL:http://www.openmobilealliance.org

[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/

[REPPRO] “SyncML Representation Protocol”, Version 1.2, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL:http://www.openmobilealliance.org/

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

[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

[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. These objects are primarily associated with OMA DM and SyncML configuration.

Since device manufacturers will always develop new functions in their devices and since these functions often are proprietary, no standardized management objects exist for them. To make these functions manageable in the devices that have them, a device description framework is needed that can provide servers with the necessary information they must have in order to manage the new functions. The intention with this framework is that device manufacturers will publish descriptions of their devices as they enter the market. Organizations operating device management servers should then only have to feed the new description to their servers for them to automatically recognize and manage the new functions in the 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.

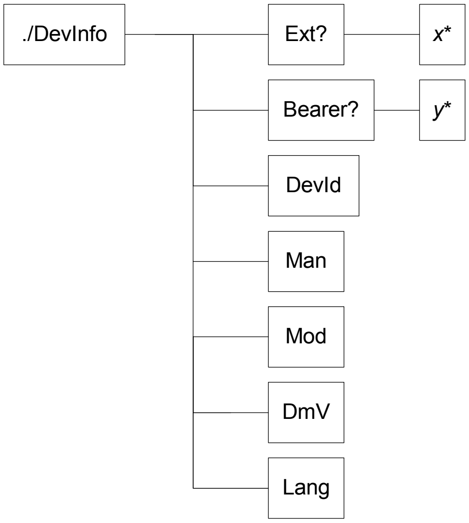


Figure 1. Example of a management object pictured using the graphical notation

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.

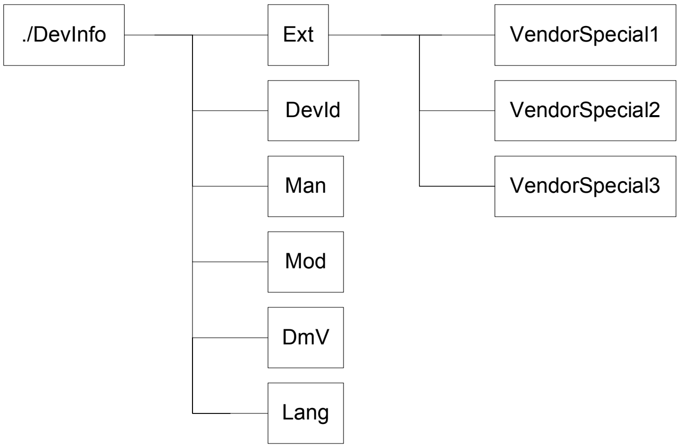


Figure 2. Example of an instantiated ./DevInfo object

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. 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	MAY	MAY	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:1.0

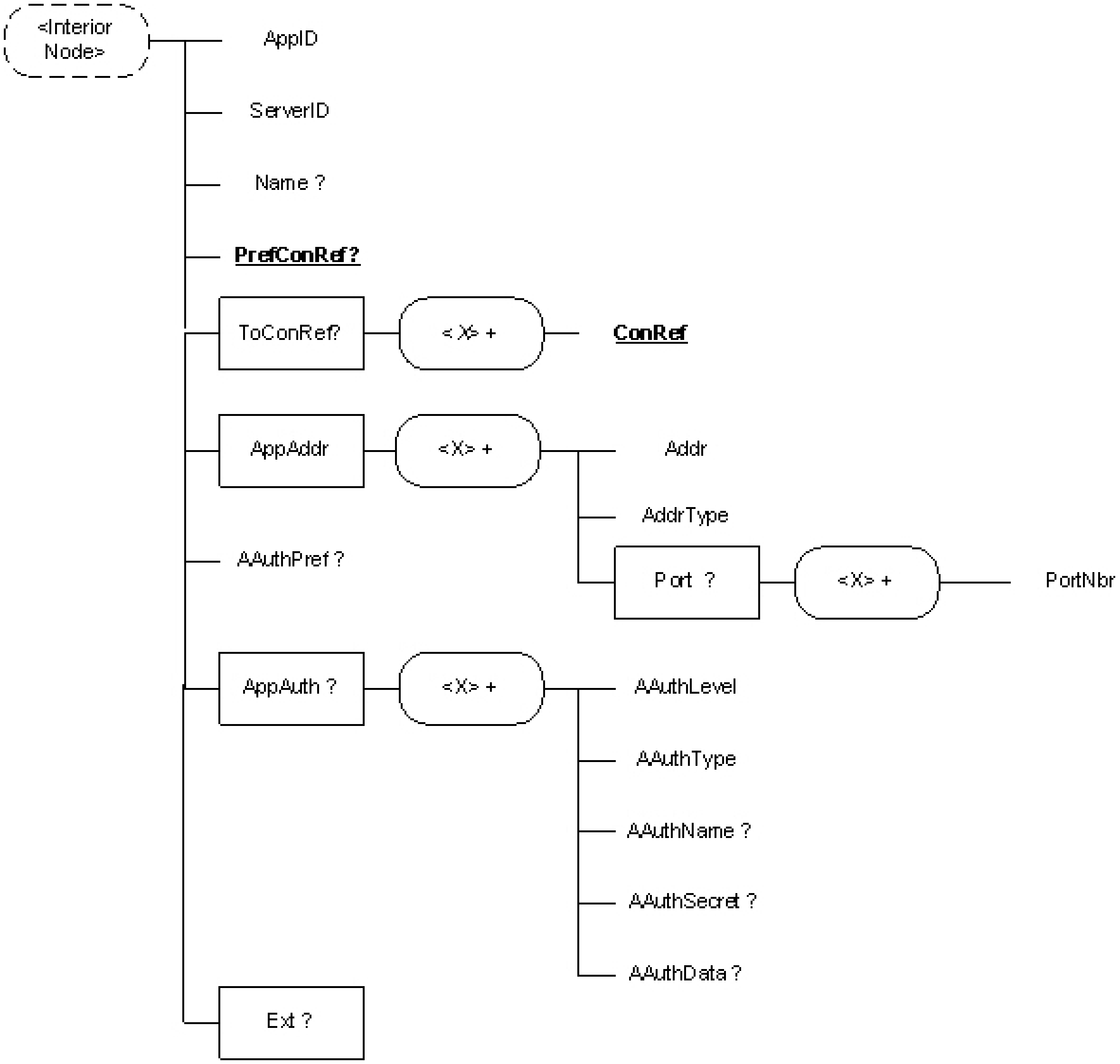


Figure 3. The DM Account Management Object

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].

5.3.1.1 Node: <X>

This interior node acts as a placeholder for one or more accounts or for a fixed node.

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

5.3.1.2 Node: <X>/AppID

This node specifies the application ID for device management account object.

- Occurrence: One
- Format: Chr
- Access Types: Get
- Values: w7

5.3.1.3 Node: <X>/ServerID

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

- Occurrence: One
- Format: Chr
- Access Types: Get
- Values: <Server identifier>

5.3.1.4 Node: <X>/Name

This node specifies user displayable name for the management server.

- Occurrence: ZeroOrOne
- Format: Chr
- Access Types: Get
- Values: <User displayable name>

5.3.1.5 Node: <X>/PrefConRef

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

- Occurrence: ZeroOrOne
- Format: Chr

- Access Types: Get
- Values: URI to a management object or implementation specific identifier.

5.3.1.6 Node: <X>/ToConRef/

The ToConRef 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.

- Occurrence: ZeroOrOne
- Format: Node
- Access Type: Get
- Value: N/A

5.3.1.7 Node: <X>/ToConRef/<X>

This run-time node acts as a placeholder for one or more connectivity parameters.

- Occurrence: OneOrMore
- Format: Node
- Access Type: Get
- Value: N/A

5.3.1.8 Node: <X>/ToConRef/<X>/ConRef

The ConRef leaf indicates the linkage to connectivity parameters.

- Occurrence: One
- Format: Chr
- Access Type: Get
- Value: URI to a management object or implementation specific identifier.

5.3.1.9 Node: <X>/AppAddr

This node is used to specify multiple Management Server addresses.

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

5.3.1.10 Node: <X>/AppAddr/<X>

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

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

5.3.1.11 Node: <X>/AppAddr/<X>/Addr

This node specifies a Management Server address.

- Occurrence: One
- Format: Chr
- Access Types: Get
- Values: Dependent upon AddrType.

5.3.1.12 Node: <X>/AppAddr/<X>/AddrType

This node specifies a Management Server address type.

- Occurrence: One
- Format: Chr
- Access Types: Get
- Values: “URI”, “IPv4” or “IPv6”. If no value exists the default type MUST be “URI”.

5.3.1.13 Node: <X>/AppAddr/<X>/Port

This node specifies Port information for Management Server address.

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

5.3.1.14 Node: <X>/AppAddr/<X>/Port/<X>

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

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

5.3.1.15 Node: <X>/AppAddr/<X>/Port/<X>/PortNbr

This node specifies port number.

- Occurrence: One
- Format: Chr
- Access Types: Get
- Values: The port number MUST be a decimal number and must fit within the range of a 16 bit unsigned integer.

5.3.1.16 Node: <X>/AAuthPref

This is a string-valued parameter whose possible values are 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 a client supports this leaf node and the value is empty, the default behaviour is to indicate the authentication mechanism negotiated in the previous session if one exists.

- Occurrence: ZeroOrOne
- Format: Chr
- Access Types: Get
- Values: See AAuthTypes in section 5.3.1.20.

5.3.1.17 Node: <X>/AppAuth

This node specifies authentication information.

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

5.3.1.18 Node: <X>/AppAuth/<X>

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

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

5.3.1.19 Node: <X>/AppAuth/<X>/AAuthLevel

This node specifies the authentication level.

- Occurrence: One
- Format: Chr
- Access Types: Get
- Values:

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

Table 1: AAuthLevel Values

5.3.1.20 Node: <X>/AppAuth/<X>/AAuthType

This node specifies the authentication type.

- Occurrence: One

- Format: Chr
- Access Types: Get
- 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 [REPPRO].
SAFEWORD	Optional	'syncml:auth-safeword' authentication done according to [REPPRO].
DIGIPASS	Optional	'syncml:auth-digipass' authentication done according to [REPPRO].
TRANSPORT	Optional	Secure Transport authentication is used. Transport layer authentication is beyond the scope of OMA DM Security.

Table 2: AAuthType Values

5.3.1.21 Node: <X>/AppAuth/<X>/AAuthName

This node specifies the authentication name.

- Occurrence: ZeroOrOne
- Format: Chr
- Access Types: Get
- Values: <Authentication name>

5.3.1.22 Node: <X>/AppAuth/<X>/AAuthSecret

This node specifies the authentication secret.

- Occurrence: ZeroOrOne
- Format: Chr
- Access Types: No Get
- Values: <Authentication secret>

5.3.1.23 Node: <X>/AppAuth/<X>/AAuthData

This node specifies the authentication data relating to the AAuthType.

- Occurrence: ZeroOrOne
- Format: Bin
- Access Types: No Get
- Values: <Authentication data>

5.3.1.24 Node: <X>/Ext/

The Ext is an interior node for where the vendor specific information about device management application is being 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.

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

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.

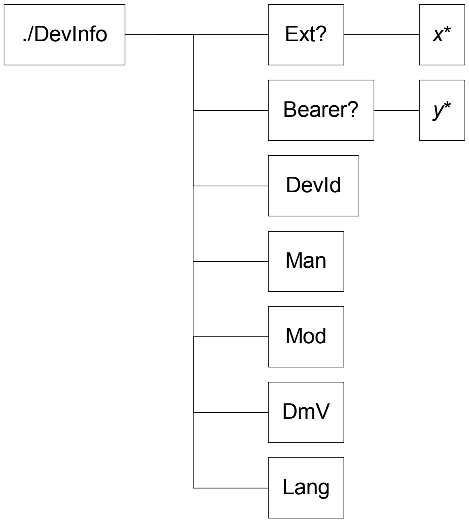


Figure 4. The DevInfo management object

The nodes making up DevInfo have the following meanings:

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

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

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

Man
The manufacturer identifier.

Mod
A model identifier (manufacturer specified string).

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

Lang
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.

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

5.3.3 The DevDetail management object

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

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

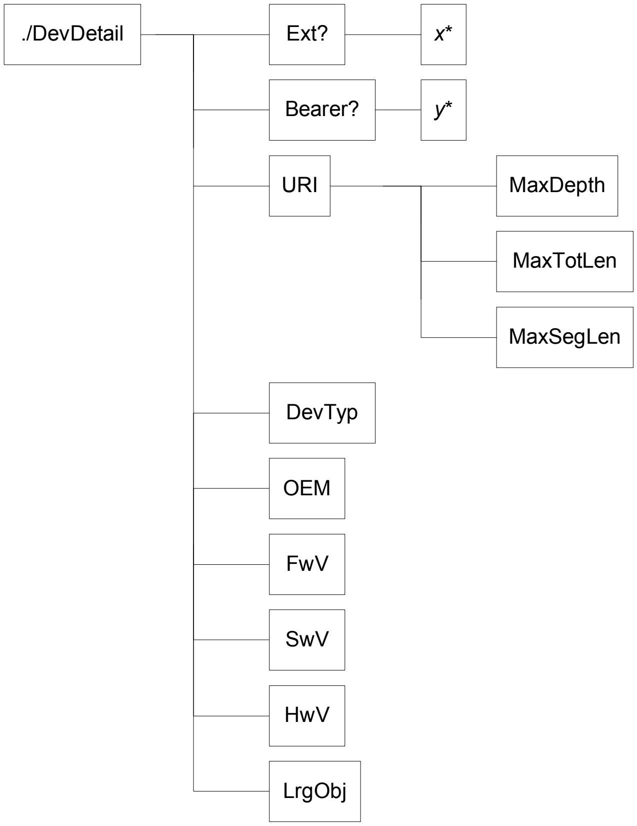


Figure 5. The DevDetail management object

The nodes making up DevDetail have the following meanings:

Ext

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

Bearer

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.

URI/MaxDepth

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

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

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.

DevTyp

Device type, e.g. PDA, pager, or phone.

OEM

Original Equipment Manufacturer of the device.

FwV

Firmware version of the device.

SwV

Software version of the device.

HwV

Hardware version of the device.

LrgObj

Indicates whether the device supports the OMA DM Large Object Handling specification, 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 [DevDetailDDF].

5.3.4 The Inbox URI

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

In some circumstances a Management Object's URI is not the preferred addressing method and the management object identifier is enough information for the device to resolve a suitable location for that Management Object. In that case the URI: "./Inbox" is a reserved location for this purpose.

For example a device's DDF description MAY indicate if that device is supporting the "./Inbox" concept. In that definition it is possible to define the only access type "Add". Then a server MAY send a Management Object to that device with the URI: "./Inbox" and then the device SHOULD use the management object identifier to resolve the correct location in the management tree to add that Management Object. In this case the server can not Get that object from the URI: "./Inbox" after it is added. The client's ACL for "./Inbox" MAY also set access rights so only some servers are allowed to use this feature.

DM Clients MUST NOT permit a *Get* operation on the "./Inbox". A DM Client SHALL return the status code *"Command not allowed"* (405) in response to a *Get* which targets "./Inbox" or any direct or indirect child node of "./Inbox".

The following figure shows an overview of the Inbox URI:

Figure 6. The Inbox URI

The node has the following meaning:

Inbox

An optional, interior node.

Appendix A. (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-SyncML-DMStdObj-V1_1_2-20031209-A	09 Dec 2003	SyncML DM Standardized Objects V1.1.2 Approved Release.
OMA-TS-DM_StdObj-V1_2-20070209-A	09 Feb 2007	Status changed to Approved by TP TP Doc ref# OMA-TP-2007-0075R03-INP_ERP_DM_V1.2_for_Final_Approval
OMA-TS-DM_StdObj-V1_2_1-20080521-D	21 May 2008	Updated with agreed CRs: OMA-DM-2007-0066 OMA-DM-2008-0038R02 OMA-DM-2008-0068 Approved history corrected
OMA-TS-DM_StdObj-V1_2_1-20080617-A	17 Jun 2008	Approved by TP TP ref# OMA-TP-2008-0257R01-INP_DM_V1_2_1_ERP_for_notification

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for DM v1.2 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	O	

B.2 SCR for DM v1.2 Server

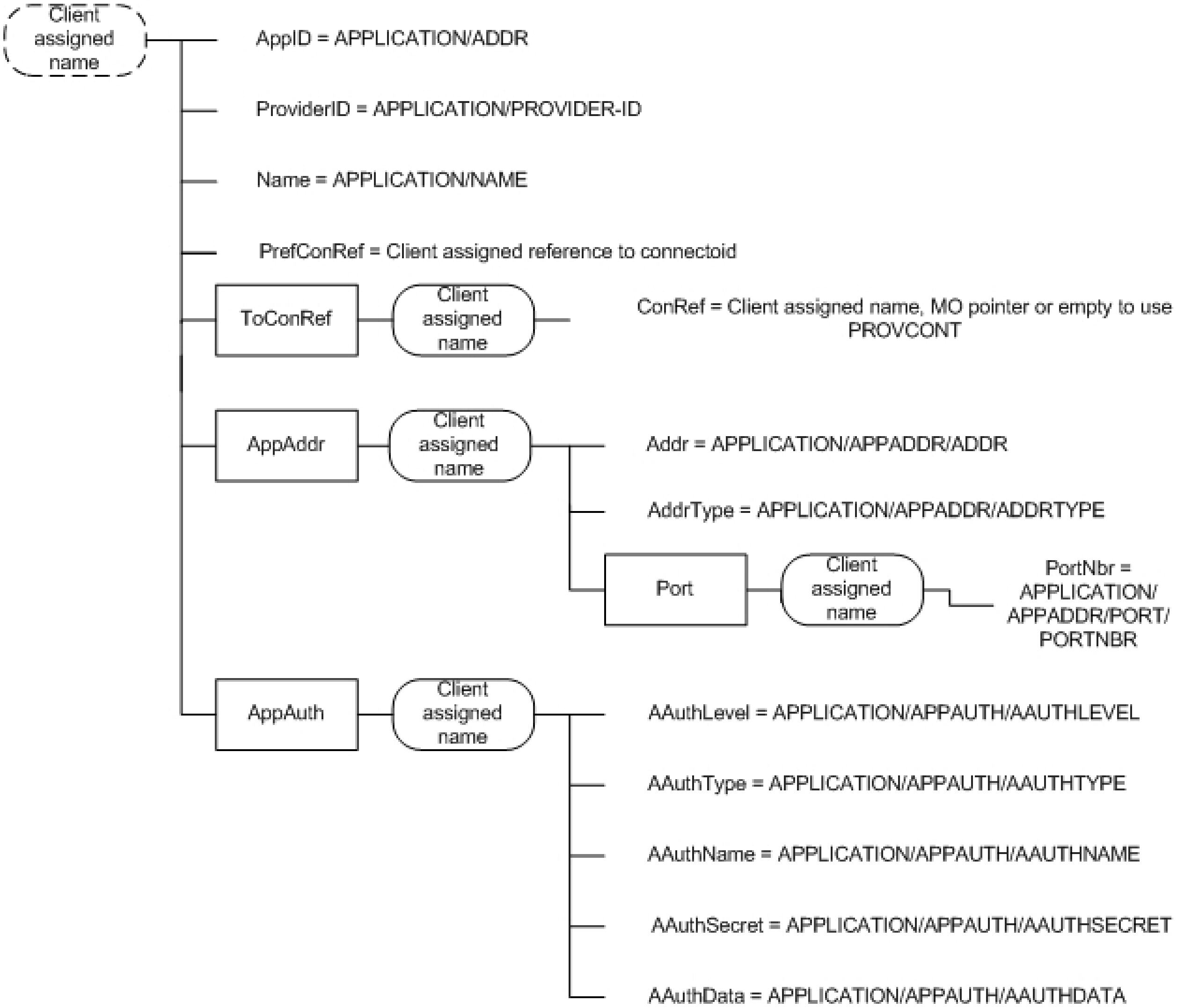
Item	Function	Reference	Status	Requirement
SCR-DM-STDOBJ-S-001	Support of DevInfo object	Section 5.3.2	M	
SCR-DM-STDOBJ-S-002	Support of DevDetail Object	Section 5.3.3	M	
SCR-DM-STDOBJ-S-003	Support of DM Account Object	Section 5.3.1	M	
SCR-DM-STDOBJ-S-004	Support of Inbox Object	Section 5.3.4	O	

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.


Client assigned name
AppID = APPLICATION/ADDR
ProviderID = APPLICATION/PROVIDER-ID
Name = APPLICATION/NAME
PrefConRef = Client assigned reference to connectoid
ToConRef
Client assigned name
ConRef = Client assigned name, MO pointer or empty to use PROVCONT
AppAddr
Client assigned name
Addr = APPLICATION/APPADDR/ADDR
AddrType = APPLICATION/APPADDR/ADDRTYPE
Port
Client assigned name
PortNbr = APPLICATION/ APPADDR/PORT/ PORTNBR
AppAuth
Client assigned name
AAuthLevel = APPLICATION/APPAUTH/AAUTHLEVEL
AAuthType = APPLICATION/APPAUTH/AAUTHTYPE
AAuthName = APPLICATION/APPAUTH/AAUTHNAME
AAuthSecret = APPLICATION/APPAUTH/AAUTHSECRET
AAuthData = APPLICATION/APPAUTH/AAUTHDATA

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