Contents

1. SCOPE ...........................................................................................................................................................................4
2. REFERENCES ........................................................................................................................................................................5
   2.1 NORMATIVE REFERENCES ...........................................................................................................................................5
   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. STANDARDIZED OBJECTS ...............................................................................................................................................9
   5.1 MANAGEMENT OBJECTS ...........................................................................................................................................9
      5.1.1 Definition and description of management objects ..............................................................................................9
      5.1.2 DDF compliance .................................................................................................................................................11
   5.2 MANAGEMENT OBJECTS STANDARDIZED BY OTHER ORGANIZATIONS ..............................................................11
   5.3 THE OMA DM MANAGEMENT OBJECTS ................................................................................................................11
      5.3.1 The DM Account management object ..............................................................................................................12
      5.3.2 The DevInfo management object ....................................................................................................................20
      5.3.3 The DevDetail management object ................................................................................................................21
      5.3.4 The Inbox URI ..................................................................................................................................................23
APPENDIX A. (INFORMATIVE) ..............................................................................................................................................25
   A.1 APPROVED VERSION HISTORY ..................................................................................................................................25
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE) ......................................................................26
   B.1 SCR FOR DM v1.2 CLIENT ..........................................................................................................................................26
   B.2 SCR FOR DM v1.2 SERVER .......................................................................................................................................26
APPENDIX C. MAPPING OF DEVICE MANAGEMENT PARAMETERS ............................................................................27

Figures

Figure 1. Example of a management object pictured using the graphical notation ......................................................10
Figure 2. Example of an instantiated ./DevInfo object ................................................................................................10
Figure 3. The DM Account Management Object ..................................................................................................13
Figure 4. The DevInfo management object .............................................................................................................21
Figure 5. The DevDetail management object ...........................................................................................................22
Figure 6. The Inbox URI .............................................................................................................................................24

Tables

Table 1: AAuthLevel Values .............................................................................................................................................18
Table 2: AAuthType Values ............................................................................................................................................19
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


[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


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.

<table>
<thead>
<tr>
<th>Character</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>one or many occurrences</td>
</tr>
<tr>
<td>*</td>
<td>zero or more occurrences</td>
</tr>
<tr>
<td>?</td>
<td>zero or one occurrences</td>
</tr>
</tbody>
</table>

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 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 OneOrMore 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 OneOrMore 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.

<table>
<thead>
<tr>
<th>Management Object</th>
<th>Client Support</th>
<th>Server Support</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMAcc</td>
<td>MUST</td>
<td>MUST</td>
<td>Settings for the DM client in a managed device.</td>
</tr>
<tr>
<td>DevInfo</td>
<td>MUST</td>
<td>MUST</td>
<td>Device information for the OMA DM server. Sent from the client to the server.</td>
</tr>
</tbody>
</table>
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
5.3.1.1 Node: <X>

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

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].
5.3.1.2  **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
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:

<table>
<thead>
<tr>
<th>Status</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLCRED</td>
<td>Optional</td>
</tr>
<tr>
<td>SRVCREC</td>
<td>Optional</td>
</tr>
<tr>
<td>OBEX</td>
<td>Optional</td>
</tr>
<tr>
<td>HTTP</td>
<td>Optional</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLCRED</td>
<td>Optional</td>
</tr>
<tr>
<td>SRVCREC</td>
<td>Optional</td>
</tr>
<tr>
<td>OBEX</td>
<td>Optional</td>
</tr>
<tr>
<td>HTTP</td>
<td>Optional</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLCRED</td>
<td>Optional</td>
</tr>
<tr>
<td>SRVCREC</td>
<td>Optional</td>
</tr>
<tr>
<td>OBEX</td>
<td>Optional</td>
</tr>
<tr>
<td>HTTP</td>
<td>Optional</td>
</tr>
</tbody>
</table>

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:

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

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.
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.

![DevDetail Diagram]

**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:

![Inbox URI](image)

**Figure 6. The Inbox URI**

The node has the following meaning:

**Inbox**

An optional, interior node.
## Appendix A. (Informative)

### A.1 Approved Version History

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMA-SyncML-DMStdObj-V1_1_2-20031209-A</td>
<td>09 Dec 2003</td>
<td>SyncML DM Standardized Objects V1.1.2 Approved Release.</td>
</tr>
<tr>
<td>OMA-TS-DM_StdObj-V1_2-20070209-A</td>
<td>09 Feb 2007</td>
<td>Status changed to Approved by TP. TP Doc ref# OMA-TP-2007-0075R03-INP_ERP_DM_V1.2_for_Final_Approval</td>
</tr>
<tr>
<td>OMA-TS-DM_StdObj-V1_2_1-20080617-A</td>
<td>17 Jun 2008</td>
<td>Approved by TP. TP ref# OMA-TP-2008-0257R01-INP_DM_V1_2_1_ERP_for_notification</td>
</tr>
</tbody>
</table>
Appendix B. Static Conformance Requirements

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

### B.1 SCR for DM v1.2 Client

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Status</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCR-DM-STDOBJ-C-001</td>
<td>Support of DevInfo object</td>
<td>Section 5.3.2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>SCR-DM-STDOBJ-C-002</td>
<td>Support of DevDetail Object</td>
<td>Section 5.3.3</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>SCR-DM-STDOBJ-C-003</td>
<td>Support of DM Account Object</td>
<td>Section 5.3.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>SCR-DM-STDOBJ-C-004</td>
<td>Support of Inbox Object</td>
<td>Section 5.3.4</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

### B.2 SCR for DM v1.2 Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Function</th>
<th>Reference</th>
<th>Status</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCR-DM-STDOBJ-S-001</td>
<td>Support of DevInfo object</td>
<td>Section 5.3.2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>SCR-DM-STDOBJ-S-002</td>
<td>Support of DevDetail Object</td>
<td>Section 5.3.3</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>SCR-DM-STDOBJ-S-003</td>
<td>Support of DM Account Object</td>
<td>Section 5.3.1</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>SCR-DM-STDOBJ-S-004</td>
<td>Support of Inbox Object</td>
<td>Section 5.3.4</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>
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.

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

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:

<table>
<thead>
<tr>
<th>APPLICATION/APPAUTH/AAUTHLEVEL</th>
<th>DMAcc AppAuth/&lt;x&gt;/AAuthLevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPSRV</td>
<td>CLCRED</td>
</tr>
<tr>
<td>CLIENT</td>
<td>SRVCREDS</td>
</tr>
<tr>
<td>OBEX</td>
<td>OBEX</td>
</tr>
</tbody>
</table>

- **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