OMA Device Management Tree and Description Serialization

Approved Version 1.3 – 24 May 2016

Open Mobile Alliance

OMA-TS-DM_TNDS-V1_3-20160524-A

Contents

Figures

Tables

1. Scope

This specification defines serialization and deserialization operations to transform a runtime management tree into an XML or WBXML file. With this specification it is possible to transform a MO or complete/part of the Management Tree to or from an XML or WBXML file.

2. References

2.1 Normative References

[DMDICT] “OMA Device Management Dictionary, Version 1.0”. Open Mobile Alliance™. OMA-SUP-DM_Dictionary-v1_0. URL:http://www.openmobilealliance.org/

[DMREPU] “OMA Device Management Representation Protocol, Version 1.3”. Open Mobile Alliance™. OMA-TS-DM_RepPro-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

[ISO8601] ISO 8601:2000, Data elements and interchange formats -- Information interchange -- Representation of dates and times URL:http://www.iso.ch/

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

[RFC2396] “Uniform Resource Identifiers (URI): Generic Syntax”. T. Berners-Lee, et al. August 1998. URL:http://www.ietf.org/rfc/rfc2396.txt

[SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org

[XMLSCHEMADT] “XML Schema Part 2: Datatypes”, W3C Recommendation 02 May 2001, URL:http://www.w3.org/XML/Schema/

2.2 Informative References

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

[DMSTDOBJ] “OMA Device Management Standardized Objects, Version 1.3”. Open Mobile Alliance™. OMA-TS-DM_StdObj-V1_3. URL:http://www.openmobilealliance.org

3. Terminology and Conventions

3.1 Conventions

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

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

3.2 Definitions

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

3.3 Abbreviations

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

4. Introduction

This specification defines how to convert a runtime Management Tree or sub tree (i.e. an MO) into an XML or WBXML structure. This specification is useful whenever a MO needs to be created from one entity, moved outside of the DM Protocol session to another entity in an interoperable fashion.

There are three scenarios in which the need for Management Tree serialization and deserialization arises.

1. The DM Server issues a *TNDS* structural query on some nodes in the Management Tree, as described in [DMTND]. The DM Client responds by returning the entire structure of the DM subtree rooted at that node, in serialized format.
2. The DM Server invokes the *Add* command on some nodes in the Management Tree. The body of the *Add* command contains a serialized subtree [DMREPU]. The DM Client deserializes the subtree and attaches it to the target node.

   Note: In lieu of executing the *Add* command on some existing nodes in the Management Tree, the DM Server may invoke the *Add* command on the *Inbox* [DMSTDOBJ]. This causes the DM Client to choose the node to which the deserialized subtree is attached. This is the approach that is used for OTA Bootstrapping using the DM Profile, as described in [DMBOOT].

3. The DM Server invokes the *Replace* command on some nodes in the Management Tree. The body of the *Replace* command contains a serialized subtree [DMREPU]. The DM Client deserializes the subtree and replaces the existing subtree at the target node with the deserialized subtree.

5. TND Serialization Definition

5.1 TNDS XML usage

This specification defines how to transform between a management sub-tree and XML structure. The DTD used for XML structure to transform a management sub-treee is the same as DTD definition for DDF, which is defined in [DMTND]. A XML document complying with this specification has a similar but different content than the [DMTND] specification. Therefore a document complying with this specification MUST use a specific MIME-type to indicate it. It is also possible to use the WBXML encoding mechanism defined in [DMTND]. The two MIME-types, one for XML and one for WBXML are specified in Appendix C.

5.2 General

During runtime the client's management tree will contain interior nodes and leaf nodes. The client may take a snap-shot on the management tree or a subset of the management tree and convert all information of that part of the management tree into either an XML or a WBXML stream. Some fields are optional and some are mandatory to support. The description for all properties is defined in [DMTND]. Properties which are optional in the management tree may not be stored in the file.

The DTD in [DMTND] defines the format of the XML or WBXML file. This specification will not define any transport binding for the XML or WBXML byte stream.

The following figure shows an example of a management tree. With this specification it is possible for the device to translate the E-Mail Management Object to or from an XML or WBXML file.

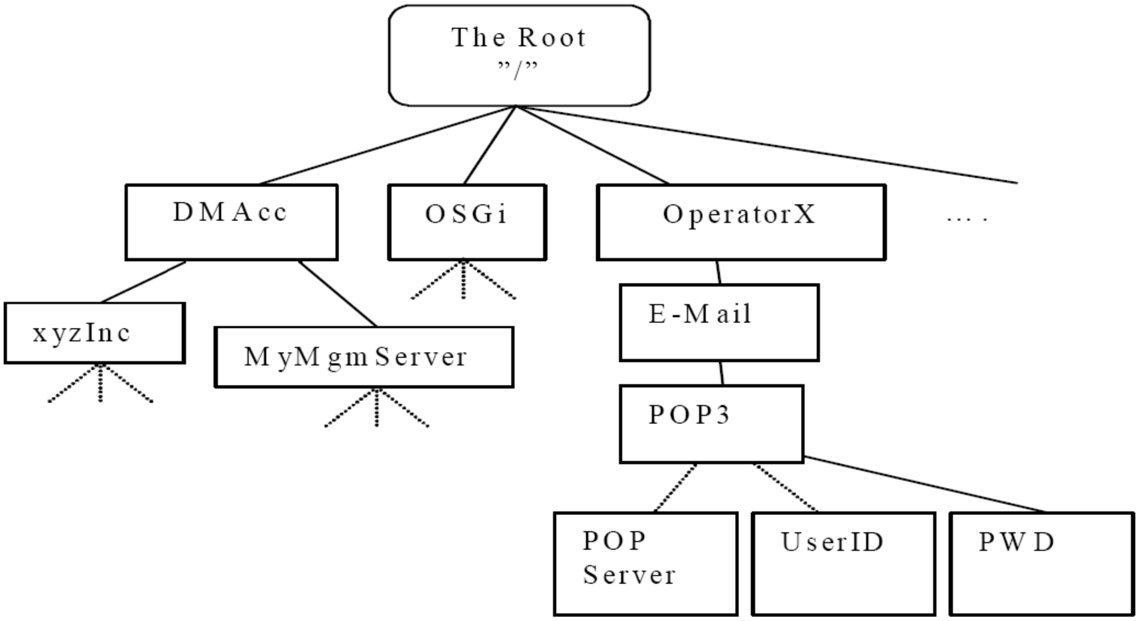

Figure 1: Example of management tree

5.3 DM Commands with TNDS data

The format of the TNDS object has its own MIME-type. Therefore if the device and server support this MIME-type it is possible to use DM commands and operate on a complete sub tree in one command. As per [DMREPPRO], TNDS data can only be used with the Add, Replace and Results Protocol Command Elements. Any attempt to invoke any other DM command with TNDS data MUST be rejected with the status code (405) Command not allowed.

The behaviour for the Add and Replace commands with TNDS data is described in [DMREPPRO].

A DM Server and DM Client MUST be able to encode and decode a TNDS object.

Note that TNDS object data MUST be escaped following CDATA escaping rule.

In the example given below, a DM Server uses a single Add command to instantiate an entire E-mail object in the Management Tree at the following location:

./OperatorX

```
<Add>
  <CmdID>4</CmdID>
  <Item>
    <Target>
       <LocURI>./OperatorX</LocURI>
    </Target>
    <Meta>
       <Format xmlns=’syncml:metinf’>xml</Format>
       <Type xmlns=’syncml:metinf’>
           application/vnd.syncml.dmtnds+xml
       </Type>
    </Meta>
    <Data>
          <![CDATA[
          <MgmtTree xmlns=’syncml:dmddf1.2’>
            <VerDTD>1.2</VerDTD>
            <Node>
              <NodeName>E-Mail</NodeName>
              <RTProperties>
                <Format>
                  <node/>
                </Format>
                <Type><DDFName>com.operatorX.dm/1.0/EMail</DDFName></Type>
              </RTProperties>
              <Node>
                <NodeName>POP3</NodeName>
                <Node>
                  <NodeName>POPServer</NodeName>
                  <RTProperties >
                    <Format>
                      <chr/>
                    </Format>
                    <Type ><MIME>text/plain</MIME></Type>
                  </RTProperties >
                  <Value >mail.Operatorx.com</Value>
                </Node>
                <Node>
                  <NodeName>UserID</NodeName>
                  <RTProperties>
                    <Format>
                      <chr/>
                    </Format>
                    <Type><MIME>text/plain</MIME></Type>
                  </RTProperties>
                  <Value>UserName</Value>
                </Node>
                <Node>
                  <NodeName>PWD</NodeName>
                  <RTProperties>
                    <Format>
                      <chr/>
                    </Format>
                    <Type><MIME>text/plain</MIME></Type>
                  </RTProperties>
                  <Value>4571F7C34A9876B3</Value>
                </Node>
              </Node>
            </Node>
          </MgmtTree>
          ]]>
    </Data>
  </Item>
</Add>
```

6. TNDS Syntax

6.1 Supported properties in file

All properties are defined and described as part of DDF definition in [DMTND]. Only a subset of these properties are valid in the file. The following table defines the subset of valid properties together with the requirements level for how it must be supported depending on if an entity is decoding or encoding a TNDS object:

Property	Decode	Encode
MgmtTree	MUST	MUST
VerDTD	MUST	MUST
Man	MAY	MAY
Mod	MAY	MAY
Node	MUST	MUST
NodeName	MUST	MUST
Path	MUST	MAY
Value	MUST	MUST
RTProperties	MUST	MUST
ACL	MUST	MAY
Format	MUST	MAY
Title	MAY	MAY
TStamp	MAY	MAY
Type	MUST	MUST
VerNo	MAY	MAY

Table 1: Supported properties in the TNDS object byte stream

6.2 TNDS DTD

The DTD is specified in [DMTND]. All extra behaviour and restriction compared to the DTD definitions is defined later in this chapter. A compliant TNDS object MUST follow all rules in this specification. The TNDS object byte stream can be encoded as XML or as WBXML. Two different MIME-types are defined to indicate what encoding type the byte stream uses.

6.3 TNDS Elements

This section defines the elements that are allowed in the TNDS object.

6.3.1 Structural elements

These elements provide various kinds of structural information of the described management object.

6.3.1.1 MgmtTree

Usage: Container for one or more described management objects.

Parent Elements: none

Restrictions: This MUST be the root element of all descriptions.

6.3.1.2 VerDTD

Usage: Specifies the major and minor version identifier of the DM Description Framework specification used to represent the DM description.

Parent Elements: MgmtTree

Restrictions: Major revisions of the specification create incompatible changes which will generally require a new parser. Minor revisions involve changes that do not impact basic compatibility of the parser. When the XML document conforms to this revision of the DM Tree and Description specification [DMTND] the value MUST be 1.2. The element type MUST be included in the MgmtTree.

6.3.1.3 Man

Usage: Specifies the device manufacturer name of the device the TNDS object is created from.

Parent Elements: MgmtTree

Restrictions: This element is OPTIONAL.

6.3.1.4 Mod

Usage: Specifies the device manufacturer’s model number of the device the TNDS object is created from.

Parent Elements: MgmtTree

Restrictions: This element is OPTIONAL.

6.3.1.5 Node

Usage: Specifies a node.

Parent Elements: MgmtTree

Restrictions: This element is recursive. A Node with a Value element MUST always terminate the recursion. It is possible for a Node to omit both the next recursive Node and a Value, this means that the hierarchy of Nodes continues elsewhere. This can be used to increase readability of very deep trees. In the continuation, the Path element MUST contain a full URI that specifies the insertion point in the tree.

Example: The following XML is a description of a number of nodes that form the URI Vendor/ISP/GWInfo/GWName.

```
<MgmtTree>
  <Node>
    <NodeName>Vendor</NodeName>
    <RTProperties>…</RTProperties>
    <Node>
      <NodeName>ISP</NodeName>
      <RTProperties>…</RTProperties>
      <Node>
        <NodeName>GWInfo</NodeName>
        <RTProperties>…</RTProperties>
        <Node>
          <NodeName>GWName</NodeName>
          <RTProperties>…</RTProperties>
          <Value>gw.yyy.se</Value>
        </Node>
      </Node>
    </Node>
  </Node>
</MgmtTree>
```

6.3.1.6 NodeName

Usage: Specifies the name of the described node.

Parent Elements: Node

Restrictions: See [RFC2396] for general restrictions on URI. The NodeName element MAY be empty. If empty, this means that the name of the node MUST be assigned when the node is created. When the node name is assigned at node creation time, the value for the name is set to the last segment of the URI specified as Target for the command this results in the node being created. See also in [DMTND].

If NodeName is omitted then the client may use the Type to decide where in the tree the structure should be created.

6.3.1.7 Path

Usage: Specifies the URI up to, but not including, the described node.

Parent Elements: Node

Restrictions: OPTIONAL element when creating a TNDS object and MANDATORY when importing nodes from a TNDS object. If omitted, the URI for the node MUST be constructed by concatenating all ancestral NodeName and Path values. This concatenated value MUST form the correct URI. Path SHOULD only be used inside Node elements that are child elements to the MgmtTree element. For general restrictions, see [RFC2396].

Example: The following XML is an alternative way to describe the same management objects as in the example in section 6.3.1.5. This description specifies the same URI as the other example: Vendor/ISP/GWInfo/GWName.

```
<MgmtTree>
  <Node>
    <NodeName>Vendor</NodeName>
    <RTProperties>…</RTProperties>
  </Node>
  <Node>
    <NodeName>ISP</NodeName>
    <Path>Vendor</Path>
    <RTProperties>…</RTProperties>
  </Node>
  <Node>
    <NodeName>GWInfo</NodeName>
    <Path>Vendor/ISP</Path>
    <RTProperties>…</RTProperties>
  </Node>
  <Node>
    <NodeName>GWName</NodeName>
    <Path>Vendor/ISP/GWInfo</Path>
    <RTProperties>…</RTProperties>
    <Value>gw.yyy.se</Value>
  </Node>
</MgmtTree>
```

6.3.1.8 Value

Usage: Specifies the value for nodes that are instantiated using the current description.

Parent Elements: Node

Restrictions: OPTIONAL element. If omitted, the node description does not specify any default value for the node. In this case the initial value of new nodes is undefined.

6.3.1.9 RTProperties

Usage: Aggregating element for run-time properties, i.e. properties that the nodes have in a device at run-time. Used to specify which properties the described node supports at run-time. Can also be used to supply default values for supported run-time properties.

Parent Elements: Node

Restrictions: MANDATORY element.

6.3.1.10 ACL

Usage: Specifies the ACL value. This MAY be used to specify a different ACL value then the default value for this node.

Parent Elements: RTProperties

Restrictions: If a value is specified it MUST be formatted according to definition in [DMTND].

6.3.1.11 Format

Usage: Specifies the Format value.

Parent Elements: RTProperties

Restrictions: If a value is specified it MUST specify the correct format of the node according to section definition in [DMTND].

6.3.1.12 TStamp

Usage: Specifies the TStamp value.

Parent Elements: RTProperties

Restrictions: If a value is specified it MUST be formatted according to section definition in [DMTND].

6.3.1.13 Type

Usage: Specifies the Type value.

Parent Elements: RTProperties

Restrictions: For leaf nodes, the Type property MUST be used to specify the correct MIME type of the node value. For interior nodes the value MUST specify a valid identifier of a DDF document, or be empty according to section definition in [DMTND].

6.3.1.14 VerNo

Usage: Specifies the VerNo value.

Parent Elements: RTProperties

Restrictions: If a value is specified it MUST be formatted according to definition in [DMTND].

6.3.1.15 b64

Usage: OMA DM format description. Specifies that the node value is Base64 encoded.

Parent Elements: Format

Restrictions: None.

6.3.1.16 bin

Usage: OMA DM format description. Specifies that the node value is binary data.

Parent Elements: Format

Restrictions: None.

6.3.1.17 bool

Usage: OMA DM format description. Specifies that the node value is a Boolean.

Parent Elements: Format

Restrictions: None.

6.3.1.18 chr

Usage: OMA DM format description. Specifies that the node value is text.

Parent Elements: Format

Restrictions: The character set used is specified either by the transport protocol, MIME content type header or XML prologue.

6.3.1.19 int

Usage: OMA DM format description. Specifies that the node value is an integer.

Parent Elements: Format

Restrictions: None.

6.3.1.20 node

Usage: OMA DM format description. Specifies that the node is an interior node.

Parent Elements: Format

Restrictions: This Format MUST only be used for interior nodes.

6.3.1.21 null

Usage: OMA DM format description. Specifies that the node value is null.

Parent Elements: Format

Restrictions: None.

6.3.1.22 xml

Usage: OMA DM format description. Specifies that the node value is XML data.

Parent Elements: Format

Restrictions: None.

6.3.1.23 date

Usage: OMA DM format description. Specifies that the node value is a date in ISO 8601 format with the century being included in the year [ISO8601].

Parent Elements: Format

Restrictions: None.

6.3.1.24 time

Usage: OMA DM format description. Specifies that the node value is a time in ISO 8601 format [ISO8601].

Parent Elements: Format

Restrictions: None.

6.3.1.25 float

Usage: OMA DM format description. Specifies that the node value is a real number corresponding to a single precision 32 bit floating point type as defined in XML Schema 1.0 as the float primitive type [XMLSCHEMADT].

Parent Elements: Format

Restrictions: None.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-DM_TNDS-V1_2-20070209-A	09 Feb 2007	Status changed to Approved by TP TP Ref # OMA-TP-2007-0075R03-INP_ERP_DM_V1.2_for_Final_Approval
OMA-TS-DM_TNDS-V1_3-20160524-A	24 May 2016	Status changed to Approved by TP TP Ref # OMA-TP-2016-0041R01-INP_DM_V1_3_ERP_for_final_Approval

Appendix B. Static Conformance Requirements (Normative)

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

B.1 SCR for DM Client

Item	Function	Reference	Requirement
DM-TNDS-C-001-M	Support of Encoding a TNDS object	Section 5.3	
DM-TNDS-C-002-M	Support of Decoding a TNDS object	Section 5.3	

B.2 SCR for DM Server

Item	Function	Reference	Requirement
DM-TNDS-S-001-M	Support of Encoding a TNDS object	Section 5.3	
DM-TNDS-S-002-M	Support of Decoding a TNDS object	Section 5.3	

Appendix C. Type definitions (Informative)

C.1 MIME Media Type Definition

MIME Type	Description
application/vnd.syncml.dmtnds+xml	XML encoded Serialized MO that comply with this specification
application/vnd.syncml.dmtnds+wbxml	WBXML encoded Serialized MO that comply with this specification and WBXML encoded as specified in [DMTND]