OMA Device Management Representation Protocol

Approved Version 1.3 – 24 May 2016

Open Mobile Alliance

OMA-TS-DM_RepPro-V1_3-20160524-A

Contents

Figures

No table of figures entries found.

Tables

No table of figures entries found.

1. Scope

This document describes the DM Representation Protocol and its usage.

2. References

2.1 Normative References

[3GPP-TS_23.003] 3GPP TS 23.003 "Numbering, addressing and identification"

[DMDICT] "OMA Device Management Dictionary, Version 1.0". Open Mobile Alliance™. OMA-SUP-DM_Dictionary-v1_0. 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

[DMREPDTD] "OMA Device Management Representation Protocol DTD, Version 1.2". Open Mobile Alliance™. URL:http://www.openmobilealliance.org/tech/dtd/DM_RepPro-v1_2.dtd

[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 Specification, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_TNDS-V1_3. URL:http://www.openmobilealliance.org/

[META] "SyncML Meta Information Specification, version 1.3. Open Mobile Alliance™. OMA-TS-SyncML_MetaInfo-V1_3. URL:http://www.openmobilealliance.org/

[RFC2045] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed & N. Borenstein, November 1996, URL:http://www.ietf.org/rfc/rfc2045.txtURL:http://www.ietf.org/rfc/rfc2045.txt

[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

[RFC4122] "A Universally Unique Identifier (UUID) URN Namespace", P. Leach, et al. July 2005, URL:http://www.ietf.org/rfc/rfc4122.txt

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

[WBXML1.1] "WAP Binary XML Content Format Specification", WAP Forum™, SPEC-WBXML-19990616.pdf, URL:http://www.openmobilealliance.org

[WBXML1.2] "WAP Binary XML Content Format Specification", WAP Forum™, WAP-154-WBXML, URL:http://www.openmobilealliance.org

[WBXML1.3] "WAP Binary XML Content Format Specification", WAP Forum™, WAP-192-WBXML, URL:http://www.openmobilealliance.org

[XML] "Extensible Markup Language (XML) 1.0", World Wide Web Consortium Recommendation, URL:http://www.w3.org/TR/REC-xmlURL:http://www.w3.org/TR/REC-xml

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.

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 document specifies the DM representation syntax and semantics used for device management.

The DM representation protocol is defined by a set of messages that are conveyed between entities participating in a DM operation. The messages are represented as an XML document. XML is the industry standard for text document mark-up, as defined in [XML].

The DM representation protocol also can be identified as a MIME content type. MIME is the Internet standard for identifying multipurpose message contents. It provides a useful mechanism for differentiating between different content and document types.

The DM representation protocol supports protocol models that are based on a request/response command structure, as well as those that are based on a "blind push" command structure.

The DM representation protocol embodies the concept of a DM Package. The DM Package performs some set of operations. This conceptual "package" permits either a "batch" of multiple operations put together in a single DM Message or conveyed as separate DM Messages, each containing a single operation. DM Messages are the body of the MIME entities.

5. DM Representation

5.1 DM Package and Messages

In DM, the operations are conceptually bound into a DM Package. The DM Package is just a conceptual frame for one or more DM Messages that are REQUIRED to convey a set of protocol semantics.

A DM Message is a well-formed XML document and adheres to the DTD, but does not need to be validated. For example, a DM message does not need to be validated but the XML MUST adhere to whatever explicitly defined order appears in the DTD. The document is identified by the DM representation root or document element type. This element type acts as a parent container (i.e., root element type) for the DM Message.

The DM Message, as specified before, is an individual XML document. The document consists of a header, specified by the SyncHdr element type, and a body, specified by the SyncBody element type. The DM representation header specifies routing and versioning information about the DM Message. The DM representation body is a container for one or more DM Commands. The DM Commands are specified by individual element types. The DM Commands act as containers for other element types that describe the specifics of the DM command, including any data or meta-information.

5.2 DM Commands

DM Representation defines the following "request" commands:

- Add. Allows the originator to ask that a data element or data elements supplied by the originator be added to data accessible to the recipient.
- Alert. Allows the originator to notify the recipient. The notification can be used as an application-to-application message or a message intended for display through the recipient's user interface.
- Atomic. Allows the originator to indicate that a set of commands to be performed with all or nothing semantics.
- Copy. Allows the originator to ask that a data element or data elements accessible to the recipient be copied.
- Delete. Allows the originator to ask that a data element or data elements accessible to the recipient be deleted. A Delete command can include a request for the archiving of the data.
- Exec. Allows the originator to ask that a named or supplied executable is invoked by the recipient.
- Get. Allows the originator to ask for a data element or data elements from the recipient. A get can include the resetting of any meta-information that the recipient maintains about the data element or collection.
- Map. Defined in [DMREPDTD] and is not used by DM.
- Move. Defined in [DMREPDTD] and is not used by DM.
- Put. Defined in [DMREPDTD] and is not used by DM.
- Replace. Allows the originator to ask that a data element or data elements accessible to the recipient be replaced. This command makes a complete replacement of the data element.
- Search. Defined in [DMREPDTD] and is not used by DM.
- Sequence. Allows the originator to indicate that a set of commands is to be performed in the specified sequence.
- Sync. Defined in [DMREPDTD] and is not used by DM.

DM Representation defines the following "response" commands:

- Status. Indicates the completion status of an operation or that an error occurred while processing a previous request.
- Results. Used to return the data results of a Get Command.

5.3 XML Usage

The DM Messages are represented in a mark-up language defined by [XML]. The DM representation protocol is an XML application. The DM DTD (Document Type Definition) defines the XML document type used to represent a DM Message. The DM DTD can be found in Section 8, but it is not necessary to read the DTD in order to understand the protocol.

DM Messages are specified using well-formed XML. However, the DM Messages need not be valid XML. That is, the DM Messages do not need to specify the XML declaration or prolog. They only need to specify the body of the XML document. This restriction allows for the DM Messages to be specified with greater terseness than well-formed, valid XML documents.

DM makes heavy use of XML name spaces. Name spaces MUST be declared on the first element type that uses an element type from the name space.

Names in XML are case sensitive. By convention in the DM DTD, the element type and attribute list names are specified using the convention that the first character in each word of the name is in upper case text and remainder of the characters in each word of the names specified in lower case text. For example, SyncML for the DM representation Language tag or MsgRef for the Message Reference tag.

DM representation also makes use of XML standard attributes, such as xml:lang. Any XML standard attribute can be used in a DM document.

XML can be viewed as more verbose than alternative binary representations. This is often cited as a reason why it might not be appropriate for low bandwidth network protocols. In most cases, DM representation uses shortened element type and attributes. This provides a minor reduction in verbosity. Additionally, the DM Messages can be encoded in the WBXML tokenized, binary format as defined in section 6.2. The use of this format is external to specification of the DM protocol and transparent to any DM application. The combination of the use of shortened element type names and an alternative binary format makes DM competitive, from a compressed format perspective, with alternative, but private, binary representations.

5.4 MIME Usage

The [RFC2045] Internet standard provides an industry-accepted mechanism for identifying different content types. The DM Message is identified by a MIME media type. The media type for the DM Message is registered within the vendor tree. The MIME content types for SyncML Device Management are specified in section 6.1 of this document. One of these MIME content types MUST be used for identifying DM Messages within transport and session level protocols that support MIME content types.

5.5 Identifiers

Identifiers in DM message, such as in the Source or Target element types, can be a combination of Uniform Resource Identifiers (URI), as defined by [RFC2396], Uniform Resource Names (URN) and textual names.

In DM message, all URI and URN values are specified as parsable character data in element types or as character data in attribute lists. Applications MUST specify a valid URI or URN value. Even with an integrated "validating XML parser", as defined in [XML], an application will need to confirm the validity of any URI or URN.

DM uses the SYNCML URN type to identify DM specific name spaces and unique names. Other URN types MAY be used. For instance, the LocURI element type could contain one of the following URN:

IMEI URN	Identify an International Mobile Equipment Identifiers [3GPP-TS_23.003]. The IMEI URN specifies a valid, 15 digit IMEI. The format of the URN is IMEI: ###############
ESN URN	Identify an Electronic Serial Number. The ESN specifies a valid, 8 digit ESN. The format of the URN is ESN: ########
MEID URN	Identify a Mobile Equipment Identity. The MEID URN specifies a valid, 15 digit MEID. The format of the URN is MEID: ###############
UUID URN	Identify an Universally Unique IDentifier (UUID). The UUID specifies a valid, hex digit character string as defined in [RFC4122]. The format of the URN is UUID:########-####-####-############

Other URN types MAY be used in the LocURI element type also.

6. OMA Device Management Usage

6.1 MIME Usage

There are two MIME content types for the OMA Device Management Message. The MIME content type of application/vnd.syncml.dm+xml identifies the clear-text XML representation for the DM Message. The MIME content type of application/vnd.syncml.dm+wbxml identifies the WBXML binary representation for the DM Message. Appendix C of this specification specifies the MIME content type registration for these two MIME media types.

One of these two MIME content types MUST be used for identifying OMA Device Management Messages within transport and session level protocols that support MIME content types.

6.2 WBXML Usage

All clients and servers MUST expect any 1.x version of WBXML, and all clients and servers MUST use any of the following versions of WBXML [WBXML1.1], or [WBXML1.2], or [WBXML1.3].

7. Mark-up Language Description

Examples in this section make use of XML snippets. They are not intended to be complete XML documents. They are only provided to illustrate an example usage of the element type in question.

7.1 Common Use Elements

The following are common element types used by numerous other element types. The table lists the mandatory and optional elements that servers and clients send and receive.

Command	Support of Management Server		Support of Management Client	
	Sending	Receiving	Sending	Receiving
Chal	MUST	MUST	MUST	MUST
Cmd	MUST	MUST	MUST	MUST
CmdID	MUST	MUST	MUST	MUST
CmdRef	MUST	MUST	MUST	MUST
Cred	MUST	MUST	MUST	MUST
Final	MUST	MUST	MUST	MUST
LocName	MUST	MUST	MUST	MUST
LocURI	MUST	MUST	MUST	MUST
MoreData	MUST	MUST	SHOULD	SHOULD
MsgID	MUST	MUST	MUST	MUST
MsgRef	MUST	MUST	MUST	MUST
RespURI	MAY	MUST	MAY	MUST
SessionID	MUST	MUST	MUST	MUST
Source	MUST	MUST	MUST	MUST
SourceRef	MUST	MUST	MUST	MUST
Target	MUST	MUST	MUST	MUST
TargetRef	MUST	MUST	MUST	MUST
VerDTD	MUST	MUST	MUST	MUST
VerProto	MUST	MUST	MUST	MUST

7.1.1 Archive

Restrictions: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.2 Chal

Usage: Specifies an authentication challenge. The receiver of the challenge specifies authentication credentials, of the given authentication type and format, in the next request.

Parent Elements: Status

Restrictions: The Meta element type specifies any meta-information about the challenge. The Type and Format element types within the Meta element type specify the authentication scheme type and format, respectively. The default type is syncml:auth-basic for the "Basic" form of authentication. The type value syncml:auth-md5, or syncml:auth-sha256 MUST be explicitly specified to indicate the "MD5 Digest Access", or “SHA256 Digest Access” authentication scheme. If the "MD5 Digest Access" or “SHA256 Digest Access” authentication scheme is used, the NextNonce element type can be specified if the challenger requests the use of a new nonce string. The format value MUST be b64, when using the clear-text, XML representation.

An authentication challenge can be specified for each of a number of DM "security layers". For example, a challenge can be specified against the DM server, database or an individual command on a database. To challenge a DM Server, a Chal element type is sent in the Status command corresponding to the SyncHdr of the associated DM request. To challenge a database, the Chal element type is sent in the Status command corresponding to the Alert or Sync command associated with the database. To challenge a command on a database, the Chal element type is sent in the Status command corresponding to an individual command (e.g., Add, Alert, Delete) on the database. Mechanisms for authentication challenges at the transport level are handled within the individual transport.

If absent and if the status code is (200) OK, then the same credentials MUST be used in the next DM request.

If absent and if the status code is (212) Authentication accepted, then credentials need not be specified for any subsequent DM requests within the current session. The session is authenticated.

When using syncml:auth-md5, syncmd:auth-sha256 or syncml:auth-MAC, the Meta Format for the NextNonce element MUST be specified and it MUST be b64.

Content Model:

```
(Meta)
```

Attributes: None.

Example: The following is an example of a "MD-5" authentication challenge. The password and userid are requested to be Base64 character encoded. The type and format of the authentication scheme are specified by the meta-information in the Meta element type.

```
<Status>
  <MsgRef>0</MsgRef>
  <Cmd>SyncHdr</Cmd>
  <TargetRef>http://www.datamgr.org/servlet/manageit</TargetRef>
  <SourceRef>IMEI:001004FF1234567</SourceRef>
  <Chal>
    <Meta>
      <Type xmlns='syncml:metinf'>syncml:auth-md5</Type>
      <Format xmlns='syncml:metinf'>b64</Format>
      <NextNonce xmlns='syncml:metinf'>ZG9iZWhhdmUNCg==</NextNonce>
    </Meta>
  </Chal>
  <Data>401</Data>
</Status>
```

7.1.3 Cmd

Usage: Specifies the name of the DM command referenced by a Status element type.

Parent Elements: Status

Restrictions: The value MUST be one of Add, Alert**,** Atomic, Copy, Delete, Exec, Get, Map, Move, Put, Replace, Results, Search**,** Sequence, Status**,** Sync.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Status>
  <MsgRef>1</MsgRef>
  <CmdRef>2</CmdRef>
  <CmdID>1234</CmdID>
  <Cmd>Replace</Cmd>
  <TargetRef>./antivirus_data</TargetRef>
  <!-- OK, antivirus update loaded-->
  <Data>200</Data>
</Status>
```

7.1.4 CmdID

Usage: Specifies a DM message-unique command identifier.

Parent Elements: Add, Alert, Atomic, Copy, Delete, Exec, Get, Map, Move, Put, Replace, Results, Search, Sequence, Status, Sync

Restrictions: A text value that MUST be unique within the DM Message.

The element type MUST always be present and the value MUST NOT be the text string "0".

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Status>
  <MsgRef>1</MsgRef>
  <CmdRef>2</CmdRef>
  <CmdID>1234</CmdID>
  <Cmd>Replace</Cmd>
  <TargetRef>./antivirus_data</TargetRef>
  <!-- OK, antivirus update loaded-->
  <Data>200</Data>
</Status>
```

7.1.5 CmdRef

Usage: Specifies the CmdID referenced by a Status element type.

Parent Elements: Results, Status

Restrictions: MUST refer to the identifier of the DM command reference by the Status element type.

The only instance where the element type can be absent in the Status command is the case where the Status command refers to the SyncHdr of the associated DM request message. For example, a status can be sent back to the originator for exceptions (e.g., (401) Unauthorized) found within the SyncHdr of the originator's request.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Status>
  <MsgRef>1</MsgRef>
  <CmdRef>2</CmdRef>
  <CmdID>1234</CmdID>
  <Cmd>Replace</Cmd>
  <TargetRef>./antivirus_data</TargetRef>
  <!-- OK, antivirus update loaded-->
  <Data>200</Data>
</Status>
```

7.1.6 Cred

Usage: Specifies an authentication credential for the originator.

Parent Elements: Add, Alert, Copy, Delete, Exec, Get, Put, Map, Move, Replace, Search, Status, Sync, SyncHdr

Restrictions: The Meta element type specifies any meta-information about the credentials. The Type and Format element types within the Meta element type specify the credential scheme type and format, respectively. The default type is syncml:auth-basic for the "Basic" form of authentication. The type value syncml:auth-md5 MUST be explicitly specified to indicate the "MD5 Digest" authentication scheme. The type value syncml:auth-sha256 MUST be explicitly specified to indicate the "SHA-256 Digest" authentication scheme. The format MUST be b64, when using the clear-text, XML representation. However, when using "Basic" form of authentication, the b64 format does not indicate that the credentials are base64 encoded twice. The Data element type specifies the credential value.

If absent, and no other authentication credential was specified in either a parent command or in the SyncHdr element type, then no authentication credential is specified.

If an authentication credential was specified by a parent command or in the SyncHdr element type, then that authentication credential specified there is assumed to be sufficient for the operation specified by the current element type. Specifying insufficient authentication credentials will result in a (401) Unauthorized exception condition.

If the authentication challenge is received (See the Chal element type) for the request, the credential type and format of the next request MUST be applied to it.

In addition, OMA DM restricts the usage of the Cred element to within the sync header element: SyncHdr. The originator MUST NOT supply credentials within individual commands.
When using syncml:auth-md5, the Meta Format for the Cred element MUST be specified and it MUST be b64

Content Model:

```
(Meta?, Data)
```

Attributes: None.

Example: The following is an example of an MD5 digest authentication credential scheme consisting of the character string Bruce2:OhBehave:Nonce. The MD5 Digest is also Base64 character encoded. The type and format of the credential, as well as the next nonce are specified by the meta-information in the Meta element type.

```
<Cred>
  <Meta>
    <Type xmlns='syncml:metinf'>syncml:auth-md5</Type>
    <Format xmlns='syncml:metinf'>b64</Format>
  </Meta>
  <Data>Zz6EivR3yeaaENcRN6lpAQ==</Data>
</Cred>
```

7.1.7 Field

Restrictions by DM: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.8 Filter

Restrictions by DM: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.9 FilterType

Restrictions by DM: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.10 Final

Usage: Indicator that the DM message is the last message in the current DM package.

Parent Elements: SyncBody

Restrictions: The element type MUST only be specified on the last message of the DM package. If not present, then more messages follow this DM message in the current DM package.

Content Model:

```
(EMPTY)
```

Attributes: None.

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>...blah, blah...</SyncHdr>
  </SyncBody>
  ...blah, blah...
  <Final/>
  </SyncBody>
</SyncML>
```

7.1.11 Lang

Restrictions: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.12 LocName

Usage: Specifies the display name for the target or source address.

Parent Elements: Target, Source

Restrictions: Used for sending userid for MD5 authentication.

Content Model:

```
(#PCDATA)
```

Attributes: None.

7.1.13 LocURI

Usage: Specifies the target or source specific address.

Parent Elements: Target, Source, SourceParent, TargetParent

Restrictions: MUST be either an absolute or a relative URI or a well-known URN. For instance, the LocURI element type could contain one of the following URN:

- IMEI URN: Identify an International Mobile Equipment Identifiers [3GPP-TS_23.003]. The IMEI URN specifies a valid, 15 digit IMEI. The format of the URN is IMEI:###############
- ESN URN: Identify an Electronic Serial Number. The ESN specifies a valid, 8 digit ESN. The format of the URN is ESN:########
- MEID URN: Identify a Mobile Equipment Identifier. The MEID URN specifies a valid, 14 digit MEID. The format of the URN is MEID:##############
- UUID URN: Identify an Universally Unique IDentifier (UUID). The UUID specifies a valid, hex digit character string as defined in [RFC4122]. The format of the URN is UUID:########-####-####-############

Other URN types MAY be used in the LocURI element type also.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<SyncHdr>
  <VerDTD>1.2</VerDTD>
  <VerProto>DM/1.3</VerProto>
  <SessionID>1</SessionID>
  <MsgID>1</MsgID>
  <Target>
    <LocURI>http://www.syncml.org/mgmt-server</LocURI>
  </Target>
  <Source>
    <LocURI>IMEI:493005100592800</LocURI>
  </Source>
</SyncHdr>
```

7.1.14 MoreData

Usage: Indicator that a data element is incomplete and there will be one or more subsequent chunks.

Parent Elements: Item

Restrictions: The element type MUST be specified on all but the last chunk of data of an item. If not present, then the item is either contained within a single message or is the closing chunk of the data item.

Content Model:

```
(EMPTY)
```

Attributes: None.

Example:

```
<Add>
  <CmdID>15</CmdID>
  <Meta>
    <Type xmlns='syncml:metinf'>bin</Type>
    <Format xmlns='syncml:metinf'>b64</Format>
    <Size xmlns='syncml:metinf'>3000</Size>
  </Meta>
  <Item>
    <Target>
      <LocURI>./</LocURI>
    </Target>
    <Data>
      <!-- First chunk of data file -->
    </Data>
    <MoreData/>
  </Item>
</Add>
```

7.1.15 MsgID

Usage: Specifies a DM session-unique identifier for the DM Message.

Parent Elements: SyncHdr

Restrictions: The message identifier MUST be unique to the device within the DM session. The element type MUST be specified in the SyncHdr. The value is a monotonically increasing numeric value starting at one (1) for the first message in the DM session. The message identifier specified in a DM request MUST be the content of the MsgRef element type in the corresponding DM results or response status.

Content Model:

```
(#PCDATA)
```

Attributes: None

Example:

```
<SyncHdr>
  <VerDTD>1.2</VerDTD>
  <VerProto>DM/1.3</VerProto>
  <SessionID>1</SessionID>
  <MsgID>1</MsgID>
  <Target>
    <LocURI>http://www.syncml.org/mgmt-server</LocURI>
  </Target>
  <Source>
    <LocURI>IMEI:493005100592800</LocURI>
  </Source>
</SyncHdr>
```

7.1.16 MsgRef

Usage: Specifies a reference to a DM session-unique identifier referenced by a DM results or response status.

Parent Elements: Results, Status

Restrictions: The value MUST reference the message identifier of the DM message referred to by the results or response status.

Content Model:

```
(#PCDATA)
```

Attributes: None

Example:

```
<Status>
  <MsgRef>1</MsgRef>
  <CmdRef>2</CmdRef>
  <CmdID>1234</CmdID>
  <Cmd>Replace</Cmd>
  <TargetRef>./antivirus_data</TargetRef>
  <!-- OK, antivirus update loaded-->
  <Data>200</Data>
</Status>
```

7.1.17 NoResp

Restrictions: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.18 NoResults

Restrictions: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.19 NumberOfChanges

Restrictions: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.20 Record

Restrictions by DM: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.21 RespURI

Usage: Specifies the URI that the recipient MUST use for any response to this message.

Parent Elements: SyncHdr

Restrictions: The value of this element is the address, in the form of an absolute URI that the recipient MUST use for any response to this message. If the Source is not the same as this value, then the Source element MUST also be specified in the SyncHdr element type. Note that the server and databases are the same entities at this new address. Receipt of this command does not mean you SHOULD repeat commands in the previous message.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<SyncHdr>
  <VerDTD>1.2</VerDTD>
  <VerProto>DM/1.3</VerProto>
  <SessionID>1</SessionID>
  <MsgID>1</MsgID>
  <Target>
    <LocURI>http://www.syncml.org/mgmt-server</LocURI>
  </Target>
  <Source>
    <LocURI>IMEI:493005100592800</LocURI>
  </Source>
  <RespURI>http://www.deviceman.org/servlet/manageit/bruce1?user=jsmith&af
ter=20000512T133000Z</RespURI>
</SyncHdr>
```

7.1.22 SessionID

Usage: Specifies the identifier of the DM session associated with the DM Message.

Parent Elements: SyncHdr

Restrictions: The value is an opaque string. The element type MUST be specified in the SyncHdr element type in all DM Messages. The initiator SHOULD use unique SessionID for each session.

The maximum length of a SessionID is 2 bytes. Therefore the maximum SessionID is “FFFF”.

Content Model:

```
(#PCDATA)
```

Attributes: None

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.2' >
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.3</VerProto>
    <SessionID>2E2</SessionID>
    <MsgID>1</MsgID>
    <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
    </Target>
    <Source>
      <LocURI>IMEI:493005100592800</LocURI>
    </Source>
  </SyncHdr>
  <SyncBody>
    ...blah, blah...
  </SyncBody>
</SyncML>
```

7.1.23 SftDel

Restrictions: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.24 Source

Usage: Specifies source routing or mapping information.

Parent Elements: Item, Map, MapItem, Search, Sync, SyncHdr,

Restrictions: When specified in the Item element type, the Source element type specifies the database item that is the source of the DM command.

When specified in the SyncHdr element type, the Source element type specifies the source routing information for the network device that originated the DM Message.

If the RespURI element type is also specified within the SyncHdr, then the Source element type specifies the source routing information for a proxy originator of the DM message.

Content Model:

```
(LocURI, LocName?)
```

Attributes: None.

Example: The following is an example of the usage in a `SyncHdr` element type.

```
<SyncHdr>
  <VerDTD>1.2</VerDTD>
  <VerProto>DM/1.3</VerProto>
  <SessionID>1</SessionID>
  <MsgID>1</MsgID>
  <Target>
    <LocURI>http://www.syncml.org/mgmt-server</LocURI>
  </Target>
  <Source>
    <LocURI>IMEI:493005100592800</LocURI>
  </Source>
</SyncHdr>
```

7.1.25 SourceParent

Restrictions by DM: This element is defined in [DMREPDTD], but this is not used in OMA Device Management Protocol.

7.1.26 SourceRef

Usage: Specifies the Source referenced by a Status or Results element type

Parent Elements: Status, Results

Restrictions: When specified in the Status element type, specifies the source address specified in the command associated with the response status. When specified in the Results element type, specifies the source address specified in the associated Search or Get command.

The element type MUST be specified in a Status command corresponding to any DM command that includes the Source element type.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Status>
  <CmdID>4321</CmdID>
  <MsgRef>1</MsgRef>
  <CmdRef>1234</CmdRef>
  <Cmd>Copy</Cmd>
  <TargetRef>./DM/WAPSetting/1</TargetRef>
  <SourceRef>./Common/WAP/1</SourceRef>
  <Data>200</Data>
</Status>
```

7.1.27 Target

Usage: Specifies target routing or mapping information.

Parent Elements: Item, Map, MapItem, Search, Sync, SyncHdr,

Restrictions: When specified in the Item element type, the Target element type specifies the database item that is the target of the DM command.

When specified in the SyncHdr element type, the Target element type specifies the target routing information for the network device that is receiving the DM Message.

The Filter element type can only be specified when the Target Item element type is specified within an Alert element.

Content Model:

```
(LocURI, LocName?, Filter?)
```

Attributes: None.

Example: The following is an example of the usage in a SyncHdr element type.

```
<SyncHdr>
  <VerDTD>1.2</VerDTD>
  <VerProto>DM/1.3</VerProto>
  <SessionID>1</SessionID>
  <MsgID>1</MsgID>
  <Target>
    <LocURI>http://www.syncml.org/mgmt-server</LocURI>
  </Target>
  <Source>
    <LocURI>IMEI:493005100592800</LocURI>
  </Source>
</SyncHdr>
```

7.1.28 TargetParent

Usage: Specifies the parent information of the item that is specified by Target LocURI.

Parent Elements: Item

Content Model:

```
(LocURI)
```

Attributes: None

7.1.29 TargetRef

Usage: Specifies the Target referenced by a Status or Results element type

Parent Elements: Status, Results

Restrictions: When specified in the Status element type, specifies the target address used in the command associated with the response status. When specified in the Results element type, specifies the target address specified in the associated Search or Get command.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Status>
  <CmdID>4321</CmdID>
  <MsgRef>1</MsgRef>
  <CmdRef>1234</CmdRef>
  <Cmd>Copy</Cmd>
  <TargetRef>./DM/WAPSetting/1</TargetRef>
  <SourceRef>./Common/WAP/1</SourceRef>
  <Data>200</Data>
</Status>
```

7.1.30 VerDTD

Usage: Specifies the major and minor version identifier of the DM representation protocol specification used to represent the DM message.

Parent Elements: SyncHdr

Restrictions: Major revisions of the specification create incompatible changes that will generally require a new DM message 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 representation protocol specification the value MUST be 1.2. The element type MUST be included in the SyncHdr.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<SyncHdr>
  <VerDTD>1.2</VerDTD>
  <VerProto>DM/1.3</VerProto>
  <SessionID>1</SessionID>
  <MsgID>1</MsgID>
  <Target>
    <LocURI>http://www.syncml.org/mgmt-server</LocURI>
  </Target>
  <Source>
    <LocURI>IMEI:493005100592800</LocURI>
  </Source>
</SyncHdr>
```

7.1.31 VerProto

Usage: Specifies the version identifier of the DM protocol specification used with the DM Message.

Parent Elements: SyncHdr

Restrictions: The first DM Message in each DM Package sent from an originator to a recipient MUST include the VerProto element type in the SyncHdr.

Major revisions of the specification create incompatible changes that may require a new management client. Minor revisions involve changes that do not impact basic compatibility of existing management clients.

When the DM message conforms to this revision of the OMA Device Management protocol specification the value MUST be 'DM/1.3'.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<SyncHdr>
  <VerDTD>1.2</VerDTD>
  <VerProto>DM/1.3</VerProto>
  <SessionID>1</SessionID>
  <MsgID>1</MsgID>
  <Target>
    <LocURI>http://www.syncml.org/mgmt-server</LocURI>
  </Target>
  <Source>
    <LocURI>IMEI:493005100592800</LocURI>
  </Source>
</SyncHdr>
```

7.2 Message Container Elements

The following element types provide the basic container support for the DM message.

Command	Support of Management Server		Support of Management Client	
	Sending	Receiving	Sending	Receiving
SyncML	MUST	MUST	MUST	MUST
SyncHdr	MUST	MUST	MUST	MUST
SyncBody	MUST	MUST	MUST	MUST

7.2.1 SyncML

Usage: Specifies the container for a DM Message.

Parent Elements: None. This is the root or document element.

Restrictions by DM: Within transports that support MIME content-type identification, this object MUST be identified as

application/vnd.syncml.dm+xml (for clear-text, XML representation) or application/vnd.syncml.dm+wbxml (for binary, WBXML representation).

Content Model:

```
(SyncHdr, SyncBody)
```

Attributes:

Name	Type	Occurrence	Description
xmlns	CDATA	REQUIRED	Value MUST be the text: ‘SYNCML:SYNCML1.2’

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.3</VerProto>
    <SessionID>1</SessionID>
      <MsgID>1</MsgID>
    <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
    </Target>
    <Source>
      <LocURI>IMEI:493005100592800</LocURI>
    </Source>
  </SyncHdr>
  <SyncBody>
    ...blah, blah...
  </SyncBody>
</SyncML>
```

7.2.2 SyncHdr

Usage: Specifies the container for the revisioning, routing information in the DM message.

Parent Elements: SyncML

Restrictions: The OPTIONAL Meta is used to convey meta-information about the DM messages, such as the maximum byte size of a DM response.

Content Model:

```
(VerDTD, VerProto, SessionID, MsgID, Target, Source, RespURI?, NoResp?,
Cred?, Meta?)
```

Attributes: None

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.3</VerProto>
    <SessionID>1</SessionID>
    <MsgID>1</MsgID>
    <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
    </Target>
    <Source>
      <LocURI>IMEI:493005100592800</LocURI>
    </Source>
  </SyncHdr>
  <SyncBody>
    ...blah, blah...
  </SyncBody>
</SyncML>
```

7.2.3 SyncBody

Usage: Specifies the container for the body or contents of the DM message.

Parent Elements: SyncML

Restrictions: None.

Content Model:

```
((Alert | Atomic | Copy | Exec | Get | Map | Move | Put | Results | Search
| Sequence | Status | Sync | Add | Replace | Delete)+, Final?)
```

Attributes: None.

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    ...blah, blah...
  </SyncHdr>
  <SyncBody>
    <Status>
      <MsgRef>2</MsgRef>
      <CmdID>1</CmdID>
      <CmdRef>0</CmdRef>
      <Cmd>SyncHdr</Cmd>
      <Data>200</Data>
    </Status>
    <Alert>
      <CmdID>2</CmdID>
      <Data>1100</Data> <!—- User displayable notification -->
      <Item></Item>
      <Item>
        <Data>Your antivirus software is being updated.</Data>
      </Item>
    </Alert>
    <Get>
      <CmdID>3</CmdID>
      <Item>
        <Target>
          <LocURI>./antivirus_data/version</LocURI>
        </Target>
      </Item>
    </Get>
    <Final/>
  </SyncBody>
</SyncML>
```

7.3 Data Description Elements

The following element types are used as data description elements for data exchanged in a DM Message.

Command	Support of Management Server		Support of Management Client	
	Sending	Receiving	Sending	Receiving
Data	MUST	MUST	MUST	MUST
Item	MUST	MUST	MUST	MUST
Meta	MUST	MUST	MUST	MUST
Correlator	MAY	MUST	MAY	MAY

7.3.1 Data

Usage: Specifies discrete data.

Parent Elements: Alert, Cred, Item, Status, Search

Restrictions: The content information can be either parsable character data or mark-up data. If the element type contains any mark-up, then the name space for the element types MUST be declared on the element types in the content information.

When specified in an Alert, the element type specifies the type of alert.

When specified in a Cred, the element type specifies the authentication credentials.

When specified in an Item, the element type specifies the item data.

When specified in a Status, the element type specifies the request status code type.

It is REQUIRED that either the mark-up characters of the Data element content are properly escaped according to [XML] specification rules or that the CDATA sections are used.

Content Model:

```
(#PCDATA)
```

Attributes: None.

Example:

```
<Item>
  <Data>MINDT=10</Data>
</Item>
```

7.3.2 Item

Usage: Specifies a container for item data.

Parent Elements: Add, Alert, Copy, Delete, Exec, Field, Get, Put, Move, Record, Replace, Results, Status

Restrictions: If the source URI for the data is an external entity, then the Data element is absent. In this case, the recipient will need to retrieve the data from the specified network location.

Any Data element in Item MUST be the last element in Item.

The LocURI element type in the Target, Source or TargetParent element types for any of the DM commands can be a relative URI. This restriction is not captured by the DM DTD.

When specified in an Add, Copy, Delete, Exec, Get, Put, Replace, or Results command, the element type specifies data item that is the operand for the command.

When specified in an Alert, the element type specifies the parameters for the alert type.

When specified in a Status, the element type specifies additional information about the request status code type. For example, it might specify the component of the request that caused the status condition.

When an Item contains information for a managed node, and the meta format is not null, the Data element MUST be specified.

Content Model:

```
(Target?, Source?, SourceParent?, TargetParent?, Meta?, Data?)
```

Attributes: None.

Example:

```
<Item>
  <Data>MINDT=10</Data>
</Item>
```

7.3.3 Meta

Usage: Specifies meta-information about the parent element type.

Parent Elements: Add, Atomic, Chal, Copy, Cred, Delete, Get, Filter, Item, Map, Move,Put, Replace, Results, Search, Sequence, Sync

Restrictions: When specified in the Chal, the element type specifies meta-information about the authentication scheme requested.

When specified in the Cred, the element type specifies meta-information about the authentication credential.

When specified in the Atomic, Sequence and Sync, then the scope for the meta-information includes all the contained commands, unless the meta-information is overridden by a Meta in a contained command.

When specified in the Results, the element type specifies meta-information about the results set.

When specified in the Add, Copy, Delete, Get, and Replace commands, the element type specifies meta-information about the DM command.

When specified in a command that includes Items (e.g. Add, Delete, Replace, Move or Copy) it is recommended that the scope for the meta-information includes all the contained items. If a contained item also includes Meta information then it is recommended that it is considered to override specific elements within the Meta, not the whole Meta of the containing

command. For example, if a command includes a Type element within a Meta and a contained item includes a Size element within a Meta then it is recommended that the Type element be considered to apply to the contained item.

Content Model:

```
(#PCDATA)
```

Attributes: None

Example:

```
<Cred>
  <Meta>
    <Type xmlns='syncml:metinf'>syncml:auth-md5</Type>
    <Format xmlns='syncml:metinf'>b64</Format>
  </Meta>
  <Data>Zz6EivR3yeaaENcRN6lpAQ==</Data>
</Cred>
```

7.3.4 Correlator

Usage: Specifies a link between an Exec command and an asynchronous response.

Parent Elements: Alert, Exec

Restrictions: **None**

Content Model:

```
(#PCDATA)
```

Attributes: None

Example:

```
<Correlator>
  abc1234
</Correlator>
```

7.4 Meta Information Elements

The following specifies the SyncML Common Meta-Information [META] element types that are used in DM protocol. Use of the elements not listed in this table is implementation specific decision and is not defined by this specification.

Element Type	Support of Management Server		Support of Management Client	
	Sending	Receiving	Sending	Receiving
EMI	MAY	MAY	MAY	MAY
Format	MUST	MUST	MUST	MUST
MaxMsgSize	MAY	MUST	MAY	MUST
MaxObjSize	MUST	MUST	SHOULD	SHOULD
MetInf	MUST	MUST	MUST	MUST
NextNonce	MUST	MUST	MUST	MUST
Size	MUST	MUST	MUST	MUST
Type	MUST	MUST	MUST	MUST

7.5 Protocol Management Elements

The following element types are used to support the DM protocol.

Command	Support of Management Server		Support of Management Client	
	Sending	Receiving	Sending	Receiving
Status	MUST	MUST	MUST	MUST

7.5.1 Status

Usage: Specifies the request status code for a corresponding DM command.

Parent Elements: SyncBody

Restrictions: A Status command only applies to the command corresponding to the specified CmdRef (i.e., 1:1 correspondence of a command and a Status). If there were multiple Item elements specified in the command, and if the items' status code were not the same, then a Status MUST be returned for each of the items. If all of the items had the same status code, a Status for all of the items MAY be returned. In these cases the SourceRef and TargetRef elements are used to identify the Item, which the status code applies to. If all of the items in the command had the same status code, then it is also allowed to return a single Status for the entire command. When returning a single Status for a command with multiple items, the SourceRef and TargetRef elements MUST NOT be specified in the Status command.

Additionally, if the Status command is associated with a command that had other commands inside it (e.g., Sync, Atomic, Sequence), then the status value only applies to the corresponding command, and is not related to the status of the commands inside it.

Ordering of Status commands in a DM response MUST match the order of the commands in the corresponding DM request. That is, when there are multiple commands in a DM request, then the corresponding Status commands MUST appear in the DM response in the same order as the associated commands appeared in the DM request.

In addition, the status on the SyncHdr MUST be the first status element in the SyncBody of the response. Even in the case where the statuses for the previous request span multiple messages/responses, the status on SyncHdr MUST be the first status element followed by other statuses and/or remaining statuses.

The CmdID element type specifies the DM message-unique identifier for this command.

The MsgRef element type specifies the MsgID of the associated DM request.

The CmdRef element type specifies the CmdID of the associated DM request. The element type MUST be present. If "0", the Status command corresponds to a status code for the SyncHdr of the DM message referenced by the Status command.

The Cmd element type specifies the name of the DM command associated with the DM request. The value of the element type can also be "SyncHdr" when the CmdRef element type has a value of "0".

The optional TargetRef element type specifies the target addresses used for the associated command. If the Item element of the command associated with the Status command has a Target element, the path used for this command MUST be put into the TargetRef of the Status command if the TargetRef element is present. If more than one TargetRef element type is specified, then the request status code applies to all of these TargetRef values. If the request status code is applicable to the entire list of multiple items specified in the associated request command, then the TargetRef element type MUST NOT be specified. If the Status element refers to a command containing a single Item, the TargetRef element MAY be omitted to minimize message size. For example, if the server specifies “./Inbox” as the Target, if successfully added, the client would return the actual URI of the new MO in TargetRef.

The OPTIONAL SourceRef element type specifies the source address from the associated command. If the Item element of the command associated with the Status command has a Source element, the value MUST be copied into the SourceRef of the Status command. If more than one SourceRef element type is specified, then the request status code applies to all of these SourceRef values. If the request status code is applicable to the entire list of items specified in the associated request command, then the SourceRef element type MUST NOT be specified.

The Cred element type specifies authentication credential for the command.

The Chal element type specifies the authentication challenge for the command or the message. If the status code in the Data element is (401) Unauthorized or (407) Authentication required, the challenge SHOULD be included.

The Data element type specifies the request status code type.

The OPTIONAL and repeatable Item element type contains additional information about the status condition, such as the DM command.

This specification permits a Status command to be issued against another Status command. This case will probably not normally be encountered. However, there are extreme cases where this feature is necessary. For example, if a server returns a (401) Unauthorized status code with a request for an authentication scheme that is not supported by the client, the client might use a (406) Optional feature unsupported to notify the server that that requested authentication scheme is not supported and negotiate a authentication scheme it does support. DM servers and DM clients not supporting such a usage case need provide no further response to the DM entity issuing the "Status on a Status".

A Status MUST also be returned for the SyncHdr. However, if a client creates a message containing only a successful Status on a SyncHdr, the entire message MUST NOT be sent. A server MUST send this message.

Status codes are listed in Section 11, Response Status Codes.

A Status command MUST NOT be sent in response to a Results command if the Status code is 200 otherwise a Status command MUST be sent. In the case of sending or receiving a large object, Alert 1222 (More Messages) MUST be used to continue the message exchange.

Content Model:

```
(CmdID, MsgRef, CmdRef, Cmd, TargetRef*, SourceRef*, Cred?, Chal?, Data,
Item*)
```

Attributes: None.

Example:

```
<Status>
  <CmdID>1</CmdID>
  <MsgRef>2</MsgRef>
  <CmdRef>0</CmdRef>
  <Cmd>SyncHdr</Cmd>
  <Data>200</Data>
</Status>
```

7.6 Protocol Command Elements

The following element types are used to represent device management commands in a DM Message.

Command	Support of Management Server	Support of Management Client
	Sending	Receiving
Add	MUST	MUST
Atomic	MUST	SHOULD
Copy	MAY	MAY
Delete	MUST	MUST
Exec	MAY	MAY
Get	MUST	MUST
Replace	MUST	MUST
Sequence	MUST	MUST

Command	Support of Management Server	Support of Management Client
	Receiving	Sending
Alert	MUST	MUST
Results	MUST	MUST

7.6.1 Add

Usage: Specifies the DM command to add data to a data collection.

Parent Elements: Atomic, Sequence, Sync, SyncBody

Restrictions: Add creates a new node and returns error if there is an existing node, is not allowed to create node at the Add target URI, or if the specified URI cannot be resolved.

Nodes MUST be added as children of existing interior nodes. The root (.) interior node MUST exist, device manufacturers MAY provide additional existing leaf or interior nodes.

If any parent interior node along the path of the Target LocURI does not exist, the device MAY add it. When these parent interior nodes are implicitly added with success, the Status element MUST include Items with Target element contains URI of implicitly added node. When adding interior nodes implicitly, the ACLs of the implicitly created nodes SHALL be empty, e.g. <Data/>, to allow each such node to inherit the ACL from its parent node. However the exception to this rule, as specified in [DMTND] §7.7.1.1 SHALL apply to implicitly added nodes: If a server is adding an interior node and does not have Replace access rights on the parent of the new node then the device MUST automatically set the ACL of the new node so that the creating server has Add, Delete and Replace rights on the new node.

In case the Add operation fails because the device fails to implicitly add a missing interior node, the status code SHOULD be the same as if the device had tried to add the interior node explicitly. Additionally, the returned Status element in such a failure case SHOULD include an Item element. The Item element, if present, MUST contain a Target element which includes the LocURI of the interior node that the device failed to add.

If the MIME-Type is as defined in [DMTNDS] then multiple nodes MAY be created with one Add command. Client MUST send status code 415, “Unsupported media type or format”, if the device does not support DMTNDS objects. The device can only report one status for all created nodes if the DMTNDS object contains multiple nodes. If the creation of any nodes from the DMTNDS object fails then the client MUST return the same error status code as if that failure node was created with a normal Add command and the devices Management Tree SHOULD not be changed as result of this operation. ACL values MAY be included in the DMTNDS object and these values MUST follow the rules specified in [DMTND] §7.7.1.

Paths in DMTNDS objects are interpreted relative to the target URI in the Add command.

If the adding node’s sub-tree contains the node which is not allowed to be Zero occurrence, the DM Server SHOULD add the node as part of DMTNDS object which contains all required nodes with value.

The mandatory CmdID element type specifies the message-unique identifier for the command.

The Cred element MUST NOT be used at command level.

Meta element type specifies meta-information to be used for the command. Specifying the node type in the meta-information is REQUIRED as specified in [DMTND]. For example, the common media type or format for all the items can be specified. The scope of the meta-information is limited to the command. The Size meta element MAY be used to notify the recipient about the size of the data item being added.

One or more Item element types MUST be specified. The Item element type specifies the data items to be transferred to the recipient. The Target specified within the Item element type MUST be a full device URI.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
(200) OK	The command accessed leaf node and it completed successfully.
(213) Chunked item accepted	Chunked item accepted and buffered
(215) Not executed	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
(216) Atomic roll back OK	Command was inside Atomic element and Atomic failed. This command was rolled back successfully.
(401) Unauthorized	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
(404) Not Found	The specified data item doesn't exist on the recipient. This may also imply that the stated URI for the location of the new management object cannot be resolved
(405) Command not allowed	Command not allowed. The requested command is not allowed on the target.
(407) Authentication required	No authentication credentials were specified. A suitable challenge can also be returned.
(413) Request entity too large	The data item to be transferred is too large (e.g., there are restrictions on the size of data items transferred to the recipient).
(414) URI too long	URI in command is too long. Either string presenting URI or segment in URI is too long or URI has too many segments.
(415) Unsupported media type or format	The media type or format for the data item is not supported by the recipient.
(418) Already exists	The requested Add command failed because the target already exists.
(420) Device full	The recipient device storage is full.
(424) Size mismatch	The chunked object was received, but the size of the received object did not match the size declared within the first chunk.
(425) Permission denied	The server does not have the proper ACL permissions.
(500) Command failed	Non-specific errors created by the recipient while attempting to complete the command.
(516) Atomic roll back failed	Command was inside Atomic element and Atomic failed. This command was not rolled back successfully. Server should take action to try to recover client back into original state.

Content Model:

```
(CmdID, NoResp?, Cred?, Meta?, Item+)
```

Attributes: None.

Example:

```
<Add>
  <CmdID>2</CmdID>
  <Item>
    <Target><LocURI>./SCM/Download/Pakage777</LocURI></Target>
    <Meta>
      <Format xmlns='syncml:metinf'>xml</Format>
      <Type xmlns='syncml:metinf'>
        application/vnd.syncml.dmtnds+xml</Type>
      <Size xmlns='syncml:metinf'>37214</Size>
    </Meta>
    <Data>
      <!--TNDS encoded management subtree -->
    </Data>
  </Item>
</Add>
```

7.6.2 Alert

Usage: Specifies the DM command for sending custom content information to the recipient. The command provides a mechanism for communicating content information, such as state information or notifications to an application on the recipient device. In addition, this command provides a "standard way to specify non-standard" extended commands, beyond those defined in this specification.

Parent Elements: Atomic, Sequence, SyncBody

Restrictions: The Alert command is specifically used to convey notifications, such as device management session requests, to the recipient. For example, a mobile device will use this command to initiate a "client-initiated, management session" with a network server. The mandatory CmdID element type specifies the message-unique identifier for the command.

The Cred element MUST NOT be used at command level.

The mandatory Data element type MUST be used to specify the type of alert.

The Correlator element type MUST be identical to the Correlator value of an Exec command if the alert is sent as an asynchronous response to that Exec command.

Optionally, one or more Item element types MAY be specified. For example, Alert 1224, which is used to send client event information to a server, requires the use of one or more Item elements. Each Item conveys an independent event. Each Item MUST contain a Meta element indicating the Type and Format of the event data.

Currently, any valid DM Type and Format (e.g. “text/plain” and “xml”, respectively) are allowed.

The Item element type specifies parameters for the Alert command. The command returns one of the following status codes.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
(200) OK	The command and the associated Alert action are completed successfully.
(202) Accepted for processing	The command was accepted successfully, but the Alert action has not yet been executed successfully. A subsequent exception condition can be created to relate the eventual completion status of the associated Alert action.
(214) Operation Cancelled	The user cancelled the user interaction Alert.
(215) Not Executed	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
(216) Atomic rollback OK	Command was inside Atomic element and Atomic failed. This command was rolled back successfully.
(304) Not modified	The Confirmation UI Alert produced a negative response from the user.
(401) Unauthorized	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
(405) Command not allowed	The device management protocol does not allow the Alert command to be specified at within the current DM package.
(406) Optional feature not supported	The specified Alert command is not supported by the recipient.
(407) Authentication required	No authentication credentials were specified. A suitable challenge can also be returned. A suitable challenge can also be returned.
(408) Request timeout	The user didn't respond to the user interaction Alert within the timeout period.
(412) Incomplete command	The Alert command didn't include all the correct parameters in the Item element type.
(415) Unsupported media type or format	The media type or format for the data item is not supported by the recipient.
(416) Requested range not satisfiable	The client is not able to display the user interaction Alert because of a device limitation (like too long choice).
(500) Command failed	Non-specific errors created by the recipient while attempting to complete the command.
(516) Atomic rollback failed	Command was inside Atomic element and Atomic failed. This command was not rolled back successfully. Server should take action to try to recover client back into original state.

See alert codes in Section 12 of this document.

Content Model:

```
(CmdID, NoResp?, Cred?, Data?,  Correlator?, Item*)
```

Attributes: None.

Example:

```
<Alert>
  <CmdID>2</CmdID>
  <Data>1200</Data> <!-- Server-initiated session -->
</Alert>
```

7.6.3 Atomic

Usage: Specifies the DM command to request that the subordinate commands be executed as a set or not at all.

Parent Elements: Sequence, Sync, SyncBody

Restrictions: The set of commands inside Atomic MUST be processed in the same way as commands inside Sequence (as described in Section 7.6.15, below), with all subordinate commands to be executed as a set or not at all.

If a client can execute all the atomic commands together (and thus guarantee the result) then a client MAY split the responses up over multiple messages.

If a client cannot execute all the atomic commands together (and thus cannot guarantee the results of commands not executed) and status responses would go into multiple messages, then the Atomic command MUST fail with status code 517 - Atomic response too large to fit in message. Previously executed commands in Atomic command MUST be rolled back.

If a command within Atomic fails, the failure response code corresponding to the failed command within Atomic MUST be returned.

The Atomic, Get and Exec commands MUST NOT be nested within an Atomic command. The client MUST check the syntax of Atomic command before executing the nested commands. If Atomic, Get or Exec is nested within Atomic, the client MUST NOT execute any of the nested commands and the failure response code “(500) Command failed” MUST be returned for the Atomic command. At the same time, the failure response code “(215) Not Executed” MUST be returned for each command within the Atomic command.

The mandatory CmdID element type specifies the message-unique identifier for the command.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
(200) OK	The command completed successfully.
(215) Not executed	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The command is not allowed within a nested Atomic command.
(401) Unauthorized	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
(406) Optional Feature Not Supported	The specified Atomic command is not supported by the recipient.
(407) Authentication required	No authentication credentials were specified. A suitable challenge can also be returned.
(500) Command failed	Atomic command is illegally nested with Atomic, Get or Exec command.
(507) Atomic failed	Error occurs while performing an individual command specified in an Atomic element type.
(517) Atomic Response too large to fit.	The response to an Atomic command was too large to fit in a single message.

Content Model:

```
(CmdID, NoResp?, Meta?, (Add | Delete | Copy | Atomic | Map | Move |
Replace | Sequence | Sync | Get | Exec | Alert)+)
```

Attributes: None.

Example:

```
<Atomic>
  <CmdID>42</CmdID>
  <Alert>
    <!—User confirmation -->
  </Alert>
  <Replace>
    ... blah, blah ...
  </Replace>
</Atomic>
```

7.6.4 Copy

Restrictions: This element is defined in [DMREPDTD], but it is depreciated in OMA Device Management Protocol v1.3.

Usage: Specifies that the DM command to copy data items from one location to another in the recipient's database.

Parent Elements: Atomic, Sequence, Sync, SyncBody

Restrictions: Implementation MUST treat the data of the copy and the data of the original independently after the copy is complete. It is implementation dependent when a physical copy of the item is made in the recipient.

The Copy command in this version of the specification is NOT intended to be used to attempt to change the media type of a data item, compress the data item or otherwise transform a target data item. It is intended to provide a facility for duplicating or moving data (as can be obtained by using Copy followed by a Delete of the original) on the client without having to send this data to a server and back to achieve the same effect.

The mandatory CmdID element type specifies the message-unique identifier for the command.

The Cred element MUST NOT be used at command level.

The optional Meta element type specifies meta-information to be used for the command. For example, the common media type or format for all the items can be specified. The scope of the meta-information is limited to the command.

One or more Item element types MUST be specified. The Item element type specifies the data item to be copied on the recipient's management tree. Copy MUST be specified within an Atomic, Sequence or SyncBody element type and the Target and Source specified within the Item element type in the Copy command MUST be a full device URI.

In this version, the source and the destination nodes MUST be both leaf nodes. Assuming both nodes are leaves, the value of the source node overwrites the value of the target node. If the Copy command cannot be executed because the target node cannot be overwritten with the value of the source node for reasons other than access control rights, (403) Forbidden status MUST be sent back.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
`(200) OK`	The command and the associated Alert action are completed successfully.
`(215) Not executed`	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
`(216) Atomic roll back OK`	Command was inside Atomic element and Atomic failed. This command was rolled back successfully.
`(401) Unauthorized`	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
`(403) Forbidden`	Forbidden. The command could not be executed because the source cannot be copied to the destination URI for reasons other than access control rights.
`(405) Command not allowed`	The requested command is not allowed on the target.
`(406) Optional Feature`	The specified Copy command is not supported by the recipient.

Not Supported	
(407) Authentication required	No authentication credentials were specified. A suitable challenge can also be returned.
(414) URI too long	URI in command is too long. Either string presenting URI or segment in URI is too long or URI has too many segments.
(418) Already exists	The target data item already exists in the recipient management tree.
(420) Device full	There is insufficient space in the recipient management tree for the data item.
(425) Permission denied	The server does not have the proper ACL permissions.
(500) Command failed	Non-specific errors created by the recipient while attempting to complete the command.
(510) Data store failure	Error occurs while the recipient copying the data item within the recipient's management tree.
(516) Atomic roll back failed	Command was inside Atomic element and Atomic failed. This command was not rolled back successfully. Server should take action to try to recover client back into original state.

Content Model:

```
(CmdID, NoResp?, Cred?, Meta?, Item+)
```

Attributes: None.

Example:

```
<Copy>
  <CmdID>4</CmdID>
  <Item>
    <Target>./DM/WAPSetting/1</Target>
    <Source>./Common/WAP/1</Source>
  </Item>
</Copy>
```

7.6.5 Delete

Usage: Specifies the DM command to delete data from a data collection.

Parent Elements: Atomic, Sequence, Sync, SyncBody

Restrictions: The Delete command deletes a node, and the entire sub-tree beneath that node if one exists, subject to access rights and the AccessType status of the node. The purpose of the Delete command is to delete nodes. To delete node values, use the Replace command.

The following rules apply when deleting nodes that have child nodes.

1. If all the child nodes along with the target node can be deleted, a "complete delete" was achieved, and the (200) OK status is returned to indicate this.
2. Permanent nodes cannot be deleted. If attempt to delete a permanent node is made, (405) Command not allowed status is returned.
3. The root node (.) cannot be deleted. Attempts to do so always return the (405) Command not allowed status.

The mandatory CmdID element type specifies the message-unique identifier for the command.

The Cred element MUST NOT be used at command level.

One or more Item element types MUST be specified. The Item element type specifies the data item deleted from the management tree. The Target specified within the Item element type MUST be a full device URI.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
`(200) OK`	The command and the associated individual commands were completed successfully.
`(215) Not executed`	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
`(216) Atomic roll back OK`	Command was inside Atomic element and Atomic failed. This command was rolled back successfully.
`(401) Unauthorized`	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
`(403) Forbidden`	The target of a Delete command is a node that cannot be deleted for reasons other than access control (for example, if the node is in use).
`(404) Not found`	The recipient determined that the data item doesn't exist on the recipient's management tree.
`(405) Command not allowed`	The requested command is not allowed on the target.
`(407) Authentication required`	No authentication credentials were specified. A suitable challenge can also be returned.
`(414) URI too long`	URI in command is too long. Either string presenting URI or segment

	in URI is too long or URI has too many segments.
(425) Permission denied	The server does not have the proper ACL permissions.
(500) Command failed	Non-specific error(s) occurred on the recipient while attempting to complete the command.
(516) Atomic roll back failed	Command was inside Atomic element and Atomic failed. This command was not rolled back successfully. Server should take action to try to recover client back into original state.

Content Model:

```
(CmdID, NoResp?, Archive?, SftDel?, Cred?, Meta?, Item+)
```

Attributes: None.

Example:

```
<Delete>
  <CmdID>5</CmdID>
  <Item>
    <Target>./DM/WAPSetting/1</Target>
  </Item>
</Delete>
```

7.6.6 Exec

Usage: Specifies the DM command to execute an action on the recipient network device.

Parent Elements: SyncBody, Atomic, Sequence

Restrictions: Implementations MUST behave as if the execution were synchronous, i.e. as if the target were executed and returned a value. When used to start a long-running action, Exec SHOULD be implemented to return a status code indicating whether the action was successfully launched, and perhaps a local identifier for that action as well.

The mandatory CmdID element type specifies the message-unique identifier for the command.

The Cred element MUST NOT be used at command level.

The Correlator SHOULD be used if the server is expecting an asynchronous response to an Exec command.

The optional Meta element type specifies meta-information to be used for the command. For example, the common media type or format for all the items can be specified. The scope of the meta-information is limited to the command.

Exactly one Item element type MUST be specified. The Item element type specifies a data item to be used as an argument to the executed process. Exec MUST be specified within a Sequence or SyncBody element type and the Target specified within the Item element type in the Exec command MUST be a full device URI identifying an executable leaf node in the device tree.

Note that the nature of the target of the Exec command, how it interprets arguments, and how it returns values are all dependent upon the node description for the target.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
(200) OK	The command and the associated Alert action are completed successfully.
(202) Accepted for processing	The request to either run a remote execution of an application or to alert a user or application was successfully received.
(215) Not executed	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
(401) Unauthorized	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
(403) Forbidden	Forbidden. The command could not be executed for reasons other than access control rights.
(405) Command not allowed	The requested command is not allowed on the target.
(406) Optional Feature Not Supported	The specified Exec command is not supported by the recipient.
(407) Authentication required	No authentication credentials were specified. A suitable challenge can also be returned.
(414) URI too long	URI in command is too long. Either string presenting URI or segment in URI is too long or URI has too many segments.
(420) Device full	There is insufficient space in the recipient management tree for the data item.
(425) Permission denied	The server does not have the proper ACL permissions.
(500) Command failed	Non-specific errors created by the recipient while attempting to complete the command.
(510) Data store failure	Error occurs while the recipient copying the data item within the recipient's management tree.

Content Model:

```
(CmdID, NoResp?, Cred?, Meta?, Correlator?, Item)
```

Attributes: None.

Example:

```
<Exec>
  <CmdID>3</CmdID>
  <Item>
    <Target>
      <LocURI>./3gppmetrics/operations/start </LocURI>
    </Target>
    <Data> </Data>
  </Item>
</Exec>
```

7.6.7 Get

Usage: Specifies the DM command to retrieve data from the recipient.

Parent Elements: SyncBody, Sequence, Atomic

Restrictions: Data returned from a Get command is returned in a Results element type in a subsequent message. The mandatory CmdID element type specifies the message-unique identifier for the command.

Path element values in DMTNDS objects are interpreted relative to the target URI in the Get command.

If the client does not support DMTNDS and the target of Get command is an interior node, list of the children node names MUST be returned in the Results element. The child list type is defined in [DMTND].

The Cred element MUST NOT be used at command level.

One or more Item element types MUST be specified. The Item element type specifies the data items to be returned from the recipient. The Target specified within the Item element type MUST be a full device URI.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
(200) OK	The command completed successfully.
(215) Not executed	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
(217) OK with inherited ACL	The command completed successfully with inherited ACL returned. The Get command was performed to get ACL on a node which has Empty ACL
(401) Unauthorized	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
(404) Not found	The specified data item doesn't exist on the recipient.
(405) Command not allowed	The requested command is not allowed on the target.
(406) Optional feature not supported	The recipient did not recognize the feature specified after the “?” at the end of the URI.
(407) Authentication	No authentication credentials were specified. A suitable challenge can

required	also be returned.
(413) Request entity too large	The requested data item is too large to be transferred at this time.
(414) URI too long	URI in command is too long. Either string presenting URI or segment in URI is too long or URI has too many segments.
(415) Unsupported media type or format	The media type or format for the data item is not supported by the recipient.
(425) Permission denied	The server does not have the proper ACL permissions.
(500) Command failed	Non-specific errors created by the recipient while attempting to complete the command.

Content Model:

```
(CmdID, NoResp?, Lang?, Cred?, Meta?, Item+)
```

Attributes: None.

Example:

```
<Get>
  <CmdID>4</CmdID>
  <Item>
    <Target>
      <LocURI>./antivirus_data/version</LocURI>
    </Target>
  </Item>
</Get>
```

7.6.8 Map

Restrictions: This element is defined in [DMREPDTD], but it is not used in OMA Device Management Protocol.

7.6.9 MapItem

Restrictions: This element is defined in [DMREPDTD], but it is not used in OMA Device Management Protocol.

7.6.10 Move

Restrictions: This element is defined in [DMREPDTD], but it is not used in OMA Device Management Protocol.

7.6.11 Put

Restrictions: This element is defined in [DMREPDTD], but it is not used in OMA Device Management Protocol. s

7.6.12 Replace

Usage: Specifies the DM command to replace data.

Parent Elements: Atomic, Sequence, Sync, SyncBody

Restrictions: The Replace command is used to overwrite the value of an existing node. If the node does not exist, it MUST NOT be created and status code 404 is returned. Replace will return the status (418) Already Exists if the new name is identical to one of the nodes siblings.

The originator of the command SHOULD determine what features/properties of the data item are supported by the recipient and only send supported properties. The device information document on the recipient contains this information.

If the MIME-Type is as defined in [DMTNDS] then a complete sub-tree MAY be replaced at once. A device MUST NOT replace any nodes if the device does not support the format of data in one (or more) of the DMTNDS node(s), or if the data of a node is out of range (either enumeration or size). If the device accepts the replacement of a complete sub tree then the complete sub tree in the DMTNDS object MUST replace all existing sub nodes in the device. If some of the nodes in the DMTNDS object are new compared to the existing ones in the device then the device MUST create these nodes. If some of the old nodes are not included in the DMTNDS object then the old nodes MUST be deleted. ACL values MAY be included in the DMTNDS object and these values MUST follow the rules specified in [DMTND] §7.7.1.

Client MUST send status code 415, “Unsupported media type or format”, if the device does not support DMTNDS.

The device can only report one status for all replaced nodes if the DMTNDS object contains multiple nodes. If the replace of any nodes from the DMTNDS object fails then the client MUST return the same error status code as if that failure node was replaced with a normal Replace command and the devices Management Tree SHOULD not be changed as result of this operation.

The tree that results from the execution of a Replace command with this MIME-Type MUST be consistent with a tree that would have resulted if the recipient had deleted all sub-nodes and Replaced the first node and thereafter processed a series of successful Add commands, each of which adds one of the nodes of the DMTNDS object.

Paths in DMTNDS objects are interpreted relative to the target URI in the Replace command.

The mandatory CmdID element type specifies the message-unique identifier for the command.

The Cred element MUST NOT be used at command level.

Meta element type specifies meta-information to be used for the command. The scope of the meta-information is limited to the command. The Size meta element MAY be used to notify the recipient about the size of the data item being added.

One or more Item element types MUST be specified. The Item element type specifies the data item replaced in the management tree. The Target and Source specified within the Item element type MUST be a full device URI.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
`(200) OK`	The command accessed an existing leaf node and it completed successfully.
`(213) Chunked item accepted`	Chunked item accepted and buffered.
`(215) Not executed`	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
`(216) Atomic roll back OK`	Command was inside Atomic element and Atomic failed. This command was rolled back successfully.
`(401) Unauthorized`	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
`(403) Forbidden`	The target of a Replace command is a node that cannot be modified for reasons other than access control (for example, if the node is in use).

(404) Not Found	The specified data item doesn't exist on the recipient.
(405) Command not allowed	Command not allowed. The requested command is not allowed on the target. Any attempt to add a child node to a leaf node results in a (405) Command not allowed Status. Additionally, Format, Name and Type properties of permanent nodes cannot be changed, if such an attempt is made, (405) Command not allowed status code is sent back.
(407) Authentication required	No authentication credentials were specified. A suitable challenge can also be returned.
(413) Request entity too large	The data item to be transferred is too large (e.g., there are restrictions on the size of data items transferred to the recipient).
(414) URI too long	URI in command is too long. Either string presenting URI or segment in URI is too long or URI has too many segments.
(415) Unsupported media type or format	The media type or format for the data item is not supported by the recipient.
(418) Already Exists	The requested Replace command failed because the target already exists.
(420) Device full	The recipient device storage is full.
(424) Size mismatch	The chunked object was received, but the size of the received object did not match the size declared within the first chunk.
(425) Permission denied	The server does not have the proper ACL permissions.
(500) Command failed	Non-specific errors created by the recipient while attempting to complete the command.
(516) Atomic roll back failed	Command was inside Atomic element and Atomic failed. This command was not rolled back successfully. Server should take action to try to recover client back into original state.

Content Model:

```
(CmdID, NoResp?, Cred?, Meta?, Item+)
```

Attributes: None.

Example:

```
<Replace>
  <CmdID>4</CmdID>
  <Item>
    <Target>
      <LocURI>./antivirus_data/version</LocURI>
    </Target>
    <Data>antivirus-inc/20020213a/1</Data>
  </Item>
</Replace>
```

7.6.13 Results

Usage: Specifies the DM command that is used to return the results of a Search or Get command.

Parent Elements: SyncBody

Restrictions: The mandatory CmdID element type specifies the DM message-unique identifier for this command.

The OPTIONAL MsgRef element type specifies the MsgID of the associated DM request. If the MsgRef is not present in a Results element type, then the MsgRef value of "1" MUST be assumed.

The CmdRef element type specifies the CmdID of the associated DM request. If not present, the response status code is associated with CmdID value of "1".

The OPTIONAL Meta element type specifies meta-information to be used for the command. For example, the common media type or format for all the items can be specified. The scope of the meta-information is limited to the command.

The OPTIONAL TargetRef element type specifies the target address from the associated command.

The OPTIONAL SourceRef element type specifies the source address from the associated command.

One or more Item element types MUST be specified. The Item element type specifies the results. The Source specified within the Item element type SHOULD be a relative URI, as relative to the corresponding SourceRef.

Results to a command MUST be sent after the Status to the same command.

Content Model:

```
(CmdID, MsgRef?, CmdRef, Meta?, TargetRef?, SourceRef?, Item+)
```

Attributes: None

Example:

```
<Results>
  <MsgRef>1</MsgRef><CmdRef>4</CmdRef>
  <CmdID>3</CmdID>
  <Item>
    <Source>
      <LocURI>./antivirus_data/version</LocURI>
    </Source>
    <Data>antivirus-inc/20010522b/5</Data>
  </Item>
</Results>
```

7.6.14 Search

Restrictions: This element is defined in [DMREPDTD], but it is not used in OMA Device Management Protocol.

7.6.15 Sequence

Usage: Specifies the DM command to order the processing of a set of DM commands.

Parent Elements: Atomic, Sync, SyncBody

Restrictions: The mandatory CmdID element type specifies the message-unique identifier for the command.

One or more Add, Replace, Delete, Copy, Get, Exec or Alert element types MUST be specified. These element types MUST be processed in the specified sequence.

Status code (215) Not Executed MUST be sent back for the commands within the Sequence whose execution was aborted. The status code for the Sequence itself MUST be 200 if you begin executing the Sequence.

The command MUST return a valid status code as defined in section 11, Status codes listed here are for implementation guidance only:

Status code	Meaning
(200) OK	The command completed successfully.
(215) Not executed	Command was not executed, as a result of: • User interaction as user chose to abort or cancel, • The parent Atomic command failed, causing this command to fail.
(216) Atomic roll back OK	Command was inside Atomic element and Atomic failed. This command was rolled back successfully.
(401) Unauthorized	The originator's authentication credentials specify a principal with insufficient rights to complete the command.
(407) Authentication required	No authentication credentials were specified. A suitable challenge can also be returned.
(500) Command failed	Non-specific errors created by the recipient while attempting to complete the command.
(516) Atomic roll back failed	Command was inside Atomic element and Atomic failed. This command was not rolled back successfully. Server should take action to try to recover client back into original state.

Content Model:

```
(CmdID, NoResp?, Meta?, (Add | Replace | Delete | Copy | Atomic | Map |
Move | Sync | Get | Alert | Exec)+)
```

Attributes: None.

Example: The following is an incomplete (i.e., Add and Delete commands only include skeleton content) example for a Sequence command containing two Add commands, followed by a Delete command.

```
<Sequence>
  <CmdID>1234</CmdID>
  <Add>
    <CmdID>1235</CmdID>
    ...blah, blah...
  </Add>
  <Add>
    <CmdID>1236</CmdID>
    ...blah, blah...
  </Add>
  <Delete>
    <CmdID>1237</CmdID>
    ...blah, blah...
  </Delete>
</Sequence>
```

7.6.16 Sync

Restrictions: This element is defined in [DMREPDTD], but it is not used in OMA Device Management Protocol.

8. DM DTD

```
<!--

DM Representation Protocol  V1.2 Document Type Definition



DM Representation is an XML language. Typical usage:

   <?xml version="1.0"?>

   <!DOCTYPE SyncML PUBLIC "-//SYNCML//DTD SYNCML 1.2//EN"

             "http://www.openmobilealliance.org/tech/DTD/OMA-TS-
SyncML_RepPro_DTD-V1_2.dtd"

             [<?oma-syncml-ver supported-versions="1.2"?>]>

   <SyncML>

      ...

   </SyncML>


Terms and conditions of use are available from the

Open Mobile Alliance Ltd. web site at

http://www.openmobilealliance.org/useterms.html

-->


<?xml version="1.0" encoding="UTF-8"?>

<!-- Root Element -->

<!ELEMENT SyncML (SyncHdr, SyncBody)>

<!ELEMENT SyncHdr (VerDTD, VerProto, SessionID, MsgID, Target, Source,
RespURI?, NoResp?, Cred?, Meta?)>

<!ELEMENT SyncBody ((Alert | Atomic | Copy | Exec | Get | Map | Put |
Results | Search | Sequence | Status | Sync | Add | Move | Replace |
Delete)+, Final?)>

<!-- Commonly Used Elements -->

<!-- Archive indicator for Delete -->

<!ELEMENT Archive EMPTY>

<!-- Value must be one of "Add" | "Alert" | "Atomic" | "Copy" | "Delete" |
"Exec" | "Get" | "Map" | "Move" | "Put" | "Replace" | "Results" | "Search"
| "Sequence" |
```

```
"Status" | "Sync". -->
<!ELEMENT Cmd (#PCDATA)>
<!-- Authentication Challenge -->
<!ELEMENT Chal (Meta)>
<!-- Sync message unique identifier for command -->
<!ELEMENT CmdID (#PCDATA)>
<!-- Reference to command identifier -->
<!ELEMENT CmdRef (#PCDATA)>
<!-- Credentials -->
<!ELEMENT Cred (Meta?, Data)>
<!-- Final message flag -->
<!ELEMENT Final EMPTY>
<!-- Desired language for results -->
<!ELEMENT Lang (#PCDATA)>
<!-- Location displayable name -->
<!ELEMENT LocName (#PCDATA)>
<!-- Location URI -->
<!ELEMENT LocURI (#PCDATA)>
<!-- Indication for more data to come -->
<!ELEMENT MoreData EMPTY>
<!-- DM Message ID -->
<!ELEMENT MsgID (#PCDATA)>
<!-- Reference to a DM Message ID -->
<!ELEMENT MsgRef (#PCDATA)>
<!-- No Response Status Requested Indicator -->
<!ELEMENT NoResp EMPTY>
<!-- No Results Requested Indicator -->
<!ELEMENT NoResults EMPTY>
<!-- NumberOfChanges used to display progress information -->
<!ELEMENT NumberOfChanges (#PCDATA)>
<!-- URI recipient used for response -->
<!ELEMENT RespURI (#PCDATA)>
```

```
<!-- DM session identifier -->

<!ELEMENT SessionID (#PCDATA)>

<!-- Soft delete indicator for Delete -->

<!ELEMENT SftDel EMPTY>

<!-- Source location -->

<!ELEMENT Source (LocURI, LocName?)>

<!ELEMENT SourceParent (LocURI)>

<!ELEMENT SourceRef (#PCDATA)>

<!-- Target location information -->

<!ELEMENT Target (LocURI, LocName?, Filter?)>

<!ELEMENT TargetParent (LocURI)>

<!ELEMENT TargetRef (#PCDATA)>

<!-- DM specification major/minor version info. -->

<!-- For this version of the DTD, the value is "1.2" -->

<!ELEMENT VerDTD (#PCDATA)>

<!-- Data sync protocol major/minor version -->

<!-- For example, "xyz/1.2" -->

<!ELEMENT VerProto (#PCDATA)>

<!-- Synchronization data elements -->

<!-- Item element type -->

<!ELEMENT Item (Target?, Source?, SourceParent?, TargetParent?, Meta?,
Data?, MoreData?)>

<!-- Meta element type -->

<!-- Element types in the content MUST have name space declared. -->

<!--The Meta content would be something such as: <Meta> <Type
xmlns='syncml:metinf'>text/calendar</Type> <Format
xmlns='syncml:metinf'>xml</Format> </Meta>-->

<!ELEMENT Meta (#PCDATA)>

<!--Correlator element type -->

<!ELEMENT Correlator (#PCDATA)>

<!-- Actual data content -->

<!ELEMENT Data (#PCDATA)>

<!-- DM Commands -->
```

```
<!-- Add operation. -->

<!ELEMENT Add (CmdID, NoResp?, Cred?, Meta?, Item+)>

<!-- Alert operation. -->

<!-- Alert types are either "User Agent" or "Application" oriented -->

<!ELEMENT Alert (CmdID, NoResp?, Cred?, Data?, Correlator?, Item*)>

<!-- Atomic operation. All or nothing semantics. -->

<!ELEMENT Atomic (CmdID, NoResp?, Meta?, (Add | Replace | Delete | Copy |
Atomic | Map | Move | Sequence | Sync | Get | Exec | Alert)+)>

<!-- Copy operation. -->

<!ELEMENT Copy (CmdID, NoResp?, Cred?, Meta?, Item+)>

<!-- Delete operation. -->

<!ELEMENT Delete (CmdID, NoResp?, Archive?, SftDel?, Cred?, Meta?, Item+)>

<!-- Exec operation -->

<!-- Executable can either be referenced with Target element type -->

<!-- or can be specified in the Data element type. -->

<!ELEMENT Exec (CmdID, NoResp?, Cred?, Meta?, Correlator?, Item)>

<!-- Get operation. -->

<!ELEMENT Get (CmdID, NoResp?, Lang?, Cred?, Meta?, Item+)>

<!-- MAP operation. Create/Delete an item id map kept at the server. -->

<!ELEMENT Map (CmdID, Target, Source, Cred?, Meta?, MapItem+)>

<!ELEMENT MapItem (Target, Source)>

<!-- Move operation. -->

<!ELEMENT Move (CmdID, NoResp?, Cred?, Meta?, Item+)>

<!-- Put operation. -->

<!ELEMENT Put (CmdID, NoResp?, Lang?, Cred?, Meta?, Item+)>

<!-- Replace operation. -->

<!ELEMENT Replace (CmdID, NoResp?, Cred?, Meta?, Item+)>

<!-- Results operation. -->

<!ELEMENT Results (CmdID, MsgRef?, CmdRef, Meta?, TargetRef?, SourceRef?,
Item+)>

<!-- Search operation. -->

<!ELEMENT Search (CmdID, NoResp?, NoResults?, Cred?, Target?, Source+,
Lang?, Meta, Data)>
```

```
<!-- Sequence operation. -->

<!ELEMENT Sequence (CmdID, NoResp?, Meta?, (Add | Replace | Delete | Copy
| Atomic | Map | Move | Sync | Get | Alert | Exec)+)>

<!-- Status operation. -->

<!ELEMENT Status (CmdID, MsgRef, CmdRef, Cmd, TargetRef*, SourceRef*,
Cred?, Chal?, Data, Item*)>

<!-- Synchronize Operation. -->

<!ELEMENT Sync (CmdID, NoResp?, Cred?, Target?, Source?, Meta?,
NumberOfChanges?, (Add | Atomic | Copy | Delete | Move | Replace |
Sequence)*)>

<!-- Filtering operations -->

<!ELEMENT Filter (Meta, Field?, Record?, FilterType?)>

<!ELEMENT Field (Item)>

<!ELEMENT Record (Item)>

<!ELEMENT FilterType (#PCDATA)>

<!-- End of DTD Definition -->
```

9. WBXML Definition

The following tables define the token assignments for the mapping of the DM representation related DTDs and element types into WBXML as defined by [WBXML1.1], or [WBXML1.2], or [WBXML1.3].

9.1 Code Space Definitions

This version of the DM representation protocol specification maps all the DM representation related DTDs into WBXML code spaces.

DTD Name	WBXML PUBLICID Token (Hex Value)	Formal Public Identifier
SyncML	FD1	-//SYNCML//DTD SyncML 1.0//EN
SyncML 1.1	FD3	-//SYNCML//DTD SyncML 1.1//EN
SyncML 1.2	0x1201	-//SYNCML//DTD SyncML 1.2//EN

The DM DTD is assigned the WBXML document public identifier (i.e., the "publicid" WBXML BNF production) associated with the 0x1201 token.

9.2 Code Page Definitions

The following code page tokens represent DM representation related DTD public identifiers. This version of the DM representation protocol specification utilizes the WBXML code page tokens for identifying DTDs.

DTD Name	WBXML Code Page Token (Hex Value)	Formal Public Identifier
SyncML	00	-//SYNCML//DTD SyncML 1.2//EN
MetInf	01	-//SYNCML//DTD MetInf 1.2//EN
Reserved for DM usage	02	Reserved for DM usage

9.3 Token Definitions

The following WBXML token codes represent element types (i.e., tags) form code page x00 (zero), DM DTD.

Element Type Name	WBXML Tag Token (Hex Value)
Add	05
Alert	06
Archive	07
Atomic	08
Chal	09
Cmd	0A
CmdID	0B
CmdRef	0C
Copy	0D

Cred	0E
Data	0F
Delete	10
Exec	11
Final	12
Get	13
Item	14
Lang	15
LocName	16
LocURI	17
Map	18
MapItem	19
Meta	1A
MsgID	1B
MsgRef	1C
NoResp	1D
NoResults	1E
Put	1F
Replace	20
RespURI	21
Results	22
Search	23
Sequence	24
SessionID	25
SftDel	26
Source	27
SourceRef	28
Status	29
Sync	2A

SyncBody	2B
SyncHdr	2C
SyncML	2D
Target	2E
TargetRef	2F
Reserved for future use.	30
VerDTD	31
VerProto	32
NumberOfChanges	33
MoreData	34
Field	35
Filter	36
Record	37
FilterType	38
SourceParent	39
TargetParent	3A
Move	3B
Correlator	3C

The WBXML token codes from code page x01 (one) represent the MetInf DTD. These token definitions are defined in the MetInf DTD specification [DMPRO].

10.Common URI Scheme Types

The following is a list of common URI scheme types

URI Scheme Type	Description
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IMEI	International Mobile Equipment Identifier
LDAP	Lightweight Directory Access Protocol
OBEX	IrDA Object Exchange Protocol
SYNCML	SyncML specific, as defined in one of the protocol or format specifications
WSP	Wireless Session Protocol
ESN	Electronic Serial Number
MEID	Mobile Equipment Identity

11.Response Status Codes

The response status codes in DM message are a numeric text value. The codes are divided into six classes. The only valid values are the standard values defined in this specification.

Note: Some of the status codes are used for data synchronization only.

Status Codes	Reason Phrase
Informational 1xx	
101	In progress. The specified DM command is being carried out, but has not yet completed.
Successful 2xx	
200	OK. The DM command completed successfully.
201	Item added. The requested item was added.
202	Accepted for processing. The request to either run a remote execution of an application or to alert a user or application was successfully performed.
203	Non-authoritative response. The request is being responded to by an entity other than the one targeted. The response is only to be returned when the request would have been resulted in a 200 response code from the authoritative target.
204	No content. The request was successfully completed but no data is being returned. The response code is also returned in response to a Get when the target has no content.
205	This status code is not used by DM.
206	Partial content. The response indicates that only part of the command was completed. If the remainder of the command can be completed later, then when completed another appropriate completion request status code SHOULD be created.
207	This status code is not used by DM.
208	This status code is not used by DM.
209	This status code is not used by DM.
210	This status code is not used by DM.
211	Item not deleted. The requested item was not found. It could have been previously deleted.
212	Authentication accepted. No further authentication is needed for the remainder of the synchronization session. This response code can only be used in response to a request in which the credentials were provided.
213	Chunked item accepted and buffered.
214	Operation cancelled. The DM command completed successfully, but no more commands will be processed within the session.
215	Not executed. A command was not executed, as a result of: • User interaction and user chose not to accept the choice. • The parent Atomic command failed, causing this command to fail.
216	Atomic roll back OK. A command was inside Atomic element and Atomic failed. This command was rolled back successfully.
217	OK with inherited ACL. The command completed successfully

	with inherited ACL returned. The Get command was performed to get ACL on a node which has Empty ACL.
Redirection 3xx	
300	Multiple choices. The requested target is one of a number of multiple alternatives requested target. The alternative SHOULD also be returned in the Item element type in the Status.
301	This status code is not used by DM.
302	Found. The requested target has temporarily moved to a different URI. The original URI SHOULD continue to be used. The URI of the temporary location SHOULD also be returned in the Item element type in the Status. The requestor SHOULD confirm the identity and authority of the temporary URI to act on behalf of the original target URI.
303	See other. The requested target can be found at another URI. The other URI SHOULD be returned in the Item element type in the Status.
304	Not modified. The requested DM command was not executed on the target. This is an additional response that can be added to any of the other Redirection response codes.
305	Use proxy. The requested target MUST be accessed through the specified proxy URI. The proxy URI SHOULD also be returned in the Item element type in the Status.
Originator Exceptions 4xx	
400	Bad request. The requested command could not be performed because of malformed syntax in the command. The malformed command MAY also be returned in the Item element type in the Status.
401	Invalid credentials. The requested command failed because the requestor MUST provide proper authentication. If the property type of authentication was presented in the original request, then the response code indicates that the requested command has been refused for those credentials.
402	Payment needed. The requested command failed because proper payment is needed. This version of DM does not standardize the payment mechanism.
403	Forbidden. The requested command failed, but the recipient understood the requested command. Authentication will not help and the request SHOULD NOT be repeated. If the recipient wishes to make public why the request was denied, then a description MAY be specified in the Item element type in the Status. If the recipient does not wish to make public why the request was denied then the response code 404 MAY be used instead.
404	Not found. The requested target was not found. No indication is given as to whether this is a temporary or permanent condition. The response code 410 SHOULD be used when the condition is permanent and the recipient wishes to make this fact public. This response code is also used when the recipient does not want to make public the reason for why a requested command is not allowed or when no other response code is appropriate.
405	Command not allowed. The requested command is not allowed on the target. The recipient SHOULD return the allowed command for

	the target in the Item element type in the Status.
406	Optional feature not supported. The requested command failed because an OPTIONAL feature in the request was not supported. The unsupported feature SHOULD be specified by the Item element type in the Status.
407	Missing credentials. This response code is similar to 401 except that the response code indicates that the originator MUST first authenticate with the recipient. The recipient SHOULD also return the suitable challenge in the Chal element type in the Status.
408	Request timeout. An expected message was not received within the REQUIRED period of time. The request can be repeated at another time. The RespURI can be used to specify the URI and optionally the date/time after which the originator can repeat the request. See RespURI for details.
409	This status code is not used by DM.
410	Gone. The requested target is no longer on the recipient and no forwarding URI is known.
411	Size REQUIRED. The requested command MUST be accompanied by byte size or length information in the Meta element type.
412	Incomplete command. The requested command failed on the recipient because it was incomplete or incorrectly formed. The recipient SHOULD specify the portion of the command that was incomplete or incorrect in the Item element type in the Status.
413	Request entity too large. The recipient is refusing to perform the requested command because the requested item is larger than the recipient is able or willing to process. If the condition is temporary, the recipient SHOULD also include a Status with the response code 418 and specify a RespURI with the response URI and optionally the date/time that the command SHOULD be repeated.
414	URI too long. The requested command failed because the target URI is too long for what the recipient is able or willing to process. This response code is seldom encountered, but is used when a recipient perceives that an intruder might be attempting to exploit security holes or other defects in order to threaten the recipient.
415	Unsupported media type or format. The unsupported content type or format SHOULD also be identified in the Item element type in the Status.
416	Requested size too big. The request failed because the specified byte size in the request was too big.
417	Retry later. The request failed at this time and the originator SHOULD retry the request later. The recipient SHOULD specify a RespURI with the response URI and the date/time that the command SHOULD be repeated.
418	Already exists. The requested Put or Add command failed because the target already exists.
419	This status code is not used by DM.
420	Device full. The response indicates that the recipient has no more storage space for the remaining synchronization data. The response includes the remaining number of data that could not be returned to the originator in the Item of the Status.
421	This status code is not used by DM.

422	This status code is not used by DM.
423	This status code is not used by DM.
424	Size mismatch. The chunked object was received, but the size of the received object did not match the size declared within the first chunk.
425	Permission Denied. The requested command failed because the sender does not have adequate access control permissions (ACL) on the recipient.
426	This status code is not used by DM.
427	Item Not empty. Parent cannot be deleted since it contains children.
428	This status code is not used by DM.
Recipient Exception 5xx	
500	Command failed. The recipient encountered an unexpected condition which prevented it from fulfilling the request
501	Command not implemented. The recipient does not support the command REQUIRED to fulfill the request. This is the appropriate response when the recipient does not recognize the requested command and is not capable of supporting it for any resource.
502	Bad gateway. The recipient, while acting as a gateway or proxy, received an invalid response from the upstream recipient it accessed in attempting to fulfill the request.
503	Service unavailable. The recipient is currently unable to handle the request due to a temporary overloading or maintenance of the recipient. The implication is that this is a temporary condition; which will be alleviated after some delay. The recipient SHOULD specify the URI and date/time after which the originator SHOULD retry in the RespURI in the response.
504	Gateway timeout. The recipient, while acting as a gateway or proxy, did not receive a timely response from the upstream recipient specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary recipient (e.g. DNS) it needed to access in attempting to complete the request.
505	DTD Version not supported. The recipient does not support or refuses to support the specified version of DM DTD used in the request DM Message. The recipient MUST include the versions it does support in the Item element type in the Status.
506	Processing error. An application error occurred while processing the request. The originator SHOULD retry the request. The RespURI can contain the URI and date/time after which the originator can retry the request.
507	Atomic failed. The error caused all DM commands within an Atomic element type to fail.
508	This status code is not used by DM.
509	Reserved for future use.
510	Data store failure. An error occurred while processing the request. The error is related to a failure in the recipient data store.
511	Server failure. A severe error occurred in the server while processing the request. The originator SHOULD NOT retry the request.

512	This status code is not used by DM.
513	Protocol Version not supported. The recipient does not support or refuses to support the specified version of the DM Protocol used in the request DM Message. The recipient MUST include the versions it does support in the Item element type in the Status.
514	Operation cancelled. The DM command was not completed successfully, since the operation was already cancelled before processing the command. The originator SHOULD repeat the command in the next session.
516	Atomic roll back failed. Command was inside Atomic element and Atomic failed. This command was not rolled back successfully. Server SHOULD take action to try to recover client back into original state.
517	Atomic response too large to fit. The response to an atomic command was too large to fit in a single message.
Application specific codes 1xxx	
1000 – 1999	These status codes are application specific status codes and the meanings of these are not defined in this specification. It is recommended to define status codes with the same grouping as above within this application specific interval but it is the application that defines the allowed values: Informational 11xx Successful 12xx Redirection 13xx Originator Exceptions 14xx Recipient Exception 15xx

12. Alert Codes

Only the alert codes listed in this section are valid in OMA DM Protocol.

OMA DM Protocol alert codes start at 1100.

Alert Code Value	Name	Description
User interaction alert codes		
1100	DISPLAY	The Alert is sent by the server and the client should display the message to provide information to the user.
1101	CONFIRM OR REJECT	This Alert is sent by the server and the client should display the message sent by the server and ask for confirmation. If the user doesn't confirm the operation, reject status MUST be sent back.
1102	TEXT INPUT	The terminal displays the message sent inside the Alert then allows the user to type in a text string. This text string is then sent back to the server in a Status message.
1103	SINGLE CHOICE	The user is presented a set of choices from which he or she is allowed to select only one.
1104	MULTIPLE CHOICE	The user is presented a set of choices from which he or she is allowed to select one or more.
1105 - 1199	-	Reserved for future OMA DM usage.
Device management session alert codes		
1200	SERVER-INITIATED MGMT	Specifies a server-initiated device management session.
1201	CLIENT-INITIATED MGMT	Specifies a client-initiated device management session.
1202 – 1220	-	Reserved for future OMA DM usage.
Special device management alert codes		
1222	NEXT MESSAGE	Specifies a request for the next message in the package. See [DMPRO].
1223	SESSION ABORT	Informs the recipient that the sender wishes to abort the device management session. See [DMPRO].
1224	CLIENT EVENT	Informs the server that an event has occurred on the client. Event data MUST be contained in Data element of an Item element.
1225	NO END OF DATA	End of Data for chunked object not received
1226	GENERIC ALERT	Generic client generated alert with or without a reference to a Management Object

1227	DM_TREE_UNCHANGED_ALERT	Informs the DM Server indicating no changes to the DM Tree have occurred after the last session.
1228	DM_TREE_CHANGED_ALERT	Informs the DM Server that data in the DM Tree has changed after the last session.
1229	REQUESTED_MO	Alert in response to a notification message that contains one or more requested MO(s).
1230	SESSIONLESS ALERT	Alert issued by the DM Client to the DM Server for sessionless reporting.
1231-1299	-	Reserved for future OMA DM usage.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-DM_RepPro-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

B.1.1 Common use elements

The following specifies the static conformance requirements for the message container elements for client devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-CUE-C-001-M	Support for ‘Chal’	Section 7.1.2	
DMREPPRO-CUE-C-002-M	Support for ‘Cmd’	Section 7.1.3	
DMREPPRO-CUE-C-003-M	Support for ‘CmdId’	Section 7.1.4	
DMREPPRO-CUE-C-004-M	Support for ‘CmdRef’	Section 7.1.5	
DMREPPRO-CUE-C-005-M	Support for ‘Cred’	Section 7.1.6	
DMREPPRO-CUE-C-006-M	Support for ‘Final’	Section 7.1.10	
DMREPPRO-CUE-C-007-M	Support for ‘LocName’	Section 7.1.12	
DMREPPRO-CUE-C-008-M	Support for ‘LocURI’	Section 7.1.13	
DMREPPRO-CUE-C-009-O	Support for ‘MoreData’	Section 7.1.14	
DMREPPRO-CUE-C-010-M	Support for ‘MsgID’	Section 7.1.15	
DMREPPRO-CUE-C-011-M	Support for ‘MsgRef’	Section 7.1.16	
DMREPPRO-CUE-C-012-O	Support for sending ‘RespURI’	Section 7.1.21	
DMREPPRO-CUE-C-013-M	Support for receiving ‘RespURI’	Section 7.1.21	
DMREPPRO-CUE-C-014-M	Support for ‘SessionID’	Section 7.1.22	
DMREPPRO-CUE-C-015-M	Support for ‘Source’	Section 7.1.24	
DMREPPRO-CUE-C-016-M	Support for ‘SourceRef’	Section 7.1.26	
DMREPPRO-CUE-C-017-M	Support for ‘Target’	Section 7.1.27	
DMREPPRO-CUE-C-018-M	Support for ‘TargetRef’	Section 7.1.29	

B.1.2 Meta Information elements

The following specifies the static conformance requirements for the meta information elements for client devices that conform to this specification

Item	Function	Reference	Requirement
DMREPPRO-MIE-C-001-O	Support for 'EMI'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-002-M	Support for 'Format'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-003-O	Support for sending 'MaxMsgSize'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-004-M	Support for receiving 'MaxMsgSize'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-005-O	Support for 'MaxObjSize'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-006-M	Support for 'MetInf'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-007-M	Support for 'NextNonce'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-008-M	Support for 'Size'	Section 7.4	DMREPPRO-DDE-C-005-O
DMREPPRO-MIE-C-009-M	Support for 'Type'	Section 7.4	DMREPPRO-DDE-C-005-O

B.1.3 Data description elements

The following specifies the static conformance requirements for the data description elements for client devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-DDE-C-001-O	Support for sending 'Correlator'	Section 7.3.4	DMREPPRO-PCE-C-007-O
DMREPPRO-DDE-C-002-O	Support for receiving 'Correlator'	Section 7.3.4	DMREPPRO-PCE-C-007-O
DMREPPRO-DDE-C-003-O	Support for 'Data' element.	Section 7.3.1	
DMREPPRO-DDE-C-004-O	Support for 'Item' element.	Section 7.3.2	
DMREPPRO-DDE-C-005-O	Support for 'Meta' element.	Section 7.3.3	

B.1.4 Protocol command elements

The following specifies the static conformance requirements for the protocol command elements for client devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-PCE-C-001-M	Support for sending '`Alert`'	Section 7.6.2	
DMREPPRO-PCE-C-002-M	Support for '`Replace`'	Section 7.6.12	
DMREPPRO-PCE-C-003-M	Support for receiving '`Add`'	Section 7.6.1	
DMREPPRO-PCE-C-	Support for receiving	Section 7.6.3	

Item	Function	Reference	Requirement
004-O	‘Atomic’		
DMREPPRO-PCE-C-006-M	Support for receiving ‘Delete’	Section 7.6.5	
DMREPPRO-PCE-C-007-O	Support for receiving ‘Exec’	Section 7.6.6	
DMREPPRO-PCE-C-008-M	Support for receiving ‘Get’	Section 7.6.7	
DMREPPRO-PCE-C-009-M	Support for receiving ‘Sequence’	Section 1.1.1	
DMREPPRO-PCE-C-010-M	Support for sending ‘Results’	Section 7.6.13	

B.1.5 Event Alert

The following specifies the static conformance requirements for the sending of the Event Alert for client devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-Alert-C-001-O	Sending Client Event Alert	Section 7.6.2	

B.1.6 WBXML

The following specifies the static conformance requirements for the WBXML support for client devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-WBXML-C-001-M	Support for receiving WBXML 1.1	Section 6.2	
DMREPPRO-WBXML-C-002-M	Support for receiving WBXML 1.2	Section 6.2	
DMREPPRO-WBXML-C-003-M	Support for receiving WBXML 1.3	Section 6.2	
DMREPPRO-WBXML-C-004-M	Support for sending WBXML 1.1 or 1.2 or 1.3	Section 6.2	

B.1.7 XML Usage

The following specifies the static conformance requirements for the XML Usage support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-XML-C-001-M	Support for namespace usage	Section 5.3	

B.1.8 MIME Usage

The following specifies the static conformance requirements for the MIME Usage support for client devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-MIME-C-001-M	Support for MIME type for DM Message.	Section 5.4	DMREPPRO-MIME-C-002-O OR DMREPPRO-MIME-C-003-O
DMREPPRO-WBXML-C-002-O	Support for "application/vnd.syncml.dm+xml" MIME-type for DM Message.	Section 6.1	
DMREPPRO-WBXML-C-003-O	Support for "application/vnd.syncml.dm+wbxml" MIME-type for DM Message3	Section 6.1	

B.1.9 Identifiers

The following specifies the static conformance requirements for the Identifiers support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-IDS-C-001-M	Support for Identifiers, such as URI, URN and textualnames	Section 5.5	

B.1.10 Message Container Elements

The following specifies the static conformance requirements for the Identifiers support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-MCE-C-001-M	Support for SyncML.	Section 7.2.1	
DMREPPRO-MCE-C-002-M	Support for SyncHdr.	Section 7.2.2	
DMREPPRO-MCE-C-003-M	Support for SyncBody.	Section 7.2.3	

B.1.11 Protocol Management Elements

The following specifies the static conformance requirements for the Identifiers support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-PME-C-001-M	Support for 'Status' element.	Section 7.5.1	

B.2 SCR for DM Server

B.2.1 Common use elements

The following specifies the static conformance requirements for the message container elements for server devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-CUE-S-001-M	Support for ‘Chal’	Section 7.1.2	
DMREPPRO-CUE-S-002-M	Support for ‘Cmd’	Section 7.1.3	
DMREPPRO-CUE-S-003-M	Support for ‘CmdId’	Section 7.1.4	
DMREPPRO-CUE-S-004-M	Support for ‘CmdRef’	Section 7.1.5	
DMREPPRO-CUE-S-005-M	Support for ‘Cred’	Section 7.1.6	
DMREPPRO-CUE-S-006-M	Support for ‘Final’	Section 7.1.10	
DMREPPRO-CUE-S-007-M	Support for ‘LocName’	Section 7.1.12	
DMREPPRO-CUE-S-008-M	Support for ‘LocURI’	Section 7.1.13	
DMREPPRO-CUE-S-009-M	Support for ‘MoreData’	Section 7.1.14	
DMREPPRO-CUE-S-010-M	Support for ‘MsgID’	Section 7.1.15	
DMREPPRO-CUE-S-011-M	Support for ‘MsgRef’	Section 7.1.16	
DMREPPRO-CUE-S-012-O	Support for sending ‘RespURI’	Section 7.1.21	
DMREPPRO-CUE-S-013-M	Support for receiving ‘RespURI’	Section 7.1.21	
DMREPPRO-CUE-S-014-M	Support for ‘SessionID’	Section 7.1.22	
DMREPPRO-CUE-S-015-M	Support for ‘Source’	Section 7.1.24	
DMREPPRO-CUE-S-016-M	Support for ‘SourceRef’	Section 7.1.26	
DMREPPRO-CUE-S-017-M	Support for ‘Target’	Section 7.1.27	
DMREPPRO-CUE-S-018-M	Support for ‘TargetRef’	Section 7.1.29	

B.2.2 Data description elements

The following specifies the static conformance requirements for the data description elements for server devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-DDE-S-001-O	Support for sending 'Correlator'	Section 7.3.4	DMREPPRO-PCE-S-007-M
DMREPPRO-DDE-S-002-M	Support for receiving 'Correlator'	Section 7.3.4	DMREPPRO-PCE-S-007-M

B.2.3 Meta Information elements

The following specifies the static conformance requirements for the meta information elements for server devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-MIE-S-001-O	Support for 'EMI'	Section 7.4	
DMREPPRO-MIE-S-002-M	Support for 'Format'	Section 7.4	
DMREPPRO-MIE-S-003-O	Support for sending 'MaxMsgSize'	Section 7.4	
DMREPPRO-MIE-S-004-M	Support for receiving 'MaxMsgSize'	Section 7.4	
DMREPPRO-MIE-S-005-O	Support for 'MaxObjSize'	Section 7.4	
DMREPPRO-MIE-S-006-M	Support for 'MetInf'	Section 7.4	
DMREPPRO-MIE-S-007-M	Support for 'NextNonce'	Section 7.4	
DMREPPRO-MIE-S-008-M	Support for 'Size'	Section 7.4	
DMREPPRO-MIE-S-009-M	Support for 'Type'	Section 7.4	

B.2.4 Protocol command elements

The following specifies the static conformance requirements for the protocol command elements for server devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-PCE-S-001-M	Support for '`Alert`'	Section 7.6.2	
DMREPPRO-PCE-S-002-M	Support for '`Replace`'	Section 7.6.12	
DMREPPRO-PCE-S-003-M	Support for sending '`Add`'	Section 7.6.1	
DMREPPRO-PCE-S-004-M	Support for sending '`Atomic`'	Section 7.6.3	
DMREPPRO-PCE-S-006-O	Support for sending '`Delete`'	Section 7.6.5	
DMREPPRO-PCE-S-007-M	Support for sending '`Exec`'	Section 7.6.6	
DMREPPRO-PCE-S-008-M	Support for sending '`Get`'	Section 7.6.7	
DMREPPRO-PCE-S-009-M	Support for sending '`Sequence`'	Section 7.6.15	
DMREPPRO-PCE-S-010-M	Support for receiving '`Results`'	Section 7.6.13	

B.2.5 Event Alert

The following specifies the static conformance requirements for the sending of the Event Alert for server devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-Alert-S-001-O	Receiving Client Event Alert	Section 7.6.2	

B.2.6 WBXML

The following specifies the static conformance requirements for the WBXML support for server devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-WBXML-S-001-M	Support for receiving WBXML 1.1	Section 6.2	
DMREPPRO-WBXML-S-002-M	Support for receiving WBXML 1.2	Section 6.2	
DMREPPRO-WBXML-S-003-M	Support for receiving WBXML 1.3	Section 6.2	
DMREPPRO-WBXML-S-004-M	Support for sending WBXML 1.1 or 1.2 or 1.3	Section 6.2	

B.2.7 XML Usage

The following specifies the static conformance requirements for the XML Usage support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-XML-S-001-M	Support for namespace usage	Section 5.3	

B.2.8 MIME Usage

The following specifies the static conformance requirements for the MIME Usage support for client devices that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-MIME-S-001-M	Support for MIME type for DM Message.	Section 5.4	
DMREPPRO-WBXML-S-002-M	Support for “application/vnd.syncml.dm+xml” MIME-type for DM Message.	Section 6.1	
DMREPPRO-WBXML-S-003-M	Support for “application/vnd.syncml.dm+wbxml” MIME-type for DM Message3	Section 6.1	

B.2.9 Identifiers

The following specifies the static conformance requirements for the Identifiers support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-IDS-S-001-M	Support for Identifiers, such as URI, URN and textualnames	Section 5.5	

B.2.10 Message Container Elements

The following specifies the static conformance requirements for the Identifiers support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-MCE-S-001-M	Support for SyncML.	Section 7.2.1	
DMREPPRO-MCE-S-002-M	Support for SyncHdr.	Section 7.2.2	
DMREPPRO-MCE-S-003-M	Support for SyncBody.	Section 7.2.3	

B.2.11 Protocol Management Elements

The following specifies the static conformance requirements for the Identifiers support for client device that conform to this specification.

Item	Function	Reference	Requirement
DMREPPRO-PME-S-001-M	Support for ‘Status’ element.	Section 7.5.1	

Appendix C. MIME Media Type Registration (Informative)

The following section is the MIME media type registrations for OMA Device Management specific MIME media types.

application/vnd.syncml.dm+xml

```
To: ietf-types@iana.org

Subject: Registration of MIME media type application/vnd.syncml.dm+xml

MIME media type name: application

MIME subtype name: vnd.syncml.dm+xml

Required parameters: None

Optional parameters: charset, verproto, verdtd. May be specified in any order
in the Content-Type MIME header field.

Content-Type MIME header.

charset Parameter

Purpose: Specifies the character set used to represent the DM document. The
default character set for DM representation protocol is UTF-8, as defined [RFC
2279].

Formal Specification: The following ABNF defines the syntax for the parameter.

chrset-param = ";" "charset" "=" <any IANA registered charset identifier>

verproto Parameter

Purpose: Specifies the major/minor revision identifiers for the OMA device
management protocol specification for the workflow of messages with OMA DM MIME
content. If present, MUST be the same value as that specified by the "VerProto"
element type in the OMA DM MIME content information. If not present, no default
value is to be assumed.

Formal Specification: The following ABNF defines the syntax for the parameter.

verprot-param = ";" "verproto" "=" "DM/" 1*DIGIT "." 1*DIGIT

verdtd Parameter

Purpose: Specifies the major/minor revision identifiers for the DM
representation protocol specification that defines the OMA DM MIME media type.
If present, MUST be the same value as that specified by the "VerDTD" element
type in the OMA DM MIME content information. If not present, the default value
"1.2" is to be assumed.

Formal Specification: The following ABNF defines the syntax for the parameter.

verdtd-param = ";" "verdtd" "=" 1*DIGIT "." 1*DIGIT
```

Encoding considerations: The default character set for the OMA DM MIME content type is UTF-8. Transfer of this character set through some MIME systems may require that the content is first character encoded into a 7bit character set with an IETF character encoding mechanism such as Base64, as defined in RFC2045.

Security considerations:

Authentication: The OMA DM MIME content type definition provides for the inclusion of authentication information for the purpose of authenticating the originator and recipient of messages containing the device management content type. The content type definition supports Basic, Base64 userid/password mark-up, MD5 digest challenge and response strings and any other registered authentication credential scheme.

Threats: The OMA DM MIME content type definition provides for the inclusion of remote execution commands. Administrators for MIME implementations that support this content type SHOULD take every standard precaution to assure the authentication of the originator of OMA DM content, as well as take every standard precaution to confirm the validity of the included remote execution command prior to allowing the command to be executed on the targeted recipient's system.

Interoperability considerations: Implementations that have support for the mandatory features of this content type will greatly increase the chances of interoperating with other implementations supporting this content type. Conformance to this content type requires an implementation to support every mandatory feature.

Published specification: http://www.openmobilealliance.org. Applications, which use this media type: This MIME content type is intended for common use by networked device management applications.

Additional information:

Magic number(s): None

File extension(s): XDM

Macintosh File Type Code(s): XDML

Person & email address to contact for further information: technical-comments@openmobilealliance.org

Intended usage: COMMON

Author/Change controller: technical-comments@openmobilealliance.org

application/vnd.syncml.dm+wbxml

To: ietf-types@iana.org

Subject: Registration of MIME media type application/vnd.syncml.dm+wbxml

MIME media type name: application

MIME subtype name: vnd.syncml.dm+wbxml

Required parameters: None

Optional parameters: charset, verproto, verdtd. May be specified in any order in the Content-Type MIME header field.

Content-Type MIME header.

charset Parameter

Purpose: Specifies the character set used to represent the DM document. The default character set for DM representation protocol is UTF-8, as defined [RFC 2279].

Formal Specification: The following ABNF defines the syntax for the parameter.

chrset-param = ";" "charset" "=" <any IANA registered charset identifier>

verproto Parameter

Purpose: Specifies the major/minor revision identifiers for the OMA device management protocol specification for the workflow of messages with OMA DM MIME content. If present, MUST be the same value as that specified by the "VerProto" element type in the OMA DM MIME content information. If not present, the default value "DM/1.3" is to be assumed.

Formal Specification: The following ABNF defines the syntax for the parameter.

verprot-param = ";" "verproto" "=" "DM/" 1*DIGIT "." 1*DIGIT

verdtd Parameter

Purpose: Specifies the major/minor revision identifiers for the DM representation protocol specification that defines the OMA DM MIME media type. If present, MUST be the same value as that specified by the "VerDTD" element type in the OMA DM MIME content information. If not present, the default value "1.2" is to be assumed.

Formal Specification: The following ABNF defines the syntax for the parameter.

verdtd-param = ";" "verdtd" "=" 1*DIGIT "." 1*DIGIT

Encoding considerations: The default character set for the OMA DM MIME content type is UTF-8. Transfer of this character set through some MIME systems may require that the content is first character encoded into a 7bit character set with an IETF character encoding mechanism such as Base64, as defined in RFC2045.

Security considerations:

Authentication: The OMA DM MIME content type definition provides for the inclusion of authentication information for the purpose of authenticating the originator and recipient of messages containing the device management content type. The content type definition supports Basic, Base64 userid/password mark-up, MD5 digest challenge and response strings and any other registered authentication credential scheme.

Threats: The OMA DM MIME content type definition provides for the inclusion of remote execution commands. Administrators for MIME implementations that support this content type SHOULD take every standard precaution to assure the authentication of the originator of DM content, as well as take every standard precaution to confirm the validity of the included remote execution command prior to allowing the command to be executed on the targeted recipient's system.

Interoperability considerations: Implementations that have support for the mandatory features of this content type will greatly increase the chances of interoperating with other implementations supporting this content type. Conformance to this content type requires an implementation to support every mandatory feature.

Published specification:

http://www.openmobilealliance.org

Applications, which use this media type: This MIME content type is intended for common use by networked device management applications.

Additional information:

Magic number(s): None

File extension(s): BDM

Macintosh File Type Code(s): BDML

Person & email address to contact for further information: technical-comments@openmobilealliance.org

Intended usage: COMMON

Author/Change controller: technical-comments@openmobilealliance.org