SyncML Representation Protocol Device Management Usage

Approved Version 1.1.2 – 13 June 2003

Open Mobile Alliance
OMA-SyncML-DMRepPro-V1_1_2-20030613-A

Continues the Technical Activities
Originated in the SyncML Initiative

Contents

1. Scope

This document covers the Device Management usage of the SyncML Representation Protocol.

The SyncML Initiative, Ltd. was a not-for-profit corporation formed by a group of companies who co-operated to produce an open specification for data synchronization and device management. Prior to SyncML, data synchronization and device management had been based on a set of different, proprietary protocols, each functioning only with a very limited number of devices, systems and data types. These non-interoperable technologies have complicated the tasks of users, manufacturers, service providers, and developers. Further, a proliferation of different, proprietary data synchronization and device management protocols has placed barriers to the extended use of mobile devices, has restricted data access and delivery and limited the mobility of the users.

SyncML Components

SyncML is a specification that contains the following main components:

- An XML-based representation protocol
- A synchronization protocol and a device management protocol
- Transport bindings for the protocol
- A device description framework for device management

2. References

2.1 Normative References

[DMCONF] "Device Management Conformance Requirements, Version 1.1.2". Open Mobile Alliance™. OMA-SyncML-DMConReqs-V1_1_2. URL:http:www.openmobilealliance.org/tech/docs

[DMPRO] "SyncML Device Management Protocol, Version 1.1.2". Open Mobile Alliance™. OMA-SyncML-DMProtocol-V1_1_2. URL:http:www.openmobilealliance.org/tech/docs

[DMTND] "SyncML Device Management Tree and Description, Version 1.1.2". Open Mobile Alliance™. OMA-SyncML-DMTND-V1_1_2. URL:http:www.openmobilealliance.org/tech/docs

[REPPRO] "SyncML Representation Protocol, version 1.1.2". Open Mobile Alliance™. OMA-SyncML-RepPro-V1_1_2. URL:http:www.openmobilealliance.org/tech/docs

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

2.2 Informative References

None

3. Terminology and Conventions

3.1 Conventions

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

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

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

3.2 Definitions

See SyncML Representation Protocol [REPPRO] and SyncML Device Management Protocol [DMPRO] for definitions of SyncML terms used within this specification.

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

4. Introduction

This document covers the Device Management usage of the SyncML Representation Protocol.

5. SyncML Device Management Usage

5.1 MIME Usage

There are two MIME content types for the SyncML Device Management Message. The MIME content type of `application/vnd.syncml.dm+xml` identifies the clear-text XML representation for the SyncML Message. The MIME content type of `application/vnd.syncml.dm+wbxml` identifies the WBXML binary representation for the SyncML Message. Appendix B 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 SyncML Device Management Messages within transport and session level protocols that support MIME content types.

6. 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.

Restrictions listed in this document are in addition to the restrictions listed in [REPPRO].

6.1 Common Use Elements

The following are common element types used by numerous other SyncML element types.

6.1.1 Archive

Restrictions: This element is not used in SyncML Device Management Protocol.

6.1.2 Chal

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

Example: The following is an example of a SyncML "Basic" 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>
```

6.1.3 Cmd

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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>
```

6.1.4 CmdID

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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>
```

6.1.5 CmdRef

Restrictions: No additional restrictions beyond those defined in [REPPRO]..

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>
```

6.1.6 Cred

Restrictions: Same restriction defined in [REPPRO].. In addition, SyncML 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

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>
```

6.1.7 Final

Restrictions: No additional restrictions beyond those defined in [REPPRO]..

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.1'>
   <SyncHdr>...blah, blah...</SyncHdr>
   </SyncBody>
   ...blah, blah...
   <Final/>
   </SyncBody>
</SyncML>
```

6.1.8 Lang

Restrictions: This element is not used in SyncML Device Management Protocol.

6.1.9 LocName

Restrictions: Used for sending userid for MD5 authentication.

6.1.10 LocURI

Restrictions: No additional restrictions beyond those defined in [REPPRO].

Example:

```
<SyncHdr>
   <VerDTD>1.1</VerDTD>
   <VerProto>DM/1.1</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
   </Target>
   <Source>
      <LocURI>IMEI:493005100592800</LocURI>
   </Source>
</SyncHdr>
```

6.1.11 MoreData

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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>
```

6.1.12 MsgID

Restrictions: No additional restrictions beyond those defined in [REPPRO].

Example:

```
<SyncHdr>
   <VerDTD>1.1</VerDTD>
   <VerProto>DM/1.1</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
   </Target>
   <Source>
      <LocURI>IMEI:493005100592800</LocURI>
   </Source>
</SyncHdr>
```

6.1.13 MsgRef

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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>
```

6.1.14 NoResp

Restrictions: This element is not used in SyncML Device Management Protocol.

6.1.15 NoResults

Restrictions: This element is not used in SyncML Device Management Protocol.

6.1.16 NumberOfChanges

Restrictions: This element is not used in SyncML Device Management Protocol.

6.1.17 RespURI

Restrictions: No additional restrictions beyond those defined in [REPPRO].

Example:

```
<SyncHdr>
   <VerDTD>1.1</VerDTD>
   <VerProto>DM/1.1</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&after=2000
0512T133000Z</RespURI>
</SyncHdr>
```

6.1.18 SessionID

Restrictions: No additional restrictions beyond those defined in [REPPRO].

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.1' >
   <SyncHdr>
      <VerDTD>1.1</VerDTD>
      <VerProto>DM/1.1</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>
```

6.1.19 SftDel

Restrictions: This element is not used in SyncML Device Management Protocol.

6.1.20 Source

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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

```
<SyncHdr>
   <VerDTD>1.1</VerDTD>
   <VerProto>DM/1.1</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
   </Target>
   <Source>
      <LocURI>IMEI:493005100592800</LocURI>
   </Source>
</SyncHdr>
```

6.1.21 SourceRef

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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>
```

6.1.22 Target

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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

```
<SyncHdr>
   <VerDTD>1.1</VerDTD>
   <VerProto>DM/1.1</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
   </Target>
   <Source>
      <LocURI>IMEI:493005100592800</LocURI>
   </Source>
</SyncHdr>
```

6.1.23 TargetRef

Restrictions: No additional restrictions beyond those defined in [REPPRO].

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>
```

6.1.24 VerDTD

Restrictions: No additional restrictions beyond those defined in [REPPRO].

Example:

```
<SyncHdr>
   <VerDTD>1.1</VerDTD>
   <VerProto>DM/1.1</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
   </Target>
   <Source>
      <LocURI>IMEI:493005100592800</LocURI>
   </Source>
</SyncHdr>
```

6.1.25 VerProto

Restrictions: 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 SyncML message conforms to this revision of the SyncML Device Management protocol specification the value MUST be 'DM/1.1'.

Example:

```
<SyncHdr>
   <VerDTD>1.1</VerDTD>
   <VerProto>DM/1.1</VerProto>
   <SessionID>1</SessionID>
   <MsgID>1</MsgID>
   <Target>
      <LocURI>http://www.syncml.org/mgmt-server</LocURI>
   </Target>
   <Source>
      <LocURI>IMEI:493005100592800</LocURI>
   </Source>
</SyncHdr>
```

6.2 Message Container Elements

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

6.2.1 SyncML

Restrictions: 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).

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.1'>
   <SyncHdr>
      <VerDTD>1.1</VerDTD>
      <VerProto>DM/1.1</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>
```

6.2.2 SyncHdr

Restrictions: No additional restrictions beyond those defined in [REPRO].

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.1'>
   <SyncHdr>
      <VerDTD>1.1</VerDTD>
      <VerProto>DM/1.1</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>
```

6.2.3 SyncBody

Restrictions: No additional restrictions beyond those defined in [REPRO].

Example:

```
<SyncML xmlns='SYNCML:SYNCML1.1'>
   <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>
```

6.3 Data Description Elements

The following element types are used as container elements for data exchanged in a SyncML Message.

6.3.1 Data

Restrictions: No additional restrictions beyond those defined in [REPRO].

Example:

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

6.3.2 Item

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

Example:

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

6.3.3 Meta

Restrictions: No additional restrictions beyond those defined in [REPRO].

Example:

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

6.4 Protocol Management Elements

6.4.1 Status

Restrictions: 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.

Example:

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

6.5 Protocol Command Elements

6.5.1 Add

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.

The mandatory `CmdID` element type specifies the SyncML 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 mandatory 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.

Status code	Meaning
(200) OK	The command accessed leaf node and it completed successfully.
(215) Not executed	Command was not executed, as a result of user interaction and user chose to abort or cancel.
(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.
(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.

Example:

```
<Add>
   <CmdID>2</CmdID>
   <Meta>
      <Format xmlns="syncml:metinf">b64</Format>
      <Type xmlns="syncml:metinf">
         application/antivirus-inc.virusdef
      </Type>
   </Meta>
   <Item>
      <Meta>
         <Size xmlns='syncml:metinf'>37214</Size>
      </Meta>
      <Target><LocURI>./antivirus_data</LocURI></Target>
      <Data>
         <!—Base64-coded antivirus file -->
      </Data>
   </Item>
</Add>
```

6.5.2 Alert

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 SyncML message-unique identifier for the command.

The `Cred` element MUST NOT be used at command level.

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

Optionally, one or more Item element types can 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.

In the future, some standard event types may be defined. Currently, any valid SyncML Type and Format (e.g. "text/plain" and "xml", respectively) are allowed.

A server MUST send back status 200 (Ok) when it is capable of processing the Data in the Alert. A server MUST send back status 406 (Optional Feature Not Supported) when it is not able to process the Data in the Alert.

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

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 and user chose to abort or cancel.
(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 SyncML 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 7 of this document.

Example:

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

6.5.3 Atomic

Restrictions: The set of commands inside `Atomic` MUST be processed in the same way as commands inside `Sequence` (as described in Section 6.5.14, 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 an atomic fails, the failure response code MUST be returned.

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

The remainder of the command consists of one or more `Add`, `Alert`, `Delete`, `Copy`, or `Replace` SyncML commands that are the scope of the `Atomic` functionality.

Nested `Atomic` commands and `Get` commands are not legal. A nested `Atomic` command or `Get` command will generate an error `(500) Command failed`.

Status code	Meaning
`(200) OK`	The command completed successfully.
`(215) Not executed`	Command was not executed, as a result of user interaction and user chose to abort or cancel.
`(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`	Nested `Atomic` command was detected.
`(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.

Example:

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

6.5.4 Copy

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 SyncML 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 is sent back.

Status code	Meaning
`(200) OK`	The command and the associated `Alert` action are completed successfully.
`(215) Not executed`	Command was not executed as the user chose to abort/cancel management operation/command.
`(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 Not Supported`	The specified `Copy` 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.
`(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.

Example:

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

6.5.5 Delete

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 has 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 SyncML 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.

Status code	Meaning
`(200) OK`	The command and the associated individual commands are completed successfully.
`(215) Not executed`	Command was not executed as the user chose to abort/cancel management operation/command.
`(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 determines that the data item doesn't exists 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.
`(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.

Example:

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

6.5.6 Exec

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 process, such as a service, `Exec` should be implemented to return a status code indicating whether the process was successfully launched, and perhaps a local identifier for that process as well.

The mandatory `CmdID` element type specifies the SyncML 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.

At least 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.

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.

Status code	Meaning
(200) OK	The command and the associated Alert action are completed successfully.
(215) Not executed	Command was not executed as the user chose to abort/cancel management operation/command.
(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 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.

Example:

```
<Exec>
   <CmdID>3</CmdID>
   <Item>
      <Target>
         <LocURI>./bin/shutdown</LocURI>
      </Target>
      <Data>argument1</Data>
      <Data>argument2</Data>
      <Data>argument3</Data>
   </Item>
</Exec>
```

6.5.7 Get

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

If 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.

Status code	Meaning
(200) OK	The command completed successfully.
(215) Not executed	Command was not executed as the user chose to abort/cancel management operation/command.
(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 required	No authentication credentials were specified. A suitable challenge can 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.

Example:

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

6.5.8 Map

Restrictions: This element is not used in SyncML Device Management Protocol.

6.5.9 MapItem

Restrictions: This element is not used in SyncML Device Management Protocol.

6.5.10 Put

Restrictions: This element is not used in SyncML Device Management Protocol.

6.5.11 Replace

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.

The mandatory `CmdID` element type specifies the SyncML 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.

Status code	Meaning
`(200) OK`	The command accessed an existing leaf node and it completed successfully.
`(215) Not executed`	Command was not executed as the user chose to abort/cancel management operation/command.
`(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.
`(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.

Example:

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

6.5.12 Results

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

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>
```

6.5.13 Search

Restrictions: This element is not used in SyncML Device Management Protocol.

6.5.14 Sequence

Restrictions: The mandatory `CmdID` element type specifies the SyncML 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` will be sent back for the commands whose execution was aborted.

Status code	Meaning
`(200) OK`	The command completed 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.

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>
```

6.5.15 Sync

Restrictions: This element is not used in SyncML Device Management Protocol.

7. Alert Codes

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

SyncML DM Protocol alert codes start at 1100.

Alert Code Value	Name	Description
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 SyncML 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
1226-1299	-	Reserved for future SyncML usage.

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	CONTINUE OR ABORT	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, abort code 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 SyncML usage.

Appendix A. Static Conformance Requirements (Normative)

The Static Conformance Requirements for SyncML Representation, Device Management Usage can be found in [DMCONF].

Appendix B. Change History (Informative)

B.1 Approved Version History

Reference	Date	Description
n/a	n/a	No previous version within OMA

B.2 Draft/Candidate Version 1.1.2 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-SyncML-DMRepPro-V1_1_2	02 Apr 2003	All	The initial version of this document, based on SyncML DM 1.1.1.
	14 Apr 2003	All	Change Requests listed in OMA-DM-2003-0047R3
	08 May 2003	All	Editorial corrections
Candidate Versions OMA-SyncML-DMRepPro-V1_1_2	13 Jun 2003	n/a	Initial Candidate version – approval for status change from TP ref TP doc # OMA-TP-2003-0266R1

Appendix C. MIME Media Type Registration (Informative)

The following section is the MIME media type registrations for SyncML 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 SyncML document. The
default character set for SyncML 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 SyncML device
management protocol specification for the workflow of messages with SyncML MIME
content. If present, MUST be the same value as that specified by the "VerProto"
element type in the SyncML MIME content information. If not present, the
default value "DM/1.1" 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 SyncML
representation protocol specification that defines the SyncML MIME media type.
If present, MUST be the same value as that specified by the "VerDTD" element
type in the SyncML MIME content information. If not present, the default value
"1.1" 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 SyncML 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 SyncML 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 SyncML 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 SyncML 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.syncml.org/docs/syncml_dm_represent_v11_20020215.pdf

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: admins@syncml.org

Intended usage: COMMON

Author/Change controller: admins@syncml.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 SyncML document. The default character set for SyncML 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 SyncML device management protocol specification for the workflow of messages with SyncML MIME content. If present, MUST be the same value as that specified by the "VerProto" element type in the SyncML MIME content information. If not present, the default value "DM/1.1" 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 SyncML representation protocol specification that defines the SyncML MIME media type. If present, MUST be the same value as that specified by the "VerDTD" element type in the SyncML MIME content information. If not present, the default value "1.1" 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 SyncML 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 SyncML 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 SyncML 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 SyncML 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.syncml.org/docs/syncml_dm_represent_v11_20020215.pdf

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: admins@syncml.org

Intended usage: COMMON

Author/Change controller: admins@syncml.org