[MS-MDM]:
Mobile Device Management Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
08/08/2013	1.0	New	Released new document.
11/14/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

Release: Friday, October 25, 2013

1 Introduction

The Mobile Device Management Protocol (MDM) is used for managing devices which have previously enrolled into a management system through the Mobile Device Enrollment Protocol (MDE) [MS-MDE].

MDM is a subset of the Open Mobile Association (OMA) **Device Management** Protocol version 1.2.1 (OMA-TS-DM_Protocol-V1_2_1-20080617-A) [OMA-DMP1.2.1].

Sections 1.8, 2, and 3 of this specification are normative and contain [RFC2119] language. Sections 1.5 and 1.9 are also normative but cannot contain [RFC2119] language. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

> **Common Information Model (CIM)**
> **Common Information Model (CIM) class**
> **Common Information Model (CIM) instance**
> **Common Information Model (CIM) method**
> **Common Information Model (CIM) namespace**
> **Common Information Model (CIM) object**
> **document type definition (DTD)**
> **globally unique identifier (GUID)**
> **Interface Definition Language (IDL)**
> **Windows Management Instrumentation (WMI)**
> **XML schema (XSD)**

The following terms are specific to this document:

> **client:** A client device that is capable of issuing **OMA-DM** commands to a **server** and responding to **OMA-DM** commands issued by a **server**.

> **Distributed Management Task Force (DMTF):** The industry organization that develops management standards and integration technology for enterprise and Internet environments.

> **location URI:** Identifies a source or target **resource**.

> **Managed Object Format (MOF):** A language based on **Interface Definition Language (IDL)** that describes management information. The MOF syntax is a method for describing object definitions in textual form. The MOF Compiler processes a **MOF** file and adds the required object definitions to the **CIM** repository.

> **message:** An atomic unit in the **OMA-DM** protocol.

> **OMA-DM:** See **Open Mobile Alliance (OMA) Device Management**.

> **Open Mobile Alliance (OMA) Device Management:** The industry organization with the goal of specifying protocols and mechanisms to achieve the management of mobile devices, including the configuration of services access and the management of software on mobile devices.

> **package:** A conceptual set of commands that can be spread over multiple messages.

resource: An endpoint that represents a distinct type of management operation or value. A **client** exposes one or more **resources** and some **resources** can have more than one instance. For example, the Win32_LogicalDisk class represents a **resource** and Win32_LogicalDisk="C:\" is a specific instance of the **resource**.

resource URI: The **Uniform Resource Identifier (URI)** that is used to identify a specific type of **resource**, such as disks or processes, on a network. For more information, see [DMTF-DSP0226] section 5.1.2.1.

server: A server capable of issuing **OMA-DM** commands to a **client** and responding to **OMA-DM** commands issued by a **client**.

Uniform Resource Identifier (URI): A string that identifies a **resource**. The **URI** is the Web service addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Uniform Resource Name (URN): This term is used as specified in [RFC1737].

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2   References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [Windows Protocol].

1.2.1   Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-MDE] Microsoft Corporation, "Mobile Device Enrollment Protocol".

[OMA-DMP1.2.1] Open Mobile Alliance, "OMA Device Management Protocol, Approved Version 1.2.1", OMA-TS-DM_Protocol-V1_2_1-20080617-A, June 2008, http://portal.opengeospatial.org/files/?artifact_id=830

[OMA-StdObj1.2.1] Open Mobile Alliance, "OMA Device Management Standardized Objects," Approved version 1.2.1", OMA-TS-DM_StdObj-V1_2_1-20080617-A , June 2008, http://technical.openmobilealliance.org/Technical/release_program/docs/DM/V1_2_1-20080617-A/OMA-TS-DM_StdObj-V1_2_1-20080617-A.pdf

[OMA-SyncMLRP1.2.2] Open Mobile Alliance, "SyncML Representation Protocol, Approved Version 1.2.2", OMA-TS-SyncML-RepPro-V1_2_2-20090724-A, July 2009, http://technical.openmobilealliance.org/Technical/release_program/docs/Common/V1_2_2-20090724-A/OMA-TS-SyncML-RepPro-V1_2_2-20090724-A.pdf

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

1.2.2   Informative References

[DMTF-DSP0226] Distributed Management Task Force, Inc., "Web Services for Management (WS-Management) Specification", version 1.0.0, February 2008, http://dmtf.org/sites/default/files/standards/documents/DSP0226_1.0.0.pdf

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MSDN-IWUAInfo_GetInfo] Microsoft Corporation, "IWindowsUpdateAgentInfo_GetInfo method", http://msdn.microsoft.com/en-us/library/windows/desktop/aa387091(v=vs.85).aspx

[MSDN-SwLicProduct] Microsoft Corporation, "SoftwareLicensingProduct class", http://msdn.microsoft.com/en-us/library/windows/desktop/cc534596(v=vs.85).aspx

[MSDN-SwLicService] Microsoft Corporation, "SoftwareLicensingService class", http://msdn.microsoft.com/en-us/library/windows/desktop/cc534597(v=vs.85).aspx

[MSDN-Win32_Battery] Microsoft Corporation, "Win32_Battery class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394074(v=vs.85).aspx

[MSDN-Win32_BIOS] Microsoft Corporation, "Win32_BIOS class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394077(v=vs.85).aspx

[MSDN-Win32_CmptrSys] Microsoft Corporation, "Win32_ComputerSystem class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394102(v=vs.85).aspx

[MSDN-Win32_CmptrSysProd] Microsoft Corporation, "Win32_ComputerSystemProduct class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394105(v=vs.85).aspx

[MSDN-Win32_CurrentTime] Microsoft Corporation, "Win32_CurrentTime class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394114(v=vs.85).aspx

[MSDN-Win32_DiskDrive] Microsoft Corporation, "Win32_DiskDrive class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394132(v=vs.85).aspx

[MSDN-Win32_DisplayCnfg] Microsoft Corporation, "Win32_DisplayConfiguration class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394137(v=vs.85).aspx

[MSDN-Win32_DsktpMntr] Microsoft Corporation, "Win32_DesktopMonitor class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394122(v=vs.85).aspx

[MSDN-Win32_EncryptVol] Microsoft Corporation, "Win32_EncryptableVolume class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa376483(v=vs.85).aspx

[MSDN-Win32_InfrdDevice] Microsoft Corporation, "Win32_InfraredDevice class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394158(v=vs.85).aspx

[MSDN-Win32_LocalTime] Microsoft Corporation, "Win32_LocalTime class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394171(v=vs.85).aspx

[MSDN-Win32_LogicalDisk] Microsoft Corporation, "Win32_LogicalDisk class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394173(v=vs.85).aspx

[MSDN-Win32_NtwkAdptr] Microsoft Corporation, "Win32_NetworkAdapter class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394216(v=vs.85).aspx

[MSDN-Win32_NtwkAdptrCnfg] Microsoft Corporation, "Win32_NetworkAdapterConfiguration class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394217(v=vs.85).aspx

[MSDN-Win32_OpSys] Microsoft Corporation, "Win32_OperatingSystem class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394239(v=vs.85).aspx

[MSDN-Win32_PhysMemory] Microsoft Corporation, "Win32_PhysicalMemory class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394347(v=vs.85).aspx

[MSDN-Win32_PnPDevice] Microsoft Corporation, "Win32_PnPDevice class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394352(v=vs.85).aspx

[MSDN-Win32_PortBattery] Microsoft Corporation, "Win32_PortableBattery class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394357(v=vs.85).aspx

[MSDN-Win32_Processor] Microsoft Corporation, "Win32_Processor class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394373(v=vs.85).aspx

[MSDN-Win32_QFE] Microsoft Corporation, "Win32_QuickFixEngineering class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394391(v=vs.85).aspx

[MSDN-Win32_Service] Microsoft Corporation, "Win32_Service class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394418(v=vs.85).aspx

[MSDN-Win32_Share] Microsoft Corporation, "Win32_Share class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa394435(v=vs.85).aspx

[MSDN-Win32_SysBIOS] Microsoft Corporation, "Win32_SystemBIOS class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394467(v=vs.85).aspx

[MSDN-Win32_SysEnclose] Microsoft Corporation, "Win32_SystemEnclosure class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394474(v=vs.85).aspx

[MSDN-Win32_TimeZone] Microsoft Corporation, "Win32_TimeZone class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394498(v=vs.85).aspx

[MSDN-Win32_Tpm] Microsoft Corporation, "Win32_Tpm class", http://msdn.microsoft.com/en-us/library/windows/desktop/aa376484(v=vs.85).aspx

[MSDN-Win32_UTCTime] Microsoft Corporation, "Win32_UTCTime class",
http://msdn.microsoft.com/en-us/library/windows/desktop/aa394510(v=vs.85).aspx

[RFC1737] Sollins, K., and Masinter, L., "Functional Requirements for Uniform Resource Names",
RFC 1737, December 1994, http://www.rfc-editor.org/rfc/rfc1737.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI):
Generic Syntax", STD 66, RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.txt

[RFC5023] Gregorio, J. Ed., and de hOra, B., Ed., "The Atom Publishing Protocol", RFC 5023,
October 2007, http://www.ietf.org/rfc/rfc5023.txt

1.3 Overview

The Mobile Device Management Protocol is a **client**/**server** protocol that is used to manage mobile devices which have previously been enrolled into a management service by using the Mobile Device Enrollment Protocol (MDE) [MS-MDE].

MDM supports the following capabilities:

- Inventory collection

- Settings management

- Application management

- Certificate provisioning

- Wi-Fi & VPN profile management

- Data protection

In this document, the endpoint that initiates the HTTP connection and sends HTTP request messages is referred to as the client. The **entity** that responds to the HTTP connection request and sends HTTP response messages is referred to as the server.

1.4 Relationship to Other Protocols

MDM depends on HTTP for the transfer of all protocol messages [RFC2616].
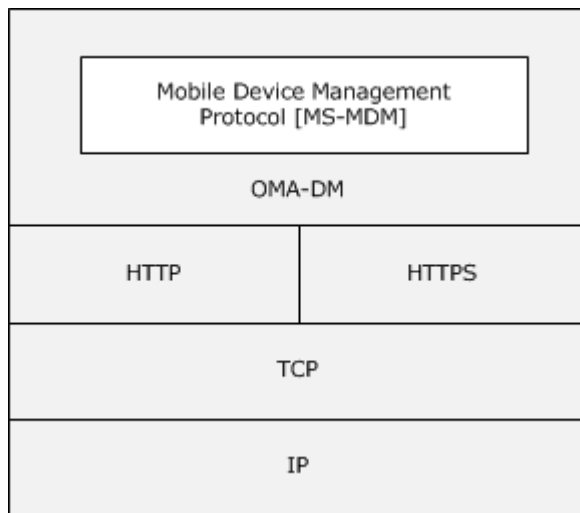
Figure 1: Relationship to other protocols

1.5 Prerequisites/Preconditions

The Mobile Device Enrollment Protocol (MDE) is a prerequisite to using this protocol. Before a device can be managed by using MDM, the device has to already be enrolled in a management service by using MDE. Configuration information for bootstrapping MDM is persisted on the device as part of the enrollment process. The location and the method for retrieving configuration information is implementation-specific.

MDM configuration information includes:

- Service endpoint

- Identity certificate for TLS HTTPS mutual authentication

1.6   Applicability Statement

A device has to be enrolled in a management service through the use of MDE before the device can then be managed by using MDM.

1.7   Versioning and Capability Negotiation

None.

1.8   Vendor-Extensible Fields

None.

1.9   Standards Assignments

Parameter	Value	Reference
TCP port	443	Section 2.1

2   Messages

MDM is based on the **OMA-DM** protocol [OMA-DMP1.2.1]. Messages are issued by a requester and results and status are returned by a responder as defined in [OMA-SyncMLRP1.2.2]. MDM does not modify or extend these messages in any manner.

2.1   Transport

MDM, both as defined in this document and the **OMA-DM** protocol [OMA-DMP1.2.1], uses HTTP (as specified in [RFC2616]) as the transport layer. HTTP operations are performed on resources identified by a **URI**. MDM extends the resource addressing rules used by HTTP for URI formatting as specified in section 2.2.3.

This document does not prescribe a mechanism to secure (authenticate, encrypt, and so on) MDM communications. For security recommendations relating to the protocol transport layer, see [RFC5023] section 15.

2.2   Message Syntax

2.2.1   Namespaces

This document defines and references various XML namespaces that use the mechanisms specified in [XMLNS]. Although this document associates a specific XML namespace prefix with each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
SyncML	xmlns='SYNCML:SYNCML1.2'	Section [OMA-SyncMLRP1.2.2]

2.2.2   SyncML Message

A SyncML message is a well-formed XML document that adheres to the **document type definition (DTD)**, but which does not require validation. While a SyncML message does not require validation, the XML in the document MUST adhere to the explicit order defined in the DTD. The XML document is identified by a SyncML (section 2.2.4.1) document (or root) element type that serves as a parent container for the SyncML message.

The SyncML message consists of a header specified by the SyncHdr (section 2.2.4.2) element type and a body specified by the SyncBody (section 2.2.4.3) element type. The **SyncML header** identifies the routing and versioning information about the SyncML message. The **SyncML body** functions as a container for one or more SyncML commands (see section 2.2.7).

A SyncML command is specified by individual element types that provide specific details about the command, including any data or meta-information. The command serves as a container for these element types (see section 2.2.3).

MDM uses a subset of the SyncML message definition specified in [OMA-SyncMLRP1.2.2]. The following snippet identifies the required elements for a SyncML message as implemented by MDM.

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
```

```
     <VerDTD>1.2</VerDTD>
     <VerProto>DM/1.2</VerProto>
     <SessionID>1</SessionID>
     <MsgID>1</MsgID>
     <Target>
       <LocURI>{unique device ID}</LocURI>
     </Target>
     <Source>
       <LocURI>{management server url}</LocURI>
     </Source>
   </SyncHdr>
   <SyncBody>{Command}</SyncBody>
 </SyncML>
```

2.2.3   Common Use Elements

The following sections identify common element types used by other SyncML element types.

2.2.3.1   Cmd

The **Cmd** element type specifies the name of the SyncML command that is referenced by a Status (section 2.2.6.1) element type. The element type has the following syntax:

```
   <Cmd>(#PCDATA)</Cmd>
```

Parent Element: Status (section 2.2.6.1)

Restrictions: **Cmd** MUST be a string that is one of the following SyncML command names: Add, Atomic, Delete, Exec, Get, Replace, Results, or Status.

Content Model: (#PCDATA)

Attributes: None.

2.2.3.2   CmdID

The **CmdID** element type specifies a unique command identifier for the SyncML message. The element type has the following syntax:

```
   <CmdID>(#PCDATA)</CmdID>
```

Parent Elements: Add (section 2.2.7.1), Alert (section 2.2.7.2), Atomic (section 2.2.7.3), Delete (section 2.2.7.4), Exec (section 2.2.7.5), Get (section 2.2.7.6), Replace (section 2.2.7.7), Results (section 2.2.7.8), Status (section 2.2.6.1)

Restrictions: **CmdID** MUST always be present in the SyncML message. **CmdID** MUST be unique within the SyncML message and MUST NOT be the string "0".

Content Model: (#PCDATA)

Attributes: None.

2.2.3.3 CmdRef

The **CmdRef** element type specifies a reference to a CmdID that is used by the Status (section 2.2.6.1) element type. The element type has the following syntax:

```
<CmdRef>(#PCDATA)</CmdRef>
```

Parent Elements: Results (section 2.2.7.8), Status

Restrictions: CmdRef MUST refer to the CmdID (section 2.2.3.2) of the SyncML command referred to by Status. **CmdRef** MUST be present in the SyncML message, except when the Status command refers to the SyncHdr (section 2.2.4.2) of the associated SyncML request message. For example, a status can be sent back to the originator for exceptions (that is, (401) Unauthorized) found within the SyncHdr of the originator's request.

Content Model: (#PCDATA)

Attributes: None.

2.2.3.4 Final

The **Final** element type indicates that a SyncML message is the last message in the current SyncML package. The element type has the following syntax:

```
<Final></Final>
```

Parent Element: SyncBody (section 2.2.4.3)

Restrictions: Final MUST only be specified in the last SyncML message in a SyncML package. When **Final** is not present in a SyncML message, more messages follow the SyncML message in the current SyncML package. The semantics for the different SyncML packages are specified by the SyncML Representation Protocol [OMA-SyncMLRP1.2.2] and the **OMA-DM** protocol [OMA-DMP1.2.1].

Content Model: (EMPTY)

Attributes: None.

2.2.3.5 LocURI

The **LocURI** element type specifies the target or source-specific address. The element type has the following syntax:

```
<LocURI>(#PCDATA)</LocURI>
```

Parent Elements: Target (section 2.2.3.11), Source (section 2.2.3.9), Item (section 2.2.5.2)

Restrictions: LocURI MUST be either an absolute or a relative URI, or a well-known **Uniform Resource Name (URN)**.

Content Model: (#PCDATA)

Attributes: None.

2.2.3.6 MsgID

The **MsgID** element type specifies a unique SyncML session identifier for the SyncML message. The element type has the following syntax:

```
<MsgID>(#PCDATA)</MsdID>
```

Parent Element: SyncHdr (section 2.2.4.2)

Restrictions: MsgID MUST be specified in the SyncHdr and MUST be unique to the device within the SyncML session. The value is a monotonically increasing numeric value that starts at one (1) for the first message in the SyncML session. The **MsgID** specified in a SyncML request MUST be the content of the MsgRef (section 2.2.3.7) element type specified in the corresponding SyncML Results (section 2.2.7.8) or response Status (section 2.2.6.1).

Content Model: (#PCDATA)

Attributes: None.

2.2.3.7 MsgRef

The **MsgRef** element type specifies a reference to a MsgID (section 2.2.3.6) that is used by a SyncML Results (section 2.2.7.8) or response Status (section 2.2.6.1). The element type has the following syntax:

```
<MsgRef>(#PCDATA)</MsgRef>
```

Parent Elements: Results, **Status**

Restrictions: MsgRef MUST reference the MsgID (section 2.2.3.6) of the SyncML message referred to by a SyncML **Results** or response **Status**.

Content Model: (#PCDATA)

Attributes: None.

2.2.3.8 SessionID

The **SessionID** element type specifies the identifier of the SyncML session that is associated with the SyncML message. The **SessionID** can remain valid across the exchange of many SyncML messages between the client and server. The element type has the following syntax:

The element type has the following syntax:

```
<SessionID>(#PCDATA)</SessionID>
```

Parent Element: SyncHdr (section 2.2.4.2)

Restrictions: SessionID is an opaque string that MUST be specified in the SyncHdr in all SyncML messages. The initiator SHOULD use a unique **SessionID** for each session. Note that for practical implementations for a client, using an 8-bit incrementing **SessionID** counter is sufficient. The maximum length of a **SessionID** is 4 bytes.

Content Model: (#PCDATA)

Attributes: None.

2.2.3.9   Source

The **Source** element type specifies source routing or mapping information. The element type has the following syntax:

```
<Source>(LocURI)</Source>
```

Parent Elements: Item (section 2.2.5.2), SyncHdr (section 2.2.4.2)

Restrictions: When specified in the Item element type, **Source** specifies the resource item that is the source of the SyncML command. When specified in the SyncHdr element type, **Source** specifies the source routing information for the network device that originated the SyncML message.

Content Model: (LocURI (section 2.2.3.5))

Attributes: None.

2.2.3.10   SourceRef

The **SourceRef** element type specifies the Source (section 2.2.3.9) referenced by a Status (section 2.2.6.1) element type. The element type has the following syntax:

```
<SourceRef>(LocURI)</SourceRef>
```

Parent Elements: Status

Restrictions: When specified in the Status element type, **SourceRef** identifies the source address specified in the command associated with the response status.

The element type MAY be specified in a Status element command corresponding to any SyncML command that includes the **Source** element type.

Content Model: (LocURI (section 2.2.3.5))

Attributes: None.

2.2.3.11   Target

The **Target** element type specifies target routing information. The element type has the following syntax:

```
<Target>(LocURI)</Target>
```

Parent Elements: Item (section 2.2.5.2), SyncHdr (section 2.2.4.2)

Restrictions: When specified in the Item element type, **Target** specifies the **WMI** class that is the target of the SyncML command. When specified in the SyncHdr element type, **Target** specifies the target routing information for the network device that is receiving the SyncML message.

Content Model: (LocURI (section 2.2.3.5))

Attributes: None.

2.2.3.12   TargetRef

The **TargetRef** element type specifies the Target (section 2.2.3.11) referenced by a Status (section 2.2.6.1) element type. The element type has the following syntax:

```
<TargetRef>(LocURI)</TargetRef>
```

Parent Elements: Status

Restrictions: When specified in the Status element type, **TargetRef** identifies the target address specified in the command associated with the response status.

The element type MAY be specified in a Status element command corresponding to any SyncML command that includes the Target element type.

Content Model: (LocURI (section 2.2.3.5))

Attributes: None.

2.2.3.13   VerDTD

The **VerDTD** element type specifies the major and minor version identifier of the SyncML representation protocol specification that is used to represent the SyncML message. The element type has the following syntax:

```
<VerDTD>(#PCDATA)</VerDTD>
```

Parent Element: SyncHdr (section 2.2.4.2)

Restrictions: VerDTD MUST be specified in the SyncHdr. When the XML document conforms to the current revision of the SyncML representation protocol specification, **VerDTD** MUST be 1.2. Note that major revisions to a specification can create incompatibilities that generally require a new SyncML parser. Minor revisions involve changes that do not impact basic compatibility of the parser.

Content Model: (#PCDATA)

Attributes: None.

2.2.3.14   VerProto

The **VerProto** element type specifies the major and minor version identifier of the SyncML representation protocol specification that is used to represent the SyncML message. The element type has the following syntax:

```
<VerProto>(#PCDATA)</VerProto>
```

Parent Element: SyncHdr (section 2.2.4.2)

Restrictions: VerProto MUST be specified in the SyncHdr. When the XML document conforms to the current revision of the SyncML representation protocol specification, **VerProto** MUST be 1.2. Note that major revisions to a specification can create incompatibilities that generally require a new SyncML parser. Minor revisions involve changes that do not impact basic compatibility of the parser.

Content Model: (#PCDATA)

Attributes: None.

2.2.4   Message Container Elements

Message container elements provide basic container support for the SyncML message.

2.2.4.1   SyncML

The **SyncML** element type serves as the container for a SyncML Message. The element type has the following syntax:

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

Parent Element: None. **SyncML** is the root (or document) element.

Restrictions: None.

Content Model: (SyncHdr (section 2.2.4.2), SyncBody (section 2.2.4.3))

Attributes: SyncML requires the **xmlns** attribute. The attribute type is CDATA and the value MUST be the string "SYNCML:SYNCML1.2".

2.2.4.2   SyncHdr

The **SyncHdr** element type serves as the container for the revisioning routing information in the SyncML message. The element type has the following syntax:

```
<SyncHdr>(VerDTD, VerProto, SessionID, MsgID, Target, Source, Meta)</SyncHdr>
```

Parent Element: SyncML (section 2.2.4.1)

Restrictions: Use of the Meta (section 2.2.5.3) element type is optional. **Meta** is used to convey meta-information about the SyncML messages, such as the maximum byte size of a SyncML response.

Content Model: (VerDTD (section 2.2.3.13), VerProto (section 2.2.3.14), SessionID (section 2.2.3.8), MsgID (section 2.2.3.6), Target (section 2.2.3.11), Source (section 2.2.3.9), Meta)

Attributes: None.

2.2.4.3 SyncBody

The **SyncBody** element type serves as the container for the body or contents of the SyncML message. The element type has the following syntax:

```
<SyncBody>((Atomic | Exec | Get | Results | Status | Add | Replace | Delete)+,
Final?)</SyncBody>
```

Parent Element: SyncML (section 2.2.4.1)

Restrictions: None.

Content Model: ((Atomic (section 2.2.7.3) | Exec (section 2.2.7.5) | Get (section 2.2.7.6) | Results (section 2.2.7.8) | Status (section 2.2.6.1) | Add (section 2.2.7.1) | Replace (section 2.2.7.7) | Delete (section 2.2.7.4))+, Final (section 2.2.3.4)?)

Attributes: None.

2.2.5 Data Description Elements

Data description elements are used as container elements for data exchanged in a SyncML message.

2.2.5.1 Data

The **Data** element type provides a container for discrete SyncML data. The element type has the following syntax:

```
<Data>(#PCDATA)</Data>
```

Parent Elements: Item (section 2.2.5.2), Status (section 2.2.6.1)

Restrictions: Data can either be parsable character content or markup content. If **Data** contains any markup content, the namespace for the element types MUST be declared on the element types in the content. When **Data** is specified in an **Item**, the element type specifies the item data. When **Data** is specified in a **Status**, the element type specifies the request status code type.

Content Model: (#PCDATA)

Attributes: None.

2.2.5.2 Item

The **Item** element type provides a container for item data. The element type has the following syntax:

```
<Item>(#PCDATA)</Item>
```

Parent Elements: Add (section 2.2.7.1), Delete (section 2.2.7.4), Exec (section 2.2.7.5), Get (section 2.2.7.6), Replace (section 2.2.7.7), Results (section 2.2.7.8), Status (section 2.2.6.1)

Restrictions: When the source **URI** for the item data is an external entity, the Data (section 2.2.5.1) element is not present and the recipient retrieves the data from the specified network location. When **Data** is present in **Item**, it MUST be the last element in **Item**.

The LocURI (section 2.2.3.5) element type can be a relative URL when used in the Target (section 2.2.3.12) or Source (section 2.2.3.9) element types for any of the SyncML commands. Note that this restriction is not captured by the SyncML **DTD**.

When specified in an **Add**, **Delete**, **Exec**, **Get**, **Replace**, or **Results** command, **Item** specifies the data item that is the operand for the command. When specified in **Status**, **Item** specifies additional information about the request status code type. For example, it might specify the component of the request that caused the status condition.

Content Model: (#PCDATA)

Attributes: None

2.2.5.3   Meta

The **Meta** element type provides a container for meta-information about the parent element type. The element type has the following syntax:

```
<Meta>(#PCDATA)</Meta>
```

Parent Elements: Add (section 2.2.7.1), Atomic (section 2.2.7.3), Delete (section 2.2.7.4), Get (section 2.2.7.6), Item (section 2.2.5.2), Replace (section 2.2.7.7), Results (section 2.2.7.8)

Restrictions: When **Meta** is specified in an Atomic or Sync command, the scope of the meta-information includes all of the contained commands, unless the meta-information is overridden by another **Meta** element in a contained command.

- When **Meta** is specified in Results, **Meta** specifies meta-information about the results set.

- When **Meta** is specified in **Add**, **Delete**, **Get**, or **Replace**, **Meta** specifies meta-information about the SyncML command.

- When **Meta** is specified in a command that includes one or more **Item** elements (e.g., **Add**, **Delete**, **Replace**), it is recommended that the scope for the meta-information includes all of the contained items. If a contained item also includes a **Meta** element, it is recommended that the included **Meta** element override specific elements within the containing **Meta** and not the entire contents. For example, if a command includes a Type element within **Meta** and a contained item includes a Size element within another **Meta**, then the Type element SHOULD be considered to apply to the contained item.

Content Model: (#PCDATA)

Attributes: None.

2.2.6   Protocol Management Elements

The **Status** element provides protocol management support for the SyncML message.

2.2.6.1 Status

The **Status** element type specifies the request status code for a corresponding SyncML command. Status codes are specified in [OMA-SyncMLRP1.2.2] section 10. The element type has the following syntax:

```
<Status>(CmdID, MsgRef, CmdRef, Cmd, Data, Item?+, TargetRef?, SourceRef?)</Status>
```

Parent Element: SyncBody (section 2.2.4.3)

Restrictions: Status only applies to the command corresponding to the specified CmdRef (section 2.2.3.3); there is a 1:1 correspondence between a command and the **Status** element. When **Status** corresponds to a command that contains other commands, the **Status** applies only to the corresponding command and not to any contained commands. The order of **Status** elements in a SyncML response MUST match the order of the corresponding commands in the SyncML request.

- The Item (section 2.2.5.2) element type is optional and can be present multiple times as required. Item contains additional information about the status condition, such as the SyncML command. When multiple Item (section 2.2.5.2) elements are specified in a command, if the status codes for all Items are not identical, a unique **Status** element MUST be returned for each Item. If all status codes are identical, the same **Status** element MAY be returned for all Items.

- **Status** MUST be returned for the SyncHdr (section 2.2.4.2) and MUST be the first **Status** element in the SyncBody of the response. Even in the case where **Status** elements for a previous request span multiple messages and responses, the **Status** in the SyncHdr MUST be the first **Status** element in the SyncBody followed by other **Status** elements and/or remaining **Status** elements for previous requests. However, when a client creates a message containing only a successful **Status** in a SyncHdr, the entire message MUST NOT be sent. A server MUST send this message.

- The CmdID (section 2.2.3.2) element type specifies the unique identifier for the SyncML message for the command.

- The MsgRef (section 2.2.3.7) element type specifies the MsgID (section 2.2.3.6) for the associated **SyncML request** from the server.

- The CmdRef element type MUST be present and specifies the CmdID for the associated **SyncML request** from the server. When CmdRef is zero, **Status** is a status code for the SyncHdr of the SyncML message referenced by the command corresponding to the **Status**.

- The Cmd (section 2.2.3.1) element type specifies the name of the SyncML command associated with the **SyncML request**. When CmdRef is "0", Cmd can also be set to `"SyncHdr"`.

- The SourceRef (section 2.2.3.10) and TargetRef (section 2.2.3.12) elements are used to identify the Item to which a status code applies. When returning a single **Status** for a command, the SourceRef and TargetRef elements MUST NOT be specified in the **Status**.

- TargetRef is an optional element that specifies the target addresses from the corresponding command. When the Item element of the corresponding command includes a Target (section 2.2.3.11) element, the value of Item. Target MAY be copied into the **Status**.TargetRef element. When more than one TargetRef element is specified, the request status code applies to all TargetRef elements. If the request status code is applicable to all Items specified in the associated request command, TargetRef MUST NOT be specified. When **Status** corresponds to a command that contains a single Item, TargetRef MAY be omitted to minimize message size.

- SourceRef is an optional element that specifies the source address from the corresponding command. When the Item element of the corresponding command includes a [Source (section 2.2.3.9)](#) element, the value of Item. Source MAY be copied into the **Status** SourceRef element. When more than one SourceRef element is specified, the request status code applies to all SourceRef elements. If the request status code is applicable to all Items specified in the associated request command, SourceRef MUST NOT be specified.

- The [Data (section 2.2.5.1)](#) element type specifies the request status code type. Status codes are specified in [OMA-SyncMLRP1.2.2] section 10.

- MDM permits a **Status** to be issued against another **Status** (or, **Status** on a **Status**). While this case is not usually encountered, 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 code to notify the server that that requested authentication scheme is not supported and negotiate an authentication scheme that it does support. SyncML servers and SyncML clients that do not support this use case are not required to provide further response to the SyncML entity that is issuing the **Status** on a **Status**.

Content Model: (CmdID, MsgRef, CmdRef, Cmd, Data, Item?+, SourceRef, TargetRef)

Attributes: None.

2.2.7   Protocol Command Elements

Protocol command element types supply the SyncML commands implemented in MDM.

2.2.7.1   Add

The **Add** element specifies the SyncML command to add data items to a data collection. The element has the following syntax:

```
<Add>(CmdID, Meta?, Item+)</Add>
```

Parent Elements: [Atomic (section 2.2.7.3)](#), [SyncBody (section 2.2.4.3)](#)

Restrictions: The [CmdID (section 2.2.3.2)](#) element type is required and specifies the unique identifier for the SyncML message for the command.

- The [Meta (section 2.2.5.3)](#) element is optional. When Meta is present, it specifies the meta-information to be used for the command.

- One or more [Item (section 2.2.5.2)](#) element types MUST be specified. The **Item** elements contain the data items to add to the data collection.

Content Model: (**CmdID**, **Meta**?, **Item**+)

Attributes: None.

2.2.7.2   Alert

The **Alert** element specifies the SyncML command to send custom content information to the recipient. **Alert** provides a mechanism for communicating content information, such as state information or notifications to an application on the recipient device. Use of the **Alert** command also

provides a standard method for specifying non-standard extended commands beyond those defined in this document. The element has the following syntax:

```
<Alert>(CmdID, Data)</Alert>
```

Parent Elements: Atomic (section 2.2.7.3), SyncBody (section 2.2.4.3)

Restrictions: The CmdID (section 2.2.3.2) element type is required and specifies the unique identifier for the SyncML message for the command.

▪ The Data (section 2.2.5.1) element specifies the custom content to send to the recipient.

Content Model: (CmdID, Data)

Attributes: None.

2.2.7.3 Atomic

The **Atomic** element specifies the SyncML command to request that subordinate commands be executed as a set or not at all. The element has the following syntax:

```
<Atomic>(CmdID, Meta?, (Add | Delete | Atomic | Replace | Get | Exec)+)</Atomic>
```

Parent Elements: SyncBody (section 2.2.4.3)

Restrictions: The CmdID (section 2.2.3.2) element type is required and specifies the unique identifier for the command in the SyncML message.

▪ The Meta (section 2.2.5.3) element is optional. When Meta is present, it specifies meta-information to be used for the command. The scope of the meta-information is limited to the command.

▪ One or more sets of subordinate commands to be executed are specified as a series of one or more of the following commands: Add (section 2.2.7.1), Delete (section 2.2.7.4), **Atomic**, Replace (section 2.2.7.7), Get (section 2.2.7.6), and Exec (section 2.2.7.5). Note that a particular command can be specified multiple times to achieve execution of contained commands. For example:

```
<Atomic>(CmdID, Meta,(Add,Get),(Delete,Replace),(Add,Exec))</Atomic>
```

Content Model: (CmdID, Meta?, (Add| Delete | **Atomic** | Replace | Get | Exec)+)

Attributes: None.

2.2.7.4 Delete

The **Delete** element specifies the SyncML command to delete data items from a data collection. The element has the following syntax:

```
<Delete>(CmdID, Meta?, Item+)</Delete>
```

Parent Elements: Atomic (section 2.2.7.3), SyncBody (section 2.2.4.3)

Restrictions: The CmdID (section 2.2.3.2) element type is required and specifies the unique identifier for the command in the SyncML message.

- The Meta (section 2.2.5.3) element is optional. When Meta is present, it specifies the meta-information to be used for the command.

- One or more Item (section 2.2.5.2) element types MUST be specified. The Item elements contain the data items to delete from the data collection.

Content Model: (CmdID, Meta?, Item+)

Attributes: None.

2.2.7.5 Exec

The **Exec** element specifies the WMI class method to execute on the recipient's device. The element has the following syntax:

```
<Exec>(CmdID, Meta?, Item)</Exec>
```

Parent Elements: SyncBody (section 2.2.4.3), Atomic (section 2.2.7.3)

Restrictions: The CmdID (section 2.2.3.2) element type is required and specifies the unique identifier for the command in the SyncML message.

- The Meta (section 2.2.5.3) element is optional. When Meta (section 2.2.5.3) is present, it specifies the meta-information to be used for the command.

- The Item (section 2.2.5.2) element type specifies the WMI class method to execute on the recipient's device.

Content Model: (CmdID, Meta?, Item)

Attributes: None.

2.2.7.6 Get

The **Get** element specifies the SyncML command to retrieve data from the recipient. The element has the following syntax:

```
<Get>(CmdID, Meta?, Item+)</Get>
```

Parent Elements: SyncBody (section 2.2.4.3), Atomic (section 2.2.7.3)

Restrictions: The CmdID (section 2.2.3.2) element type is required and specifies the unique identifier for the command in the SyncML message.

- The Meta (section 2.2.5.3) element is optional. When Meta is present, it specifies the meta-information to be used for the command.

- One or more Item (section 2.2.5.2) element types MUST be specified. The Item elements contain the data items to retrieve from the recipient.

Content Model: (CmdID, Meta?, Item+)

Attributes: None.

2.2.7.7 Replace

The **Replace** element specifies the SyncML command to replace data items. The element has the following syntax:

```
<Replace>(CmdID, Meta?, Item+)</Replace>
```

Parent Elements: Atomic (section 2.2.7.3), SyncBody (section 2.2.4.3)

Restrictions: The CmdID (section 2.2.3.2) element type is required and specifies the unique identifier for the command in the SyncML message.

- The Meta (section 2.2.5.3) element is optional. When Meta is present, it specifies the meta-information to be used for the command.

- One or more Item (section 2.2.5.2) element types MUST be specified. The Item elements contain the data items to replace.

Content Model: (**CmdID**, **Meta**?, **Item**+)

Attributes: None.

2.2.7.8 Results

The **Results** element specifies the SyncML command to return the results of a Get (section 2.2.7.6) command. The element has the following syntax:

```
<Results>(CmdID, MsgRef?, CmdRef, Cmd, Meta?, Item+)</Results>
```

Parent Elements: SyncBody (section 2.2.4.3)

Restrictions: The CmdID (section 2.2.3.2) element type is required and specifies the unique identifier for the command in the SyncML message.

- The MsgRef (section 2.2.3.7) element is optional. When MsgRef is present, it specifies the MsgID of the associated SyncML request from the originator. When MsgRef is not present in a **Results** element type, MsgRef MUST be processed as having a value of "1".

- The CmdRef (section 2.2.3.3) element type specifies the CmdID of the associated SyncML request from the originator. When CmdRef is not present in a **Results** element type, the response status code is associated with a CmdID value of `"1"`.

- The Cmd (section 2.2.3.1) element type specifies the Cmd of the associated SyncML request from the originator.

- One or more Item (section 2.2.5.2) element types MUST be specified to contain the results.

- The Source (section 2.2.3.9) element specified within the Item element SHOULD be a relative URI corresponding to the Get (section 3.1.5.1.6) command sent by the originator.

- The LocURI (section 2.2.3.5) element specified within the Item element SHOULD be a property corresponding to the **Get** command sent by the originator.

- The Data (section 2.2.5.1) element specified within the Item element SHOULD be the value for the property specified in the LocURI element.

- The Meta (section 2.2.5.3) element is optional. When Meta is present, it specifies the 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.

Content Model: (CmdID, MsgRef, CmdRef, Cmd, Meta?, Item+)

Attributes: None.

3 Protocol Details

MDM is based on a subset of the OMA-DM protocol (OMA-TS-DM_Protocol-V1_2_1-20080617-A) [OMA-DMP1.2.1]. SyncML messages issued by the client to the server and from the server to the client are defined in section 3.1.5 and are a subset of the SyncML messages defined in the SyncML Representation Protocol (OMA-TS-SyncML-RepPro-V1_2_2-20090724-A) [OMA-SyncMLRP1.2.2]. MDM does not modify or extend any elements defined in [OMA-SyncMLRP1.2.2].

Device Management Session

As noted earlier, MDM is a client/server protocol. The client device MUST establish a device management session with the server by supporting the **Package 1: Initialization from Client to Server** method defined in [OMA-DMP1.2.1] section 8.3.

After a session has been established, the server MAY issue SyncML commands indicating operations to perform against WMI classes on the client device.

The client always initiates the conversation by transmitting SyncML messages to the server via an HTTP POST. The server response to client commands, as well as other commands issued to the client, are contained in the HTTP response associated to the POST request.

An example of a short OMA-DM session is shown in the following figure.
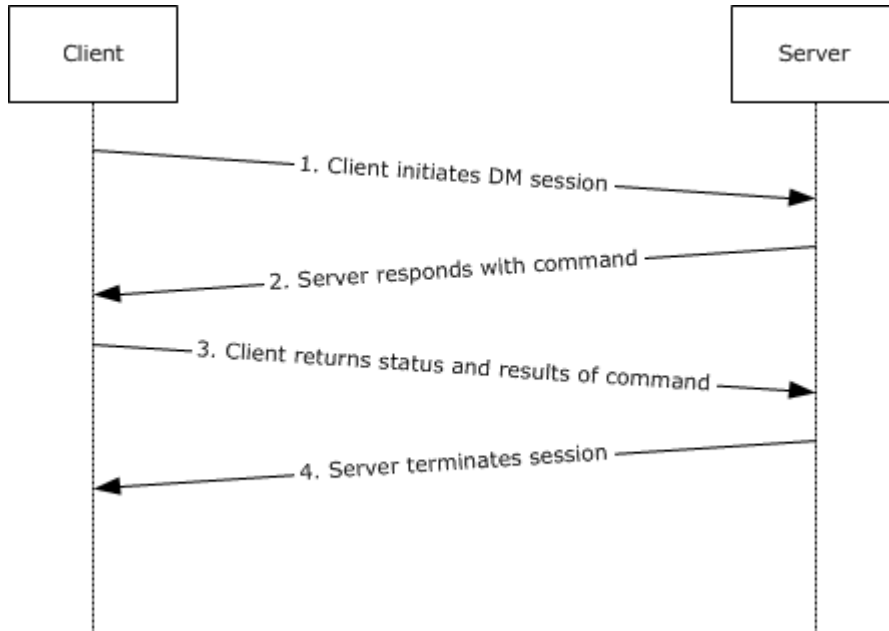
Figure 2: Example of an OMA-DM session

SyncML Message Document

As described in section 2.2.2, a SyncML message is a well-formed XML document. OMA-DM commands are transmitted between the server and the client device within SyncML messages (or, within XML documents). A SyncML message MUST contain a SyncHdr (section 2.2.4.2) element and MAY contain a SyncBody (section 2.2.4.3) element. A SyncML message MAY contain a SyncML command (see section 3.1.5).

The structure and content of a SyncML message is defined in section 2.2.2 and is a subset of the SyncML message definition specified in [OMA-SyncMLRP1.2.2]. Each message is composed of a header specified by the SyncHdr (section 2.2.4.2) element, and a message body specified by the SyncBody (section 2.2.4.3) element.

The following example demonstrates the general structure of a SyncML message document issued by the server (the Source (section 2.2.3.9)) to a specified client (the Target (section 2.2.3.11)).

```
<SyncML xmlns='SYNCML:SYNCML1.2'>
  <SyncHdr>
    <VerDTD>1.2</VerDTD>
      <VerProto>DM/1.2</VerProto>
        <SessionID>1</SessionID>
        <MsgID>1</MsgID>
        <Target>
         <LocURI>{unique device ID}</LocURI>
        </Target>
        <Source>
          <LocURI>https://contoso.com/management-server</LocURI>
        </Source>
  </SyncHdr>
  <SyncBody>
    <!-- Query for the name of the device -->
    <Get>
      <CmdID>2</CmdID>
      <Item>
        <Target>
          <LocURI>./cimv2/Win32_ComputerSystem/Win32_ComputerSystem.Name
          </LocURI>
        </Target>
      </Item>
    </Get>
    <Final />
  </SyncBody>
</SyncML>
```

The commands that MAY be issued by a requester are specified in section 3.1.5.1.

The commands that MAY be issued in response to a request are specified in section 3.1.5.2.

3.1   Common Details

3.1.1   Abstract Data Model

None.

3.1.2   Timers

None.

3.1.3   Initialization

None.

Release: Friday, October 25, 2013

3.1.4   Higher-Layer Triggered Events

None.

3.1.5   Message Processing Events and Sequencing Rules

3.1.5.1   SyncML Request Commands

The following sections specify the request commands for the SyncML message.

3.1.5.1.1   Add

The **Add** command is used by the originator to request that data elements supplied by the originator be added to data items that are accessible to the recipient.

The following XML snippet shows example usage of the **Add** command. The syntax for the **Add** element type is specified in section 2.2.7.1.

```
<SyncBody>
  <Add>
    <CmdID>1</CmdID>
    <Item>
      <Target>
        <LocURI>./cimv2/MDM_Certificate/MDM_Certificate.
            Key=%221%22/Blob</LocURI>
      </Target>
      <Meta>
        <Format xmlns="syncml:metinf">chr</Format>
      <Meta>
      <Data>{base-64 encoded raw certificate blob}</Data>
    </Item>
  </Add>
  ...
</SyncBody>
```

The following table lists the classes that are valid for use with the **Add** command.

Class reference	LocURI	Description
[MDM_AppInstallJob]	./cimv2/MDM_AppInstallJob	A class defined by the MDM protocol (see section 6) for installing applications to a device.
[MDM_Certificate]	./cimv2/MDM_Certificate	A class defined by the MDM protocol (see section 6) for managing per-application enterprise settings.
[MDM_Certificate Enrollment]	./cimv2/MDM_CertificateEnrollment	A class defined by the MDM protocol (see section 6) for provisioning certificates to the device.
[MDM_Client]	./cimv2/MDM_Client	A class defined by the MDM protocol (see section 6) that provides identification for a user/device pair. The class also provides methods for

Class reference	LocURI	Description
		unenrolling the device from management and device lock.
[MDM_ConfigSetting]	./cimv2/MDM_ConfigSetting	A class defined by the MDM protocol (see section 6) that provides the capability to get and set configuration settings for the OMA-DM agent. When targeted in the Add command, additional configuration settings can be added to the device.
[MDM_MgmtAuthority]	./cimv2/MDM_MgmtAuthority	A class defined by the MDM protocol (see section 6) for managing management endpoints. The first management endpoint is configured on the device during enrollment.
[MDM_Restrictions]	./cimv2/MDM_Restrictions	A class defined by the MDM protocol (see section 6) that provides configuration for restricting settings on the device.
[MDM_VpnApplication Trigger]	./cimv2/MDM_VpnApplicationTrigger	A class defined by the MDM protocol (see section 6) for associating an application to a VPN connection.
[MDM_WirelessProfile]	./cimv2/MDM_WirelessProfile	A class defined by the MDM protocol (see section 6) for provisioning a wireless profile to the device.

3.1.5.1.2 Alert

The **Alert** command is used by the originator to send custom content information to the recipient. The client or originator MUST send the **Alert** command as part of the SyncML device management session initialization message.

The following XML snippet shows example usage of the **Alert** command. The syntax for the **Alert** element type is specified in section 2.2.7.2.

```
<SyncBody>
  <Alert>
    <CmdID>1</CmdID>
    <Data>1201</Data> <!-- client-initiated session -->
  </Alert>
  ...
</SyncBody>
```

No classes are specified for the **Alert** command.

3.1.5.1.3 Atomic

The **Atomic** command is used by the client to start initialization of a session. If execution fails for any of the commands contained within the atomic unit, all previously executed commands within the

unit MUST be rolled back to their previous state. All commands within the atomic unit MUST be valid for any of the commands within the unit to be executed.

The following XML snippet shows example usage of the **Atomic** command. In the example, the originator issues two Replace (section 3.1.5.1.7) commands inside of the **Atomic** command. The syntax for the **Atomic** element type is specified in section 2.2.7.3.

```
<SyncBody>
  <Atomic>
    <CmdID>10</CmdID>
    <Replace>
      <CmdID>8</CmdID>
      <Item>
        <Target>
          <LocURI>./cimv2/MDM_ConfigSetting/MDM_ConfigSetting.
            SettingName=%22UnEnrollOnCertExpiry%22/SettingValue</LocURI>
        </Target>
        <Data>true</Data>
      </Item>
    </Replace>
    <Replace>
      <CmdID>9</CmdID>
      <Item>
        <Target>
          <LocURI>./cimv2/MDM_Restrictions/MDM_Restrictions.
            Key=%221%22/DataRoamingEnabled</LocURI>
        </Target>
        <Data>true</Data>
      </Item>
    </Replace>
  </Atomic>
  ...
</SyncBody>
```

No classes are specified for the **Atomic** command. Valid classes for the commands specified within an atomic unit are specified in the definition for each command in this section. For example, the valid classes for the Delete command are as specified in the definition for the Delete command in section 3.1.5.1.4.

3.1.5.1.4 Delete

The **Delete** command is used by the originator to request that data elements accessible to the recipient be deleted. The target location URI contains the **resource** to delete and is specified in the following format:

Node: <LocURI>{class-instance}</LocURI>

Property: <LocURI>{class-instance}/{property-name}</LocURI>

The following XML snippet shows example usage of the Delete command. The syntax for the Delete element type is specified in section 2.2.7.4.

```
<SyncBody>
  <Delete>
```

```
        <CmdID>1</CmdID>
        <Item>
          <Target>
            <LocURI>./cimv2/MDM_Certificate/MDM_Certificate.
                Key=%221%22</LocURI>
          </Target>
        </Item>
      </Delete>
      ...
    </SyncBody>
```

The following table lists the classes that are valid for use with the **Delete** command.

Class reference	LocURI	Description
[MDM_AppInstallJob]	./cimv2/MDM_AppInstallJob	A class defined by the MDM protocol (see section 6) for managing and installing side loaded applications on the device.
[MDM_ApplicationSetting]	./cimv2/MDM_ApplicationSetting	A class defined by the MDM protocol (see section 6) for managing per-application enterprise settings.
[MDM_Browser SecurityZones]	./cimv2/MDM_BrowserSecurityZones	A class defined by the MDM protocol (see section 6) for managing certificates provisioned on the device using MDM. The Delete command can use this class to remove certificates previously provisioned using.
[MDM_Certificate]	./cimv2/MDM_Certificate	A class defined by the MDM protocol (see section 6) that specifies the certificate installation provider.
[MDM_Certificate Enrollment]	./cimv2/MDM_CertificateEnrollment	A class defined by the MDM protocol (see section 6) that specifies the certificate enrollment provider.
[MDM_VpnApplication Trigger]	./cimv2/MDM_VpnApplicationTrigger	A class defined by the MDM protocol (see section 6) for associating an application to a VPN profile. When the application launches, the VPN connection is established. The Delete command can use this class for removing an application/VPN association.

3.1.5.1.5 Exec

The **Exec** command is used by the originator to target a static method contained in a WMI class which is then invoked by the recipient. The target LocURI (section 2.2.3.5) element contains the method to invoke and is specified in the following format:

```
    <LocURI>{class-instance}/Exec={method-name}</LocURI>
```

The Data (section 2.2.5.1) element contains the names and values for any parameters:

```
<Data>{parameter-name}={parameter-value}&{parameter-name}={parameter-value}...</Data>
```

The following XML snippet shows example usage of the **Exec** command. In the example, the originator calls the SetValues static method on the MDM_EASPolicy WMI class to set the *MinPasswordLength* parameter to a value of 5. The syntax for the **Exec** element type is specified in section 2.2.7.5.

```
<SyncBody>
  <Exec>
    <CmdID>11</CmdID>
    <Item>
      <Target>
        <LocURI>./cimv2/MDM_EASPolicy/MDM_EASPolicy.
            Key=%221%22/Exec=SetValues
        </LocURI>
      </Target>
      <Meta>
        <Format xmlns="syncml:metinf">chr</Format>
        <Type xmlns="syncml:metinf">text/plain</Type>
      </Meta>
      <Data>NamedValuesList=MinPasswordLength,5;</Data>
    </Item>
  </Exec>
  ...
</SyncBody>
```

The following table lists the classes that are valid for use with the **Exec** command.

Class reference	LocURI	Description
Class reference	LocURI	Description
[MDM_AppInstallJob]	./cimv2/MDM_AppInstallJob	A class defined by the MDM protocol (see section 6) for installing side loaded applications on the device.
[MDM_MgmtAuthority]	./cimv2/MDM_MgmtAuthority	A class defined by the MDM protocol (see section 6) for managing management endpoints. The Exec command can use this class to install additional management endpoints to the device.
[MDM_Client]	./cimv2/MDM_Client	A class defined by the MDM protocol (see section 6) that specifies the user/device association. The server can issue an Exec command and target this class to lock the device and unroll the device from management.
[MDM_SideLoader]	./cimv2/MDM_SideLoader]	A class defined by the MDM protocol (see section 6) for managing application side loading keys. The server can issue an

Class reference	LocURI	Description
		Exec command and target this class to install side loading keys on the device.
[MDM_WNSConfiguration]	./cimv2/MDM_WNSConfiguration	A class defined by the MDM protocol (see section 6) for configuring push notifications to the agent. The server supplies the application ID and PFN to the client. The client establishes a unique push notification channel **URI** with WNS. The channel URI is sent to the server by the client via a Replace (section 3.1.5.1.7) command.

3.1.5.1.6 Get

The **Get** command is used by the originator to request data elements from the recipient. The **Get** command can include the resetting of any meta-information that the recipient maintains about the data element or data collection.

Get commands are issued by the server to the client to retrieve property information, such as processor architecture. The LocURI (section 2.2.3.5) element contains the class/property which the **Get** command is specified against. Multiple **Get** commands, each with a unique CmdID (section 2.2.3.2), MAY exist in a single message. For each **Get** command issued by the server, the client MUST respond with Status (section 2.2.6.1) command. If the **Get** command completes successfully with the result of 200, the client MUST also respond with a Results (section 2.2.7.8) command.

The format for obtaining the class instance (against the **Get** command) is defined as follows:

```
<LocURI>./cimv2/{class-name}</LocURI>
```

The format for obtaining a property on the class instance is defined as follows:

```
<LocURI>{class-instance}/{property-name}
```

The following XML snippet shows example usage of the **Get** command. The syntax for the **Get** element type is specified in section 2.2.7.6.

```
<SyncBody>
  <Get>
    <CmdID>1</CmdID>
    <Item>
      <Target>
        <LocURI>./cimv2/MDM_Client/MDM_Client.
            DeviceClient=0/DeviceName</LocURI>
      </Target>
    </Item>
  </Get>
  ...
</SyncBody>
```

The following table lists the classes that are valid for use with the **Get** command.

Class reference	LocURI	Description
[DevInfo]	./DevInfo	An abstract class defined in [OMA-StdObj1.2.1] that describes device information, such as the ID, manufacturer, and model.
[MDM_AppInstallJob]	./cimv2/MDM_AppInstallJob	A class defined by the MDM protocol (see section 6) for installing and discovering applications.
[MDM_Application]	./cimv2/MDM_Application	A class defined by the MDM protocol (see section 6) for discovering and deleting modern applications.
[MDM_ApplicationFramework]	./cimv2/MDM_ApplicationFramework	A class defined by the MDM protocol (see section 6) for discovering and deleting modern application frameworks.
[MDM_ApplicationSetting]	./cimv2/MDM_ApplicationSetting	A class defined by the MDM protocol (see section 6) for managing per-application enterprise settings.
[MDM_BrowserSecurityZones]	./cimv2/MDM_BrowserSecurityZones	A class defined by the MDM protocol (see section 6) that provides configuration to modify browser security zone settings on the device.
[MDM_BrowserSettings]	./cimv2/MDM_BrowserSettings	A class defined by the MDM protocol (see section 6) that provides configuration to modify browser security settings on the device.
[MDM_Certificate]	./cimv2/MDM_Certificate	A class defined by the MDM protocol (see section 6) for managing provisioned certificates on the device.
[MDM_CertificateEnrollment]	./cimv2/MDM_CertificateEnrollment	A class defined by the MDM protocol (see section 6) for managing and provisioning

Class reference	LocURI	Description
		certificates on a device.
[MDM_Client]	./cimv2/MDM_Client	A class defined by the MDM protocol (see section 6) that provides identification for a user/device pair. The class also provides methods for unenrolling the device from management and device lock.
[MDM_ConfigSetting]	./cimv2/MDM_ConfigSetting	A class defined by the MDM protocol (see section 6) for getting and setting configuration settings for the OMA-DM agent.
[MDM_DeviceRegistrationInfo]	./cimv2/MDM_DeviceRegistrationInfo	A class defined by the MDM protocol (see section 6) that provides Active Directory device registration information.
[MDM_EASPolicy]	./cimv2/MDM_EASPolicy	A class defined by the MDM protocol (see section 6) that provides the EAS PIN Password Policy configuration on the device. A Get command targeting a property in this class returns the configured setting.
[MDM_MgmtAuthority]	./cimv2/MDM_MgmtAuthority	A class defined by the MDM protocol (see section 6) for managing management endpoints.
[MDM_RemoteAppUserCookie]	./cimv2/MDM_RemoteAppUserCookie	A class defined by the MDM protocol (see section 6) for managing remote application cookies.
[MDM_RemoteApplication]	./cimv2/MDM_RemoteApplication	A class defined by the MDM protocol (see section 6) for discovering, updating, and deleting remote applications.
[MDM_Restrictions]	./cimv2/MDM_Restrictions	A class defined by the MDM protocol (see section 6) that provides

Class reference	LocURI	Description
		configuration for restricting settings on the device. The server can issue a Get command and target a restriction property in the class to retrieve the current setting.
[MDM_SecurityStatus]	./cimv2/MDM_SecurityStatus	A class defined by the MDM protocol (see section 6) that provides security health metrics on the device.
[MDM_SideLoader]	./cimv2/MDM_SideLoader]	A class defined by the MDM protocol (see section 6) for managing application side loading keys on the device. The server can issue a Get command and target a property in this class to retrieve the current configured setting.
[MDM_VpnApplicationTrigger]	./cimv2/MDM_VpnApplicationTrigger	A class defined by the MDM protocol (see section 6) for managing application/VPN trigger associations.
[MDM_WebApplication]	./cimv2/MDM_WebApplication	A class defined by the MDM protocol (see section 6) for discovering and deleting installed web applications.
[MDM_WirelessProfile]	./cimv2/MDM_WirelessProfile	A class defined by the MDM protocol (see section 6) for managing wireless profiles on the device. The server can issue a Get command and target this class to return the current setting of a property in the class.
[MDM_WirelessProfileXml]	./cimv2/MDM_WirelessProfileXml	A class defined by the MDM protocol (see section 6) for managing wireless profiles. The server can issue a Get command and target this class to return the current setting of a

Class reference	LocURI	Description
		property in the class.
[MDM_WNSChannel]	./cimv2/MDM_WNSChannel	A class defined by the MDM protocol (see section 6) for retrieving the current push notification channel on the client.
[MDM_WNSConfiguration]	./cimv2/MDM_WNSConfiguration	A class defined by the MDM protocol (see section 6) for configuring push notification to the client agent. The server can issue a Get command and target a property in this class to return the current setting.
[SoftwareLicensingProduct]	./cimv2/SoftwareLicensingProduct	A class that exposes the product-specific properties and methods of the Software Licensing service. For more information about this class, see [MSDN-SwLicProduct].
[SoftwareLicensingService]	./cimv2/SoftwareLicensingService	A class that exposes the product-independent properties and methods of the Software Licensing service. For more information about this class, see [MSDN-SwLicService].
[Win32_Battery]	./cimv2/Win32_Battery	A WMI class that represents a battery connected to the computer system. For more information about this class, see [MSDN-Win32_Battery].
[Win32_BIOS]	./cimv2/Win32_BIOS	A WMI class that represents the attributes of the computer system's basic input/output services (BIOS) that are installed on a computer. For more information about this class, see [MSDN-Win32_BIOS].
[Win32_ComputerSystem]	./cimv2/Win32_ComputerSystem	Represents a computer

Class reference	LocURI	Description
		system running Windows. For more information about this class, see [MSDN-Win32_CmptrSys].
[Win32_ComputerSystemProduct]	./cimv2/Win32_ComputerSystemProduct	A WMI class that represents a product. This includes software and hardware used on this computer system. For more information about this class, see [MSDN-Win32_CmptrSysProd].
[Win32_CurrentTime]	./cimv2/Win32_CurrentTime	A WMI class that describes a point in time by using date and hourly components, such as milliseconds, seconds, minutes, hours, days, days of the week, week in the month, months, quarters, or years. For more information about this class, see [MSDN-Win32_CurrentTime].
[Win32_DesktopMonitor]	./cimv2/Win32_DesktopMonitor	A WMI class that represents the type of monitor or display device attached to the computer system. For more information about this class, see [MSDN-Win32_DsktpMntr].
[Win32_DiskDrive]	./cimv2/Win32_DiskDrive	A WMI class that represents a physical disk drive as recognized by a computer running Windows. For more information about this class, see [MSDN-Win32_DiskDrive].
[Win32_DisplayConfiguration]	./cimv2/Win32_DisplayConfiguration	A WMI class that represents configuration information for the display device on a computer running on Windows.<1> For more information about this class, see [MSDN-Win32_DisplayCnfg].

Release: Friday, October 25, 2013

Class reference	LocURI	Description
[Win32_EncryptableVolume]	./cimv2/Win32_EncryptableVolume	A WMI provider class that represents an area of storage on a hard disk that can be protected by using BitLocker Drive Encryption. For more information about this class, see [MSDN-Win32_EncryptVol].
[Win32_InfraredDevice]	./cimv2/Win32_InfraredDevice	A WMI class that represents the capabilities and management of an infrared device. For more information about this class, see [MSDN-Win32_InfrdDevice].
[Win32_LocalTime]	./cimv2/Win32_LocalTime	A WMI class that describes a point in time returned as Win32_LocalTime objects resulting from a query. For more information about this class, see [MSDN-Win32_LocalTime].
[Win32_LogicalDisk]	./cimv2/Win32_LogicalDisk	A WMI class that represents a data source that resolves to an actual local storage device on a computer system running Windows. For more information about this class, see [MSDN-Win32_LogicalDisk].
[Win32_NetworkAdapter]	./cimv2/Win32_NetworkAdapter	A WMI class that represents a network adapter of a computer running Windows. For more information about this class, see [MSDN-Win32_NtwkAdptr].
[Win32_NetworkAdapterConfiguration]	./cimv2/Win32_NetworkAdapterConfiguration	A WMI class that represents the attributes and behaviors of a network adapter. For more information about this class, see [MSDN-Win32_NtwkAdptrCnfg].

Release: Friday, October 25, 2013

Class reference	LocURI	Description
[Win32_OperatingSystem]	./cimv2/Win32_OperatingSystem	A WMI class that represents a Windows-based operating system installed on a computer. For more information about this class, see [MSDN-Win32_OpSys].
[Win32_PhysicalMemory]	./cimv2/Win32_PhysicalMemory	A WMI class that represents a physical memory device located on a computer system and available to the operating system. For more information about this class, see [MSDN-Win32_PhysMemory].
[Win32_PnPDevice]	./cimv2/Win32_PnPDevice	A WMI class that relates a device (known to the Configuration Manager as a PNPEntity) to the function it performs. The function is represented by a subclass of the logical device that describes the use of the device. For more information about this class, see [MSDN-Win32_PnPDevice].
[Win32_PortableBattery]	./cimv2/Win32_PortableBattery	A WMI class that contains the properties related to a portable battery, such as a notebook computer battery. For more information about this class, see [MSDN-Win32_PortBattery].
[Win32_Processor]	./cimv2/Win32_Processor	A WMI class that represents a device that can interpret a sequence of instructions on a computer running Windows. On a multiprocessor computer, one instance of the Win32_Processor class exists for each processor. For more information about this class, see [MSDN-Win32_Processor].

Class reference	LocURI	Description
[Win32_QuickFixEngineering]	./cimv2/Win32_QuickFixEngineering	A WMI class that represents a small system-wide update commonly referred to as a quick-fix engineering (QFE) update that is applied to the current operating system. For more information about this class, see [MSDN-Win32_QFE].
[Win32_Service]	./cimv2/Win32_Service	A WMI class that represents a service on a computer system running Windows. For more information about this class, see [MSDN-Win32_Service].
[Win32_Share]	./cimv2/Win32_Share	A class that represents a shared resource on a computer system running Windows. This may be a disk drive, printer, interprocess communication, or other sharable device. For more information about this class, see [MSDN-Win32_Share].
[Win32_SystemBIOS]	./cimv2/Win32_SystemBIOS	A WMI class that relates a computer system (including data, such as startup properties, time zones, boot configurations, or administrative passwords) with a system BIOS (services, languages, and system management properties). For more information about this class, see [MSDN-Win32_SysBIOS].
[Win32_SystemEnclosure]	./cimv2/Win32_SystemEnclosure	A WMI class that represents the properties associated with a physical system enclosure. For more information about this class, see [MSDN-Win32_SysEnclose].

Class reference	LocURI	Description
[Win32_TimeZone]	./cimv2/Win32_TimeZone	A WMI class that represents the time zone information for a computer system running Windows, which includes the changes required for transitioning to daylight savings time. For more information about this class, see [MSDN-Win32_TimeZone].
[Win32_Tpm]	./cimv2/Win32_Tpm	A class that represents the Trusted Platform Module (TPM), which is a hardware security chip that provides the root of trust for a computer system. For more information about this class, see [MSDN-Win32_Tpm].
[Win32_UTCTime]	./cimv2/Win32_UTCTime	A WMI class that describes a point in time that is returned as Win32_UTCTime objects resulting from a query. For more information about this class, see [MSDN-Win32_UTCTime].
[Win32_WindowsUpdateAgentVersion]	./cimv2/Win32_WindowsUpdateAgentVersion	The GetInfo method of the IWindowsUpdateAgentInfo interface is used to retrieve version information about Windows Update Agent (WUA). For more information about this class, see [MSDN-IWUAInfo_GetInfo].

3.1.5.1.7   Replace

The **Replace** command is used by the originator to request that data elements accessible to the recipient be replaced. The **Replace** command makes a complete replacement of the specified data elements. If a specified element does not already exist on the recipient, an error is returned to the originator.

The following XML snippet shows example usage of the **Replace** command. The syntax for the **Replace** element type is specified in section 2.2.7.8.

```
<SyncBody>
  <Replace>
    <CmdID>8</CmdID>
    <Item>
      <Target>
        <LocURI>./cimv2/MDM_ConfigSetting/MDM_ConfigSetting.
            SettingName=%22UnEnrollOnCertExpiry%22/SettingValue</LocURI>
      </Target>
      <Data>true</Data>
    </Item>
  </Replace>
  ...
</SyncBody>
```

The following table lists the classes that are valid for use with the **Replace** command.

Class reference	LocURI	Description
[DevInfo]	./DevInfo	An abstract class defined in [OMA-StdObj1.2.1] for addressing common device information, such as the ID, manufacturer, and model.
[MDM_AppInstallJob]	./cimv2/MDM_AppInstallJob	A class defined by the MDM protocol (see section 6) for managing side loaded applications.
[MDM_ApplicationSetting]	./cimv2/MDM_ApplicationSetting	A class defined by the MDM protocol (see section 6) for managing per-application enterprise settings. The server can issue a Replace command to update an existing setting.
[MDM_BrowserSecurityZones]	./cimv2/MDM_BrowserSecurityZones	A class defined by the MDM protocol (see section 6) that provides configuration to modify browser security zone settings on the device.
[MDM_Certificate]	./cimv2/MDM_Certificate	A class defined by the MDM protocol (see section 6) for managing provisioned certificates on the device.
[MDM_CertificateEnrollment]	./cimv2/MDM_CertificateEnrollment	A class defined by the MDM protocol (see section 6) that specifies the certificate enrollment provider.
[MDM_Client]	./cimv2/MDM_Client	A class defined by the MDM protocol (see section 6) that provides identification for a user/device pair. The class also provides methods for unenrolling

Class reference	LocURI	Description
		the device from management and device lock.
[MDM_ConfigSetting]	./cimv2/MDM_ConfigSetting	A class defined by the MDM protocol (see section 6) that provides the capability to get and set configuration settings for the OMA-DM agent.
[MDM_MgmtAuthority]	./cimv2/MDM_MgmtAuthority	A class defined by the MDM protocol (see section 6) for managing management endpoints. The server can issue a Replace command to update and existing management endpoint on the device.
[MDM_Restrictions]	./cimv2/MDM_Restrictions	A class defined by the MDM protocol (see section 6) that provides configuration for restricting settings on the device.
[MDM_SecurityStatus]	./cimv2/MDM_SecurityStatus	A class defined by the MDM protocol (see section 6) that provides security health metrics on the device.
[MDM_VpnApplicationTrigger]	./cimv2/MDM_VpnApplicationTrigger	A class defined by the MDM protocol (see section 6) for managing application/VPN trigger associations.
[MDM_WirelessProfile]	./cimv2/MDM_WirelessProfile	A class defined by the MDM protocol (see section 6) for managing wireless profiles on the device. The server can issue a Replace command and target this class to update an existing wireless profile.

3.1.5.2   SyncML Response Commands

The following sections specify the response commands for the SyncML message.

3.1.5.2.1   Status

The **Status** response command MUST be returned by the client in response to any command issued by the server. Multiple **Status** response commands, each with a unique CmdID (section 2.2.3.2), MAY exist in a single message.

The following XML snippet shows example usage of the **Status** response command. The syntax for the **Status** element type is specified in section 2.2.6.1.

```
<SyncBody>
  <Status>
```

```
      <CmdID>1</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>5</CmdRef>
      <Cmd>Get</Cmd>
      <Data>200</Data>
    </Status>
    ...
  </SyncBody>
```

3.1.5.2.2 Results

The **Results** response command MUST be returned by the recipient in response to each successful Get (section 3.1.5.1.6) command issued by the originator. Multiple **Results** response commands, each with a unique CmdID (section 2.2.3.2), MAY exist in a single SyncBody (section 2.2.4.3) element.

The following XML snippet shows example usage of the **Results** response command. The syntax for the **Results** element type is specified in section 2.2.7.8.

```
  <SyncBody>
    <Results>
      <CmdID>1</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>5</CmdRef>
      <Cmd>Get</Cmd>
      <Item>
        <Source>
          <LocURI>{wmi-class-instance/property}</LocURI>
        </Source>
        <Meta>
          <Format xmlns="syncml:metinf">chr</Format>
        </Meta>
        <Data>{property data}</Data>
      </Item>
    </Results>
    ...
  </SyncBody>
```

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4   Protocol Examples

The following example is a complete message exchange between a client and server.

```
// The client issues a DM session initialization SyncML message to
// the management server.  The message includes an Alert command
// and a Replace command along with information about the device.

<SyncML xmlns="SYNCML:SYNCML1.2">

  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.2</VerProto>
    <SessionID>1</SessionID>
    <MsgID>1</MsgID>
    <Target>
      <LocURI>https://contoso.com/management-server</LocURI>
    </Target>
    <Source>
      <LocURI>{unique device id}</LocURI>
    </Source>
  </SyncHdr>

  <SyncBody>
    <Alert>
      <CmdID>2</CmdID>
      <Data>1201</Data>
    </Alert>
    <Replace>
      <CmdID>3</CmdID>
      <Item>
        <Source>
          <LocURI>./DevInfo/DevId</LocURI>
        </Source>
        <Data>{unique device id}</Data>
      </Item>
      <Item>
        <Source>
          <LocURI>./DevInfo/Man</LocURI>
        </Source>
        <Data>Microsoft Corporation</Data>
      </Item>
      <Item>
        <Source>
          <LocURI>./DevInfo/Mod</LocURI>
        </Source>
        <Data>{operating system description}</Data>
      </Item>
      <Item>
        <Source>
          <LocURI>./DevInfo/DmV</LocURI>
        </Source>
        <Data>1.2</Data>
      </Item>
      <Item>
        <Source>
          <LocURI>./DevInfo/Lang</LocURI>
```

```
          </Source>
          <Data>en-US</Data>
        </Item>
      </Replace>
      <Final />
    </SyncBody>
</SyncML>

// The server responds with the required Status command for the SyncHdr and
// Status commands for the requested Alert and Replace commands.  The server
// requests more information from the client with a series of Get commands.

<?xml version="1.0" encoding="utf-8" ?>
<SyncML xmlns="SYNCML:SYNCML1.2">

  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.2</VerProto>
    <SessionID>1</SessionID>
    <MsgID>1</MsgID>
    <Target>
      <LocURI>{unique device id}</LocURI>
    </Target>
    <Source>
      <LocURI>https://contoso.com/management-server</LocURI>
    </Source>
  </SyncHdr>

  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>0</CmdRef>
      <Cmd>SyncHdr</Cmd>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>2</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>2</CmdRef>
      <Cmd>Alert</Cmd>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>3</CmdRef>
      <Cmd>Replace</Cmd>
      <Data>200</Data>
    </Status>

    <!-- The server issues Get commands on two WMI providers which each -->
    <!-- return a class instance. The instance is used in future calls to -->
    <!--- the Get and Replace commands to work with the class properties. -->

    <Get>
      <CmdID>4</CmdID>
      <Item>
        <Target>
```

```
          <LocURI>./cimv2/MDM_Client</LocURI>
        </Target>
      </Item>
    </Get>
    <Get>
      <CmdID>5</CmdID>
      <Item>
        <Target>
          <LocURI>./cimv2/MDM_SideLoader/MDM_SideLoader</LocURI>
        </Target>
      </Item>
    </Get>
    <Final />
  </SyncBody>
</SyncML>

//  The client responds with a Status and Results command for each
//  of the Get commands issued by the server.

<SyncML xmlns="SYNCML:SYNCML1.2">

  <SyncHdr>
    <VerDTD>1.2</VerDTD>
    <VerProto>DM/1.2</VerProto>
    <SessionID>1</SessionID>
    <MsgID>2</MsgID>
    <Target>
      <LocURI>https://contoso.com/management-server</LocURI>
    </Target>
    <Source>
      <LocURI>{unique device id}</LocURI>
    </Source>
  </SyncHdr>

  <SyncBody>
    <Status>
      <CmdID>1</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>0</CmdRef>
      <Cmd>SyncHdr</Cmd>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>2</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>4</CmdRef>
      <Cmd>Get</Cmd>
      <Data>200</Data>
    </Status>
    <Status>
      <CmdID>3</CmdID>
      <MsgRef>1</MsgRef>
      <CmdRef>5</CmdRef>
      <Cmd>Get</Cmd>
      <Data>200</Data>
    </Status>
    <Results>
      <CmdID>4</CmdID>
      <MsgRef>1</MsgRef>
```

```
          <CmdRef>4</CmdRef>
          <Cmd>Get</Cmd>
          <Item>
            <Source>
              <LocURI>./cimv2/MDM_Client</LocURI>
            </Source>
            <Meta>
              <Format xmlns="syncml:metinf">node</Format>
            </Meta>
            <!-- Class instance - opaque to the server -->
            <Data>./cimv2/MDM_Client/MDM_Client.
                DeviceClientID=0</Data>
          </Item>
        </Results>
        <Results>
          <CmdID>5</CmdID>
          <MsgRef>1</MsgRef>
          <CmdRef>5</CmdRef>
          <Cmd>Get</Cmd>
          <Item>
            <Source>
              <LocURI>./cimv2/MDM_SideLoader</LocURI>
            </Source>
            <Meta>
              <Format xmlns="syncml:metinf">node</Format>
            </Meta>
            <!-- Class instance - opaque to the server -->
            <Data>./cimv2/MDM_SideLoader/MDM_SideLoader.
                Key=%221%22</Data>
          </Item>
        </Results>
        <Final />
      </SyncBody>
    </SyncML>
```

5   Security

5.1   Security Considerations for Implementers

None.

5.2   Index of Security Parameters

None.

Release: Friday, October 25, 2013

6 Appendix A: MOF Files

The following WMI provider **MOF** files are provided for reference.

6.1 MDMAppProv MOF File

```
// Start MDMAppProv
#region MDMAppProv
#pragma include("cim_schema_2.26.0.mof")
[Description("This class provides the ability to create a job for installing or un-installing
an application and also provides the job status and error code if any")]
class MDM_AppInstallJob
{
  [Key,Description("Unique Id for Application install/un-install")]
  string JobID;
  [Description("Package full name")]
  string PackageFullName;
  [Description("Status of the job")]
  uint32 Status;
  [Description("Error code")]
  uint32 LastError;
  [Description("Progress of the application job")]
  uint32 Progress;
  [Description("Job creation time")]
  datetime CreationTime;
  [Description("List of Urls to download content from")]
  string DownloadUrlList[];
  [Description("List of dependent application frameworks")]
  string Dependencies[];
  [Description("List of content download Urls for dependent frameworks")]
  string DependencyUrlLists[];
  [Description("Action type of install, uninstall or upgrade")]
  uint32 ActionType;
  [Description("Application type of Windows modern app, web application or remote
application")]
  uint32 JobType;
  [Description("Windows modern app deployment options")]
  uint32 DeploymentOptions;
  [static,Description("Method to create application job")]
  uint32 CreateJob([In,Description("Job details as xml string")] string JobData);
};

[Description("This class provides the ability to detect an application and inventory of
applications")]
class MDM_Application
{
  [key, Description("Package full name")]
  string PackageFullName;
  [Description("User security identifier")]
  string UserSID;
  [Description("Package name")]
  string PackageName;
  [Description("Publisher name of the package")]
  string PackagePublisher;
  [Description("Version of the package")]
  string PackageVersion;
  [Description("Modern app installation path")]
```

```
  string InstallPath;
  [Description("Is it a framework application")]
  boolean IsFramework;
  [Description("Is it a resource only package")]
  boolean IsResourcePackage;
  [Description("Is it a Appx bundle")]
  boolean IsBundle;
  [Description("Is the app in development mode")]
  boolean IsDevelopmentMode;
  [Description("Comma-separated list of depedent applications")]
  string Dependencies;
};

[Description("This class provides the ability to detect a framework application and inventory
of frameworks")]
class MDM_ApplicationFramework
{
  [key, Description("Name of the framework package")]
  string PackageName;
  [key, Description("Minimum package version for the dependent framework")]
  string MinimumPackageVersion;
  [key, Description("Publisher name of the package")]
  string PackagePublisher;
  [Description("Package full name")]
  string PackageFullName;
  [Description("Version of the package")]
  string PackageVersion;
  [Description("User security identifier")]
  string UserSID;
};

[Description("This class provides ability to detect and provide inventory of remote
applications of a subscribed device")]
class MDM_RemoteApplication
{
  [Key,Description("The url of the remote application server")]
  string FeedUrl;
  [Key,Description("The name of the remote application")]
  string AppId;
};

[Description("This class provides ability to subscribe a device to remote application
server")]
class MDM_RemoteAppUserCookie
{
    [Key,Description("The url of the remote application server")]
    string FeedUrl;
    [Description("The authentication cookie used to subscribe to the server")]
    string Cookie;
    [Description("The server generated hash of cookie used to subscribe to the server")]
    string CookieHash;
};
[Description("This class provides ability to detect and provide inventory of web applications
of a device")]
class MDM_WebApplication
{
  [Key,Description("The web application name")]
  string PackageName;
  [Description("The web application shortcut file parent folder")]
```

```
  string PackageFolder;
  [Description("The web application url")]
  string PackageUrl;
};
#endregion // End MDMAppProv
```

6.2   MDMSettingsProv MOF File

```
#region MDMSettingsProv
#pragma include("cim_schema_2.26.0.mof")
Qualifier InPartition : string[],
    Scope(class);

[Description("This class provides identification for a user/device pair. It also provides
methods for unenrolling the device from management and device lock")]
class MDM_Client
{
  [Key,Description("The key to identify the instance of MDM_Client class, This is a device
id")]
  string DeviceClientID;

  [Description("This property contains the local workstation SID")]
  string DomainSID;

  [Description("This property contains the textual name of the local Operating System")]
  string PlatformID;

  [Description("This property contains the FQDN of the device")]
  string DeviceName;

  [Description("This property contains a textual description of the main processor on the
device")]
  string ProcessorDescription;

  [Description("This property contains the SID of the local user account associated to the
device")]
  string UserSid;

  [Description("This property contains the local Operating System version in the following
format: major.minor.revision")]
  string Version;

  [static,Description("This method provides ability to unenroll the device")]
  uint32 SendUnenrollRequest([In,Description("Id of the device")] string DeviceClientID);

  [static,Description("This method provides ability to lock the device")]
  uint32 LockWorkstation();
};

[Description("This class provides capability to add additional management authorities and
retrieve previously installed authorities")]
class MDM_MgmtAuthority
{
  [Key,Description("The key to identify an authority by name")]
  string AuthorityName;
  [Description("This property contains a thumb print of the trusted root certificate for this
authority")]
```

```
  string RootThumbprint;
  [Description("This property contains a thumb print of the certificate for this authority")]
  string ProvisionedCertThumbprint;
  [Description("This property contains the certificate search criteria. E.g.,
Thumbprint=122344&store=my\\system")]
  string ClientSearchCriteria;
  [Description("This property contains a list of the management server URLs associated to
this authority")]
  string ServerList;
  [static,Description("A method to create a new authority")]
  uint32 CreateNewAuthority([In,Description("Authority name")] string AuthorityName);
};

[Description("This class provides capability to get and set configuration settings for the
oma-dm agent")]
class MDM_ConfigSetting
{
  [Key,Description("The key to identify a setting by name")]
  string SettingName;

  [Description("This property contains a setting value associated to a SettingName")]
  string SettingValue;
};

[Description("This class provides information on the WNS channel used for notifications to
the MDM agent")]
class MDM_WNSChannel
{
  [key,
   Description("The identity of the MDM application")]
 string AppId;
  [Description("The SID of the enrolled user")]
  string UserSID;
  [Description("This is the channel created for WNS notifications for the MDM user on this
device")]
  string Channel;
  [Description("The expiration time for the WNS channel")]
  datetime ExpirationTime;
  [Description("WNS Channel Creation Status"),ValueMap{"0", "1"}]
  uint32 ChannelStatus;
  [Description("This is the error code in the event that WNS channel creation fails")]
  uint32 LastError;
};

[Description("This class provides the capability to configure WNS notifications for the MDM
agent")]
class MDM_WNSConfiguration
{
  [key,
  Description("The identity of the MDM application")]
 string AppId;
  [Description("ConfigurationStatus is 0 if not configured and 1 if configured")]
  uint32 ConfigurationStatus;
  [static, Description("Updates WNS Configuration")]
  uint32 UpdateConfiguration([In,Description("ConfigString is in the format <AppId>;<WNS
Package Family Name>")] string ConfigString);
};

// Valid Names/Values
```

```
    // MinPasswordLength = "1".."16"
    // DisallowConvenienceLogon = true / false
    // AutolockTimeout = "1s".."1200s"
    // MaxHistory = "1"..
    // Expiration = "1d"..
    // MaxAttemptsBeforeWipe = "4".."16"
    // MinPasswordComplexCharacters = "1" / "2" / "3" / "4"
    // Ex. NamedValuesList=MinPasswordLength=4;AutolockTimeout=300
[Description("This class provides EAS PIN Password Policy configuration on the device")]
class MDM_EASPolicy
{
    [Key,Description("The key to identify the instance of MDM_EASPolicy class")]
    Uint32  Key;
    [static, Description("Set password related settings values")]
    uint32 SetValues([In,Description("Settings name value pairs in formt :
<name1>,<value1>;<name2>,<value2>;...")] string NamedValuesList);
};


[Description("This class provides configuration to restrict settings on the device")]
class MDM_Restrictions
{
    [Key,Description("The key to identify the instance of MDM_Restrictions class")]
    Uint32  Key;
    [Description("This property allows 'Windows Error Reporting' to be enabled or disabled on
the device")]
    boolean  DiagnosticsSubmissionEnabled;
    [Description("This property allows 'Data Roaming' to be enabled or disabled on the
device")]
    boolean  DataRoamingEnabled;
    [Read,Description("This property returns whether at least one Bluetooth device is found
and is enabled on the device")]
    boolean  BluetoothEnabled;
    [Read,Description("This property returns whether at least one wifi network adapter is
found and is enabled on the device")]
    boolean  WifiEnabled;
    [Read,Description("This property returns whether the 'Sync your settings' option under
'Change PC Settings' is turned on")]
    boolean  PCSettingsSyncEnabled;
    [Read,Description("This property returns whether the 'Sync settings over metered
connection' option under 'Change PC Settings' is turned on")]
    boolean  PCSettingsMeteredNetworkSyncEnabled;
    [Read,Description("This property returns whether the 'Passwords' option under 'Change PC
Settings' is turned on")]
    boolean  PCSettingsPasswordSyncEnabled;
    [Description("This property returns the Enterprise Client Sync product's url for which
the client will sync to")]
    string EcsSyncUrl;
    [Description("This property returns whether the Enterprise Client Sync product will be
auto provisioned")]
    boolean EcsAutoProvisionEnabled;
    [Description("This propery returns the current 'User Access Control' level on the
client"),
     Values {"Always Notify", "Notify App Changes", "Notify App Changes (No Dim)", "Never
Notify"}]
    Uint32  UserAccountControlStatus;
    [Description("True if Smart Screen is enabled")]
    boolean SmartScreenEnabled;
};


[Description("This class provides security health metrics on the device")]
```

```
class MDM_SecurityStatus
{
    [Key,Description("The key to identify the instance of MDM_SecurityStatus class")]
    Uint32  Key;
    [Read,Description("This property provides the status of the firewall"),
     Values {"Good", "NotMonitored", "Poor", "Snooze"}]
    Uint32  FirewallStatus;
    [Read,Description("This property provides the status of Windows updates"),
     Values {"Disable Automatic Updates", "Notify for download and notify for install", "Auto
download and notify for install", "Auto download and schedule the install"}]
    Uint32  AutoUpdateStatus;
    [Read,Description("This property provides the status of antivirus"),
     Values {"Good", "NotMonitored", "Poor", "Snooze"}]
    Uint32  AntiVirusStatus;
    [Read,Description("This property provides the status of antivirus updates"),
     Values {"On", "Off", "Snoozed", "Expired"}]
    Uint32  AntiVirusSignatureStatus;
    [Read,Description("This property provides the status of bitlocker encryption")]
    boolean RequireEncryption;
};

[Description("This class provides methods for activating sideloading of LOB applications")]
class MDM_SideLoader
{
  [key, Description("The key to identify the instance of MDM_SideLoader class.")]
  uint32 key;
  [Description("This property provides a hash of the installed product key.")]
  string ProductKeyHash;
  [static,Description("A method for activating sideloading and installing the product key")]
  uint32 ActivateKey([In,Description("Sideloading key")] string ProductKey);
  [static,Description("A method for adding the package signing certificate")]
  uint32 AddCertificate([In,Description("Certificate In Base-64 encoded format")] string
CertificateBlob);
  [static,Description("A method to disable sideloading LOB applications")]
  uint32 UnActivateLOB();
};

[InPartition {"local-user"},Description("This class provides ability to manage wireless
profiles on the device")]
class MDM_WirelessProfile
{
    [Key, Description("Name of a wireless LAN profile.")]
    string Name;

    [Description("SSID for the wireless LAN.")]
    string SSID;

    [Description("Connect When Not Broadcasting")]
    boolean ConnectWhenNotBroadcasting;

    [Description("Auto Connect")]
    boolean AutoConnect;

    [ValueMap {"1", "2"}, Values {"ESS", "IBSS"}, Description("Connection Type")]
    uint8 ConnectionType;

    [Description("This indicates whether to connect to a hidden network or not.")]
    boolean ConnectToMorePreferedNetwork;
```

```
    [ValueMap{"1", "2", "3", "4", "5", "6", "7"},
    Values{"Open", "WPA-Peronal", "WPA-2-Personal", "WPA-Enterprise", "WPA2-Enterprise",
"shared", "WEP"},
    Description("Authentication method to be used to connect to the wireless LAN.")]
    uint8 SecurityAuthentication;

    [ValueMap{"1", "2",    "3",    "4"},
    Values {"None", "WEP", "TKIP", "AES"},
    Description("Type of data encryption to use to connect to a wireless LAN.")]
    uint8 SecurityEncryption;

    [Description("This indicates whether PMK caching will be used.")]
    boolean PMKCacheMode;

    [Description("This specifies the number of entries in the PMK cache.")]
    uint32 PMKCacheSize;

    [Description("This indicates the length of time, in minutes, that a PMK cache will be
 kept.")]
    uint32 PMKCacheTTL;

    [Description("This indicates whether pre-authentication will be used by the client.")]
    boolean PreAuthMode;

    [Description("This pecifies the number pre-authentication attempts to try on neighboring
Access Points.")]
    uint32 PreAuthThrottle;

    [Description("This indicates whether Federal Information Processing Standards (FIPS) mode
is enabled.")]
    boolean EnableFIPSCompliance;

    [ValueMap {"1", "2"},
    Values {"passPhrase", "networkKey"},
    Description("This indicates whether the shared key will be a network key or a pass
phrase.")]
    uint8 SharedKeyType;

    [Description("This contains a network key or passphrase.")]
    string SharedKeyMaterial;

    [Description("This indicates whether a shared key is encrypted.")]
    boolean SharedKeyProtected;

    [Description("This specifies whether the user credentials are cached for subsequent
connections.")]
    boolean OneXCacheUserData;

    [ValueMap{"1", "2", "3", "4"},
    Values{"MachineOrUser", "Machine", "User", "Guest"},
    Description("This specifies the type of credentials used for authentication.")]
    uint8 OneXAuthenticationMode;

    [ValueMap {"1", "2"},
    Values {"preLogon", "postLogon"},
    Description("This specifies when single sign on is performed.")]
    uint8 OneXSingleSignOnType;
```

```
    [Description("This specifies, in seconds, the maximum delay before the single sign on
connection attempt fails.")]
    uint32 OneXSingleSignOnMaxDelay;

    [Description("This specifies that different dialog boxes are presented to the user at
logon for Single Sign On, if applicable.")]
    boolean OneXSingleSignOnAllowAdditionalDialogs;

    [Description("This specifies if the virtual LAN (VLAN) used by the device changes based
on the user's credentials.")]
    boolean OneXSingleSignOnUserBasedVirtualLAN;

    [ValueMap{"13", "18", "21", "23", "25", "50"},
    Values {"TLS", "SIM", "TTLS", "AKA", "PEAP", "AKAPRIME"},
    Description("Type of the EAP used for this connection.")]
    uint8 OneXEAPType;

    [Description("EAP Configuration xml used for this connection.")]
    string OneXEAPXml;
};

[InPartition {"local-user"},Description("This class provides ability to manage wireless
profiles through WLAN xml")]
class MDM_WirelessProfileXml
{
    [key, Description("Name of a wireless LAN profile.")]
    string          Name;
    [Description("Wireless Profile xml for this connection.")]
    string          ProfileXml;
};

[Description("This class provides the ability to add an application as a trigger to MDM
installed VPN profiles")]
class MDM_VpnApplicationTrigger
{
    [key, Description("Package family name of the Application")]
    string ApplicationID;

    [Description("Is application trigger added to all MDM installed VPN profiles")]
    boolean         TriggerEnabledInAllMDMProfiles;
};

[Description("This class provides device registration details")]
class MDM_DeviceRegistrationInfo
{
    [Key,Description("The identity of the device in the AD device registration record")]
    string          DeviceId;
    [Description("The UPN of the registered user")]
    string          UPN;
    [Description("The thumbprint of the AD certificate for device registration")]
    string          CertificateThumbprint;
};

[Description("This class provides configuration to modify browser security settings on the
device")]
class MDM_BrowserSettings
{
    [Key,Description("The key to identify the instance of MDM_BrowserSettings class")]
    Uint32      Key;
```

```
    [Description("True if Force Fraud Warning is enabled")]
    boolean            ForceFraudWarning;
    [Description("True if Autofill is enabled")]
    boolean            AutofillEnabled;
    [Description("True if scripting is enabled for the Internet security zone")]
    boolean            InternetScriptingEnabled;
    [Description("True if plugins are enabled")]
    boolean            InternetPluginsEnabled;
    [Description("True if popups are disabled")]
    boolean            InternetBlockPopups;
    [Description("True if the always send do not track header setting is enabled")]
    boolean            AlwaysSendDoNotTrackHeader;
    [Description("True if the Intranet security zone is enabled")]
    boolean            IntranetSecurityZoneEnabled;
    [Description("True if the go to Intranet for single word setting is enabled")]
    boolean            GoToIntranetForSingleWord;
    [Read,
    ValueMap {"0", "1", "2"},
    Values {"High", "Medium-high", "Medium"},
    Description("Security level for the Internet security zone")]
    Uint32            InternetZoneSecurityLevel;
    [Read,
    ValueMap {"0", "1", "2", "3", "4"},
    Values {"High", "Medium-high", "Medium", "Medium-low", "Low"},
    Description("Security level for the Intranet security zone")]
    Uint32            IntranetZoneSecurityLevel;
    [Read,
    ValueMap {"0", "1", "2", "3", "4"},
    Values {"High", "Medium-high", "Medium", "Medium-low", "Low"},
    Description("Security level for the Restricted Sites security zone")]
    Uint32            RestrictedSitesZoneSecurityLevel;
    [Read,
    ValueMap {"0", "1", "2", "3", "4"},
    Values {"High", "Medium-high", "Medium", "Medium-low", "Low"},
    Description("Security level for the Trusted Sites security zone")]
    Uint32            TrustedSitesZoneSecurityLevel;
};

[Description("This class provides configuration to modify browser security zone settings on
the device")]
class MDM_BrowserSecurityZones
{
    [Key,
    Description("A namespace that may belong in a security zone")]
    string        Namespace;
    [Key,
    ValueMap {"1", "2", "3", "4"},
    Values {"Intranet", "Trusted", "Internet", "Untrusted"},
    Description("Key identifier for the security zone")]
    Uint32            Zone;
    [Description("True if the namespace exists in the specified security zone")]
    boolean            Exists;
};

[Description("Certificate Enrollment provider using Simple Certificate Enrollment Protocol")]
class MDM_CertificateEnrollment
{
    [key, Description("Enrollment Certificate Request ID")]
    string      RequestID;
```

```
    [key, Values {"1", "2"}, ValueMap {"ContextUser", "ContextMachine"},
Description("Certificate store location")]
    uint8       StoreLocation;

    [key, Description("Certificate selection criteria: Enhanced key usages with Comma as
delimiter. Matching on all specified EKUs")]
    string      EnhancedKeyUsages;

    [key, Description("Certificate selection criteria: Issuer subject names with separator
bar as delimiter. Case sensitive matching on one of specified issuers")]
    string      Issuers;

    [Description("Used for compliance to determine if certificate matching the enrollment
request and criteria is issued")]
    uint32      Status;

    [Description("Used to determine detailed enrollment request error if Status is
EnrollError")]
    uint32      Error;

    [Description("Certificate expiration threshold in days ")]
    uint32      ExpirationThreshold;

    [Description("Subject name of the issued certificate")]
    string      SubjectName;

    [Description("Subject alternative names of the issued certificate with separator bar as
delimiter.")]
    string      SubjectAlternativeNames;

    [Description("Thumbprint of the issued certificate")]
    string      Thumbprint;

    [Description("Serial number of the issued certificate")]
    string      SerialNumber;

    [Description("Valid from date of the issued certificate")]
    datetime    ValidFrom;

    [Description("Valid to date of the issued certificate")]
    datetime    ValidTo;

    [Description("XML blob containing static attributes like Issuer Name, SCEP URL, Retries
which will be used by the client for the SCEP calls to Network Device Enrollment Service
(NDES). This one comes in the CI digest")]
    string      ConfigurationParameters;


    [Description("List of target servers to associate to the certificate in credential
manager for SSO")]
    string      CredManTargetList;
};

[Description("This class provides the ability to install and delete the certificates on the
device")]
class MDM_Certificate
{
    [key, Values {"1", "2"}, ValueMap {"ContextUser", "ContextMachine"},
Description("Certificate store location")]
```

```
    uint8        StoreLocation;

    [key, Description("Internal store name to install the certificate")]
    string        StoreName;

    [key, Description("Thumbprint of the certificate")]
    string        Thumbprint;

    [read, Description("Is certificate installed")]
    boolean      IsInstalled;

    [Description("Base-64 encoded raw certificate blob")]
    string        Blob;
};

[Description("This class provides the ability to set Windows modern application settings")]
class MDM_ApplicationSetting
{
    [key, Description("Package Family Name")]
    string PackageFamilyName;

    [key, Description("Setting Name")]
    string SettingName;

    [ValueMap {"0", "1", "2", "3"},
    Values {"string", "int", "bool", "double"},
    Description("Setting Type")]
    Uint32 SettingType;

    [Description("Setting Value")]
    string SettingValue;
};
#endregion // MDMSettingsProv
```

7   Appendix B: Product Behavior

This document specifies version-specific details in the Microsoft .NET Framework. For information about which versions of .NET Framework are available in each released Microsoft Windows product or as supplemental software, see .NET Framework.

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Windows 8.1 operating system

- Windows Server 2012 R2 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 3.1.5.1.6: The Win32_DisplayConfiguration WMI class is not supported in Windows 8.1 and Windows Server 2012 R2.

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index