

[MS-WSMAN]: Web Services Management Protocol Extensions for Windows Server 2003

Intellectual Property Rights Notice for Protocol Documentation

- This protocol 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 protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- Microsoft does not claim any trade secret rights in this documentation.
- Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. If you are interested in obtaining a patent license, please contact protocol@microsoft.com.
- 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.
- All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

This protocol documentation is intended for use in conjunction with publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

A protocol specification does 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.

Revision Summary

Date	Revision History	Revision Class	Comments
12/18/2006	0.1		MCPD Milestone 2 Initial Availability
03/02/2007	1.0		MCPD Milestone 2
04/03/2007	1.1		Monthly release
05/11/2007	1.2		Monthly release

Date	Revision History	Revision Class	Comments
06/01/2007	1.2.1	Editorial	Revised and edited the technical content.
07/03/2007	1.2.2	Editorial	Revised and edited the technical content.
07/20/2007	1.2.3	Editorial	Revised and edited the technical content.
08/10/2007	2.0	Major	Updated and revised the technical content.
09/28/2007	2.0.1	Editorial	Revised and edited the technical content.
10/23/2007	3.0	Major	Added new content.
11/30/2007	3.0.1	Editorial	Revised and edited the technical content.
01/25/2008	3.0.2	Editorial	Revised and edited the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References.....	8
1.3	Protocol Overview (Synopsis).....	8
1.4	Relationship to Other Protocols.....	9
1.5	Prerequisites/Preconditions	10
1.6	Applicability Statement	10
1.7	Versioning and Capability Negotiation.....	10
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments.....	11
2	Messages	12
2.1	Message Syntax	12
2.1.1	Common Data Types	12
2.1.1.1	Fault Detail	12
2.1.1.2	CIM Data Types.....	13
2.1.2.1	DateTime	13
2.1.3	Configuration Data Types.....	13
2.1.3.1	Common Configuration Data Types	14
2.1.3.1.1	ConfigType	14
2.1.3.2	Client Configuration Data Types.....	15
2.1.3.2.1	ClientType	15
2.1.3.2.2	ClientUnencryptedType	16
2.1.3.2.3	ClientHTTPType	16
2.1.3.2.4	ClientHTTPSType	17
2.1.3.3	Server Configuration Data Types	17
2.1.3.3.1	ListenerType	17
2.1.3.3.2	ServiceUnencryptedType	19
2.1.3.3.3	ServiceHTTPType	19
2.1.3.3.4	ServiceHTTPSType	19
2.1.3.3.5	ServiceType	20
2.1.3.4	Miscellaneous Data Types.....	21
2.1.3.4.1	ThisType	21
3	Protocol Details	22
3.1	WS-Management Details	22
3.1.1	Abstract Data Model	22
3.1.2	Timers	22
3.1.3	Initialization	22
3.1.4	Higher-Layer Triggered Events.....	22
3.1.5	Message Processing	22
3.1.5.1	Common Headers.....	22
3.1.5.1.1	XML Namespace Definitions.....	22
3.1.5.1.2	wsman:SelectorSet	23
3.1.5.1.3	wsa:ReplyTo	23
3.1.5.1.4	wsa:FaultTo	23
3.1.5.1.5	wsa:MessageID	23
3.1.5.1.6	wsa:Action.....	24
3.1.5.1.7	wsa:To.....	24
3.1.5.1.8	wsa:Relates To	24

3.1.5.1.9	wsman:OperationTimeout	25
3.1.5.1.10	wsman:MaxEnvelopeSize	25
3.1.5.1.11	wsman:Locale	25
3.1.5.1.12	wsman:OptionSet	25
3.1.5.1.13	wsman:RequestEPR	26
3.1.5.2	Enumeration	26
3.1.5.2.1	wsen:Filter	26
3.1.5.2.2	wsen:Pull/wsen:MaxElements	26
3.1.5.2.3	wsman:RequestTotalItemsCountEstimate	26
3.1.5.2.4	wsman:OptimizeEnumeration	27
3.1.5.2.5	wsman:EnumerationMode	27
3.1.5.2.6	wsman:Filter	27
3.1.5.3	Transfer Get and Put	27
3.1.5.3.1	wsman:FragmentTransfer	27
3.1.5.4	Miscellaneous	27
3.1.5.4.1	Concurrent Operations	27
3.1.5.4.2	Inbound Message Size	28
3.1.5.4.3	Fault Detail	28
3.1.5.4.4	Metadata and Discovery	28
3.1.5.4.5	Binary Attachments	28
3.1.5.5	Configuration	29
3.1.5.5.1	wsman:microsoft.com/wsman/2005/06/config	29
3.1.5.5.2	wsman:microsoft.com/wsman/2005/06/config/client	29
3.1.5.5.2.1	wsman:microsoft.com/wsman/2005/06/config/client/http	29
3.1.5.5.2.2	wsman:microsoft.com/wsman/2005/06/config/client/http/unencrypted ...	30
3.1.5.5.2.3	wsman:microsoft.com/wsman/2005/06/config/client/https	30
3.1.5.5.3	wsman:microsoft.com/wsman/2005/06/config/service	31
3.1.5.5.3.1	wsman:microsoft.com/wsman/2005/06/config/service/http	31
3.1.5.5.3.2	wsman:microsoft.com/wsman/2005/06/config/service/http/unencrypt ed	32
3.1.5.5.3.3	wsman:microsoft.com/wsman/2005/06/config/service/https	32
3.1.5.5.4	wsman:microsoft.com/wsman/2005/06/config/listener	33
3.1.5.5.4.1	Enumeration of Listeners	33
3.1.5.5.4.2	Retrieval and Modification of Individual Listeners	33
3.1.6	Timer Events	34
3.1.6.1	Enumeration Garbage Collection Timer	34
3.1.7	Other Local Events	34
3.2	WS-CIM Mapping Details	35
3.2.1	Abstract Data Model	35
3.2.2	Timers	35
3.2.3	Initialization	35
3.2.4	Higher-Layer Triggered Events	35
3.2.5	Message Processing	35
3.2.5.1	Embedded Objects	35
3.2.5.2	CIM XML Namespace	35
3.2.5.3	Arrays	36
3.2.5.4	cim:Location	36
3.2.6	Timer Events	36
3.2.7	Other Local Events	36
3.3	WS-Management - CIM Binding Details	36
3.3.1	Abstract Data Model	36
3.3.2	Timers	37
3.3.3	Initialization	37
3.3.4	Higher-Layer Triggered Events	37

3.3.5	Message Processing	37
3.3.5.1	wsmb:PolymorphismMode	37
3.3.5.2	CIM Namespace	37
3.3.6	Timer Events.....	38
3.3.7	Other Local Events	38
4	Protocol Examples	39
4.1	CIM examples.....	39
4.1.1	Retrieving a CIM Instance	39
4.1.2	Enumeration of Instances	41
4.1.2.1	Enumerate Request	42
4.1.2.2	Enumerate Response	43
4.1.2.3	First Pull Request	44
4.1.2.4	First Pull Response	44
4.1.2.5	Second Pull Request	45
4.1.2.6	Second Pull Response with EndOfSequence	46
4.1.3	Modifying an Instance	47
4.1.4	Invoking a Method	49
4.2	Configuration Examples.....	51
4.2.1	Retrieving Configuration	51
4.2.2	Modify Configuration	53
4.3	Fault Detail	56
5	Security	58
5.1	Security Considerations for Implementers	58
5.2	Index of Security Parameters	58
6	Appendix A: Windows Behavior	59
7	Index.....	62

1 Introduction

Web Services for Management (WS-Management), as specified in [\[DMTF-DSP0226\]](#), is a general-purpose, SOAP-based systems management protocol that defines procedures for carrying out remote management operations.

This document specifies Microsoft extensions to the WS-Management Protocol. The extensions include:

- **WS-Management** configuration extensions. These allow configuration through the WS-Management Protocol.
- WS-Management fault detail extensions. These specify fault detail extensions that include detailed information about the fault.
- Extensions to WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#) and WS-Management CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#). These allow **CIM objects** to be accessed using the WS-Management Protocol.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

CIM Class
CIM Instance
CIM Method
CIM Namespace
CIM Object

The following terms are specific to this document:

Action URI: Identifies which operation needs to be carried out against a **resource**.

Client: The client application using the WS-Management Protocol to access the management **service**, on the local or a remote computer.

Common Information Model (CIM): The **DMTF** model that describes how to represent real-world computer and network objects. CIM uses an object-oriented paradigm, where managed objects are modeled using the concepts of classes and instances. See [\[DMTF-DSP004\]](#).

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

Endpoint Reference (EPR): A combination of WS-Addressing ([\[WSAddressing\]](#)) and **WS-Management** addressing elements that together describe an address for a **resource** in the message **SOAP** header.

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.

Property: A name/value pair that describes a unit of data for a class. Property values must have a valid **Managed Object Format (MOF)** data type.

Resource: An endpoint that represents a distinct type of management operation or value. A **service** exposes one or more resources and some resources can have more than one

instance. For example, the Win32_LogicalDisk class represents a resource. Win32_LogicalDisk="C:\\" is a specific instance of the resource.

Resource URI: The **Uniform Resource Identifier** used to identify a specific type of **resource**, such as disks or processes, on a system.

Selector: A name/value pair that represents a particular instance of a **resource**. This is essentially a filter or "key" that identifies the desired instance of the **resource**.

Service: An application that provides management services to **clients** through the WS-Management Protocol and other Web services.

SOAP: Simple Object Access Protocol. An XML-based protocol used by Web **services**. See [\[SOAP1.2/1\]](#).

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 [\[RFC3986\]](#).

Windows Management Instrumentation (WMI): The Microsoft implementation of the Web-Based Enterprise Management (WBEM) standard published by **Distributed Management Task Force (DMTF)**. WMI allows an administrator to manage local and remote computers and models computer and network objects using an extension of the **Common Information Model (CIM)** standard.

WS-Management: A public standard SOAP-based protocol for sharing management data among all operating systems, computers, and devices.

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

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.

[DMTF-DSP004] Distributed Management Task Force, "Common Information Model (CIM) Infrastructure Specification", Version 2.3, October 2005, http://www.dmtf.org/standards/published_documents/DSP0004V2.3_final.pdf

[DMTF-DSP0226] Distributed Management Task Force, Inc., "Web Services for Management (WS-Management)", Version 1.0.0a, April 2006, http://www.dmtf.org/standards/published_documents/DSP0226.pdf

[DMTF-DSP0227-NM] Distributed Management Task Force, Inc., "CIM Binding Specification (WS-Management)", Version 1.0.0b, August 2006, http://www.dmtf.org/standards/published_documents/DSP0227.pdf

[DMTF-DSP0230-NM] Distributed Management Task Force, Inc., "WS-CIM Mapping Specification", Version 1.0.0c, July 2006, http://www.dmtf.org/standards/published_documents/DSP0230.pdf

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](#)", January 2007.

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

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

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and Stewart, L., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999, <http://www.ietf.org/rfc/rfc2617.txt>

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, <http://www.ietf.org/rfc/rfc4559.txt>

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>

[WSAddressing] Box, D. et. al, "Web Services Addressing (WS-Addressing)", August 2004, <http://www.w3.org/Submission/ws-addressing/>

If you have any trouble finding [WSAddressing], please check [here](#).

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

[MS-WMI] Microsoft Corporation, "[Windows Management Instrumentation Remote Protocol Specification](#)", September 2007.

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

1.3 Protocol Overview (Synopsis)

The Web Services Management Protocol Extensions for Windows Server 2003 are a set of modifications to the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#), the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#), and the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#) for compatibility with Windows Server 2003 R2. It should be noted that conformance to these extensions does not make an implementation compatible with the WS-Management Protocol specifications as currently published by the **DMTF**. Instead, conformance makes an implementation compatible with Microsoft's implementation of WS-Management in Windows Server 2003 R2, which was based on the pre-release drafts of the WS-Management Protocol specifications available at the time Windows Server 2003 R2 was implemented and released.[<1>](#)

WS-Management is a remote management protocol that can be used for managing software and hardware components and is specified in [\[DMTF-DSP0226\]](#).

The WS-Management Protocol can provide remote access to **Common Information Model (CIM)** objects as specified by [\[DMTF-DSP004\]](#). WS-Management servers expose a set of entities that can

be managed as objects with attributes and methods. WS-Management **clients** perform management tasks by issuing object operations against objects exposed by WS-Management servers.

The CIM management schema provides a standard framework and set of base classes that describe a managed environment. Each type of managed entity is described by a **CIM class** and individual entities are managed through instances of the appropriate CIM class. For example, a logical disk drive may be managed through an instance of the CIM_DiskDrive class. This class contains various member **properties** such as Name, DeviceID, and Status. For a system with multiple storage devices, the CIM_DiskDrive class will contain one instance for each storage device on the system. The CIM_DiskDrive class may also be subclassified to add platform-specific properties such as the drive letter used on the Windows platform. CIM class definitions are similar to class definitions in other object-oriented database systems and object-oriented programming languages.

CIM-based management in a Web services environment requires that the CIM schema (classes, properties, and methods) be rendered in both XML Schema and WSDL (the Web Services Description Language). To achieve this, the Common Information Model (CIM) must be mapped to WSDL and XML Schema using a translation or mapping algorithm. The WS-CIM Mapping Specification, as specified in [\[DMTF-DSP0230-NM\]](#), specifies the normative rules and recommendations that describe the structure of the XML Schema, WSDL fragments, and metadata fragments corresponding to the elements of CIM models. The WS-CIM Mapping Specification also specifies the representation of **CIM instances** as XML instance documents.

The WS-Management CIM Binding specification, as specified in [\[DMTF-DSP0227-NM\]](#), describes how to name and access CIM entities using the WS-Management Protocol. To query the status of managed entities, the WS-Management Protocol is used to retrieve their CIM instances using operations such as Get and Enumerate. Updates to managed entities are sent using the WS-Management Put operation. To invoke a **CIM method**, a WS-Management Protocol custom method is used.

The Web Services Management Protocol Extensions for Windows Server 2003 include six sets of changes to the WS-Management Protocol, the WS-Management CIM Binding Specification, and the WS-CIM Mapping Specification:

- Differences in the **Resource URI** prefix used to identify managed entities in the WS-Management Protocol.
- Differences in the XML namespaces for WS-Management and CIM mapping namespaces.
- Differences in element tags.
- New element tags for vendor-extensible tags.
- Unsupported actions and actions with only limited support.
- New data types for configuration of Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

[<2><3>](#)

1.4 Relationship to Other Protocols

The WS-Management Protocol uses **SOAP**, as specified in [\[SOAP1.2/1\]](#), over HTTP or HTTPS for communication. The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), MUST be used as the transport to provide access to CIM data using binding techniques specified by [\[DMTF-DSP0227-NM\]](#) and mapping techniques specified by [\[DMTF-DSP0230-NM\]](#). The Web Services Management Protocol Extensions for Windows Server 2003 Protocol specifies the differences between the

protocols as defined in early drafts and supported in Windows Server 2003 R2 and the protocols as specified in [\[DMTF-DSP0226\]](#), [\[DMTF-DSP0227-NM\]](#), and [\[DMTF-DSP0230-NM\]](#).

The Windows Management Instrumentation Protocol, as specified in [\[MS-WMI\]](#), is an alternate network protocol for accessing CIM data on servers.

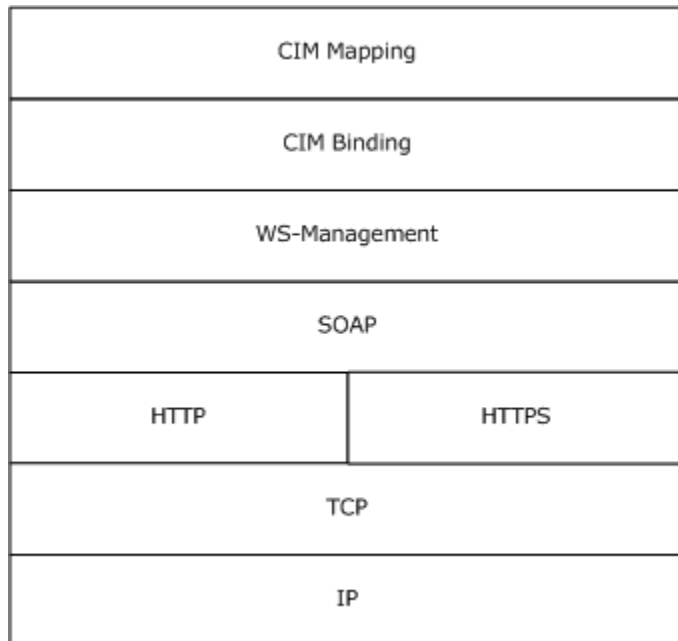


Figure 1: Web Services Management (WS-Management) Protocol Stack

1.5 Prerequisites/Preconditions

For a client that uses the WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), to communicate with a server, the server MUST have an operational SOAP1.2/HTTP1.1/TCP/IP stack. WS-Management assumes that the system already has an IP address and is thus able to communicate on the network. It also assumes that the client has already obtained the IP address and HTTP port of the server, for example, through manual configuration.

1.6 Applicability Statement

Web Services Management Protocol Extensions for Windows Server 2003 is a protocol for accessing CIM objects to remotely administer software and hardware configuration.

1.7 Versioning and Capability Negotiation

The WS-Management Protocol defines a simple request called Identity to facilitate the process of finding the protocol version or versions supported by the [services.<4>](#)

1.8 Vendor-Extensible Fields

Web Services Management Protocol Extensions for Windows Server 2003 does not define any vendor-extensible fields.

1.9 Standards Assignments

The following table shows the standard XML namespaces used within this protocol and the alias (prefix) used in the remaining sections of this protocol specification.

Prefix	XML Namespace	Specification
s	http://www.w3.org/2003/05/soap-envelope	[SOAP1.2/1]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] and [XMLSCHEMA2]
xsi	http://www.w3.org/2001/XMLSchema-instance	[XMLSCHEMA1]
a	http://schemas.xmlsoap.org/ws/2004/08/addressing	[WSAddressing] Section 1.2
w	http://schemas.xmlsoap.org/ws/2005/06/management	[DMTF-DSP0226] Section 1.5
wsmanfault	http://schemas.microsoft.com/ws/2005/02/wsmanfault	Section 2.1.1.1
cim	http://schemas.dmtf.org/wsman/2005/06/base	[DMTF-DSP0230-NM] Section 5.3

Because Web Services Management Protocol Extensions for Windows Server 2003 is based on pre-release drafts of [\[DMTF-DSP0226\]](#) and [\[DMTF-DSP0230-NM\]](#), it uses <http://schemas.xmlsoap.org/ws/2005/06/management> as the XML namespace for WS-Management instead of <http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd> and uses <http://schemas.dmtf.org/wsman/2005/06/base> as the CIM XML namespace instead of <http://schemas.dmtf.org/wbem/wscim/1/common>.

In addition, Web Services Management Protocol Extensions for Windows Server 2003 uses [cfg:wsman:microsoft.com/wsman/2005/06/config.xsd](http://schemas.microsoft.com/ws/2005/06/config) as the XML namespace for configuration of Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

Note XML namespaces are to be treated as identifiers and they do not point to any location on the Web.

2 Messages

The following sections specify the message syntax of the mailslot write message.

2.1 Message Syntax

The syntax for the messages in the WS-Management Protocol is specified in [\[DMTF-DSP0226\]](#). Web Services Management Protocol Extensions for Windows Server 2003 changes the Resource URI prefix for DMTF-compliant CIM classes, as specified in [\[DMTF-DSP0227-NM\]](#) section 5.1, from <http://schemas.dmtf.org/wbem/wscim/1/cim-schema> to <http://schemas.dmtf.org/wsman/2005/06>. Web Services Management Protocol Extensions for Windows Server 2003 defines new fault detail data types and renames the cimDateTime type.

Note XML namespaces are to be treated as identifiers and they do not point to any location on the Web.

2.1.1 Common Data Types

This section describes the common data types that **MUST** be used by Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

2.1.1.1 Fault Detail

A SOAP fault is used to carry error information within a SOAP message. Faults are returned when the SOAP message is successfully delivered by the transport and processed by the services, but the message cannot be processed properly.

The WS-Management Protocol defines an extensibility model that allows a service to include additional fault information in the s:Detail element. Web Services Management Protocol Extensions for Windows Server 2003 defines WSMANFault for reporting additional fault information. The schema for WSMANFault **MUST** be as shown below:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsmanfault="http://schemas.microsoft.com/ws/
2005/06/wsmanfault"
  targetNamespace="http://schemas.microsoft.com/ws/2005/
06/wsmanfault"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:complexType name="WSManFaultType">
  <xs:sequence>
  <xs:element name="Message"
  type="wsmanfault:MessageType"
  minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="Code" type="xs:unsignedInt"
  use="required"/>
  <xs:attribute name="Machine" type="xs:string"
  use="required"/>
  <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:element name="WSManFault"
  type="wsmanfault:WSManFaultType"/>
  <xs:complexType name="ProviderFaultType"
  mixed="true">
  <xs:sequence>
```

```

<xs:any processContents="lax" minOccurs="0"
  maxOccurs="unbounded"/>
</xs:sequence>

<xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="MessageType" mixed="true">
<xs:sequence>
<xs:element name="ProviderFault"
  type="wsmanfault:ProviderFaultType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:schema>

```

Code: An integer that MUST specify the Windows error code value.

Machine: A string that MUST specify the machine name where this fault occurred.

Message: Message is a complex type that allows mixed content. It can contain a sequence of elements of type ProviderFault and text that contains user-friendly description of the fault. Each ProviderFault element contains resource-specific fault information. ProviderFault can contain a sequence of elements of xs:any type.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST include either a Message element of xs:string type that contains a user-friendly description of the error or a ProviderFault element that MUST contain WSMFault element of type WSMFault. This WSMFault element MUST contain a Message element that is of xs:string type that contains a user-friendly description of the error.

WSManFault: WSMFault is a complex type that contains a sequence of messages, code, and the machine information.

ProviderFault: Element that MUST contain specific **resource** provider fault information.

2.1.2 CIM Data Types

This section describes CIM-related data types used by Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

2.1.2.1 DateTime

The CIM specification defines the DateTime type for specifying a time stamp (point in time) or an interval. If it specifies a time stamp, it allows preservation of the time zone offset. In both cases, DateTime allows specification of varying precision of the date and time information. WS-CIM mapping specification [\[DMTF-DSP0230-NM\]](#) section 6.1 specifies cim:cimDateTime type for mapping DateTime to XML. Web Services Management Protocol Extensions for Windows Server 2003 renames the CIM_DateTime element of cim:cimDateTime specified in [\[DMTF-DSP0230-NM\]](#) section 6.1 to CIM_DateTimeString. <5>

2.1.3 Configuration Data Types

This section defines the data types used for configuring Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

2.1.3.1 Common Configuration Data Types

This section defines the common data types used for configuring Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

2.1.3.1.1 ConfigType

ConfigType is the overall container for the Web Services Management Protocol Extensions for Windows Server 2003 clients' and servers' configuration. Note that listeners are not part of this configuration and need to be retrieved separately. The schema MUST be as shown below:

```
<xs:schema xmlns:cfg="wsman:microsoft.com/wsman/2005/06
/config.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="wsman:microsoft.com/wsman/2005/06/
config.xsd"
  elementFormDefault="qualified">
  <xs:element name="Config" type="cfg:ConfigType"/>
  <xs:complexType name="ConfigType">
    <xs:sequence>
      <xs:element name="MaxEnvelopeSizekb"
        type="xs:unsignedInt" default="50"/>
      <xs:element name="MaxTimeoutms"
        type="xs:unsignedInt"
        default="60000"/>
      <xs:element name="MaxBatchItems"
        type="xs:unsignedInt"
        default="20"/>
      <xs:element name="SoapTraceEnabled"
        type="xs:boolean"
        default="false"/>
      <xs:element name="MaxProviderRequests"
        type="xs:unsignedInt" default="25"/>
      <xs:element name="Client"
        type="cfg:ClientType"/>
      <xs:element name="Service"
        type="cfg:ServiceType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

MaxEnvelopeSizekb: Maximum SOAP data in kilobytes. The minimum value MUST be 8. The maximum value MUST be 4294967295. This configuration setting is used when processing the wsman:MaxEnvelopeSize header as specified in section [3.1.5.1.10](#).

MaxTimeoutms: The maximum timeout in milliseconds that MUST be used for any request except for Pull requests. The minimum value MUST be 500. The maximum value MUST be 4294967295. This configuration setting is used when processing the wsman:OperationTimeout header as specified in section [3.1.5.1.9](#).

MaxBatchItems: The maximum number of elements that MUST be used in a Pull response. The minimum value MUST be 1. The maximum value MUST be 4294967295. This configuration setting is used when processing Pull messages as specified in section [3.1.5.2.2](#).

SoapTraceEnabled: Enables or disables tracing of SOAP messages in Web Services Management Protocol Extensions for Windows Server 2003 clients and servers. MUST be true or false.

MaxProviderRequests: The maximum number of concurrent requests that MUST be allowed by the Web Services Management Protocol Extensions for Windows Server 2003 servers. The minimum value MUST be 1. The maximum value MUST be 4294967295. The limit is applied per provider. The categories include CIM and configuration. This configuration setting is used when processing messages as specified in section [3.1.5.4.1](#).

Client: MUST contain additional elements to configure Web Services Management Protocol Extensions for Windows Server 2003 clients as specified in section [2.1.3.2](#).

Service: MUST contain additional elements to configure Web Services Management Protocol Extensions for Windows Server 2003 servers as specified in section [2.1.3.3](#).

2.1.3.2 Client Configuration Data Types

This section defines the data types used for configuring Web Services Management Protocol Extensions for Windows Server 2003 clients.

2.1.3.2.1 ClientType

ClientType is the overall container for the Web Services Management Protocol Extensions for Windows Server 2003 client configuration. The schema MUST be as shown below:

```
<xs:complexType name="ClientType">
  <xs:sequence>
    <xs:element name="NetworkDelaysms"
      type="xs:unsignedInt"
      default="5000"/>
    <xs:element name="URLPrefix"
      type="xs:string" default="wsman"/>
    <xs:element name="HTTP"
      type="cfg:ClientHTTPType"/>
    <xs:element name="HTTPS"
      type="cfg:ClientHTTPSType"/>
  </xs:sequence>
</xs:complexType>
```

NetworkDelaysms: Extra time in milliseconds that the Web Services Management Protocol Extensions for Windows Server 2003 client MUST wait to accommodate for network delay time. The minimum value MUST be 500. The maximum value MUST be 4294967295.

URLPrefix: Default **URI** prefix that MUST be used by Web Services Management Protocol Extensions for Windows Server 2003 clients when sending requests. MUST NOT be blank. MUST be a string containing only the following characters a-zA-Z9-0_/_ and MUST NOT start with or end with '/'.

HTTP: MUST contain an additional element to configure behavior for HTTP transport as specified in section [2.1.3.2.3](#).

HTTPS: MUST contain an additional element to configure behavior for HTTPS transport as specified in section [2.1.3.2.4](#).

2.1.3.2.2 ClientUnencryptedType

ClientUnencryptedType is used to configure the authentication mechanisms that are enabled or disabled on the Web Services Management Protocol Extensions for Windows Server 2003 client when using HTTP as the network transport. The schema MUST be as shown below:

```
<xs:complexType name="ClientUnencryptedType">
  <xs:sequence>
    <xs:element name="Basic"
      type="xs:boolean" default="false"/>
    <xs:element name="Digest"
      type="xs:boolean" default="false"/>
    <xs:element name="Negotiate"
      type="xs:boolean" default="false"/>
  </xs:sequence>
</xs:complexType>
```

Basic: Enables or disables HTTP Basic Authentication (as specified in [\[RFC2617\]](#) section 2). MUST be true or false.

Digest: Enables or disables HTTP Digest Authentication (as specified in [\[RFC2617\]](#) section 3). MUST be true or false.

Negotiate: Enables or disables HTTP Negotiate Authentication (as specified in [\[RFC4559\]](#) section 4). MUST be true or false.

These configuration settings MUST be used when sending messages as specified in section [3.1.5.5.2.<6>](#)

2.1.3.2.3 ClientHTTPType

ClientHTTPType is used to configure the Web Services Management Protocol Extensions for Windows Server 2003 client when using HTTP as the network transport. The schema MUST be as shown below:

```
<xs:complexType name="ClientHTTPType">
  <xs:sequence>
    <xs:element name="Port"
      type="xs:unsignedInt" default="80"/>
    <xs:element name="Unencrypted"
      type="cfg:ClientUnencryptedType"/>
  </xs:sequence>
</xs:complexType>
```

Port: Port that MUST be used by the client when using HTTP protocol. The minimum value MUST be 1 and the maximum value MUST be 65535.

Unencrypted: MUST contain an additional element to configure unencrypted authentication as specified in section [3.1.5.5.2](#).

2.1.3.2.4 ClientHTTPSType

ClientHTTPSType is used to configure the authentication mechanisms that are enabled or disabled on the Web Services Management Protocol Extensions for Windows Server 2003 client when using HTTPS as the network transport. The schema MUST be as shown below:

```
<xs:complexType name="ClientHTTPSType">
  <xs:sequence>
    <xs:element name="Port"
      type="xs:unsignedInt" default="443"/>
    <xs:element name="Basic"
      type="xs:boolean" default="true"/>
    <xs:element name="Digest"
      type="xs:boolean" default="true"/>
    <xs:element name="Negotiate"
      type="xs:boolean" default="true"/>
  </xs:sequence>
</xs:complexType>
```

Port: MUST be Port used by the Web Services Management Protocol Extensions for Windows Server 2003 client when using HTTPS protocol. The minimum value MUST be 1. The maximum value MUST be 65535.

Basic: Enables or disables HTTP Basic Authentication (as specified in [\[RFC2617\]](#) section 2). MUST be true or false.

Digest: Enables or disables HTTP Digest Authentication (as specified in [\[RFC2617\]](#) section 3). MUST be true or false.

Negotiate: Enables or disables HTTP Negotiate Authentication (as specified in [\[RFC4559\]](#) section 4). MUST be true or false.

These configuration settings MUST be used when sending messages as specified in section [3.1.5.5.2. <7>](#)

2.1.3.3 Server Configuration Data Types

This section defines the data types used for configuring Web Services Management Protocol Extensions for Windows Server 2003 servers.

2.1.3.3.1 ListenerType

ListenerType is used by Web Services Management Protocol Extensions for Windows Server 2003 servers to listen on one or more IP addresses for WS-Management Protocol requests. ListenerType can be configured for HTTP or HTTPS on a specific IP or on an IP associated with a MAC address. The schema MUST be as shown below:

```
<xs:schema xmlns:cfg="wsman:microsoft.com/wsman/2005/06/config.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="wsman:microsoft.com/wsman/2005/06/config.xsd"
  elementFormDefault="qualified">
  <xs:element name="Listener" type="cfg:ListenerType"/>
```

```

<xs:complexType name="ListenerType">
  <xs:sequence>
    <xs:element name="IP" type="xs:string"/>
    <xs:element name="Port" type="xs:unsignedInt"/>
    <xs:element name="Hostname" type="xs:string"/>
    <xs:element name="Enabled" type="xs:boolean"
default="true"
      minOccurs="0"/>
    <xs:element name="URLPrefix" type="xs:string"
default="wsman"
      minOccurs="0"/>
    <xs:element name="Transport" type="xs:string"/>
    <xs:element name="CertificateThumbprint"
type="xs:string"
      minOccurs="0"/>
    <xs:element name="MACAddress"
type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>

```

IP: IP address for which this listener MUST be created. The value MAY be a static IP address in either IPv4 dotted-decimal format, or in IPv6 colon-delimited hexadecimal format, or the literal string "*".

Port: The TCP port for which this listener MUST be created. The minimum value MUST be 1 and the maximum value MUST be 65535.

Hostname: MUST be the Host name of the machine where the WS-Management service is running.

Enabled: Indicates if the Listener is enabled or disabled.

URLPrefix: MUST be the URI prefix on which to accept HTTP or HTTPS requests. MUST NOT be blank. MUST be a string containing only the following characters a-zA-Z9-0_/ and MUST NOT start with or end with '/'. For example, if the machine name is SampleMachine, then the WS-Management client would specify http://SampleMachine/<URLPrefix> in the destination address.

Transport: The transport used to send and receive WS-Management Protocol requests and responses. The values MUST be either HTTP or HTTPS.

CertificateThumbprint: MUST contain the thumbprint of the certificate as a 40 digit hexadecimal number. This MUST be required if Transport is HTTPS.

MACAddress: MUST be MAC address of NIC (Network Interface Card). Listeners MAY be defined for a specific IP address or for all IP address associated with a specific MAC address. Required if IP is "*". MUST be a sequence of 6 or 8 pairs of case-insensitive hexadecimal digits separated by "-" (e.g 32-a3-58-90-be-cc).

These configuration settings MUST be used when processing messages as described in section [3.1.5.5.4](#).

2.1.3.3.2 ServiceUnencryptedType

ServiceUnencryptedType is used to configure the authentication mechanisms that are enabled or disabled on the Web Services Management Protocol Extensions for Windows Server 2003 server when using HTTP as the network transport. The schema MUST be as shown below:

```
<xs:complexType name="ServiceEncryptedType">
  <xs:sequence>
    <xs:element name="Basic"
      type="xs:boolean" default="false"/>
    <xs:element name="Negotiate"
      type="xs:boolean" default="false"/>
  </xs:sequence>
</xs:complexType>
```

Basic: Enables or disables HTTP Basic Authentication (see [RFC2617](#) section 2).

Negotiate: Enables or disables HTTP Negotiate Authentication (see [RFC4559](#) section 4).

2.1.3.3.3 ServiceHTTPType

ServiceHTTPType is used to configure the authentication mechanisms that are enabled or disabled on the Web Services Management Protocol Extensions for Windows Server 2003 server when using HTTP as the network transport. The schema MUST be as shown below:

```
<xs:complexType name="ServiceHTTPType">
  <xs:sequence>
    <xs:element name="Unencrypted"
      type="cfg:ServiceUnencryptedType"/>
  </xs:sequence>
</xs:complexType>
```

Unencrypted: MUST contain an additional element to configure unencrypted authentication as described in section [3.1.5.5.3](#).

These configuration settings MUST be used when sending messages as specified in section [3.1.5.5.3](#).

2.1.3.3.4 ServiceHTTPSType

ServiceHTTPSType is used to configure the authentication mechanisms that are enabled or disabled on the Web Services Management Protocol Extensions for Windows Server 2003 server when using HTTPS as the network transport. The schema MUST be as shown below:

```
xs:complexType name="ServiceHTTPSType">
  <xs:sequence>
    <xs:element name="Basic"
      type="xs:boolean" default="true"/>
    <xs:element name="Negotiate"
      type="xs:boolean" default="true"/>
  </xs:sequence>
</xs:complexType>
```

Basic: Enables or disables HTTP Basic Authentication (as specified in [RFC2617](#) section 2). MUST be true or false.

Negotiate: Enables or disables HTTP Negotiate Authentication (as specified in [RFC4559](#) section 4). MUST be true or false.

These configuration settings MUST be used when sending messages as specified in section [3.1.5.5.3](#).

2.1.3.3.5 ServiceType

ServiceType is the overall container for the Web Services Management Protocol Extensions for Windows Server 2003 server configuration. Note that listeners are not part of this container and MUST be retrieved separately. The schema MUST be as shown below:

```
<xs:schema xmlns:cfg="wsman:microsoft.com/wsman/2005/06/config.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="wsman:microsoft.com/wsman/2005/06/config.xsd"
  elementFormDefault="qualified">
  <xs:element name="Service" type="cfg:ServiceType"/>
  <xs:complexType name="ServiceType">
    <xs:sequence>
      <xs:element name="RootSDDL" type="xs:string"
        default="O:NSG:BAD:P(A;;GA;;;BA)S:P
(AU;FA;GA;;;WD)
(AU;SA;GWGX;;;WD)"/>
      <xs:element name="MaxConcurrentOperations"
        type="xs:unsignedInt"
        default="100"/>
      <xs:element name="EnumerationTimeoutms"
        type="xs:unsignedInt"
        default="60000"/>
      <xs:element name="MaxClientCertInfoSize"
        type="xs:unsignedInt"
        default="16384"/>
      <xs:element name="MaxConnections"
        type="xs:unsignedInt"
        default="5"/>
      <xs:element name="HTTP"
        type="cfg:ServiceHTTPType"/>
      <xs:element name="HTTPS"
        type="cfg:ServiceHTTPSType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

RootSDDL: The security descriptor controls remote access to the listener. The string format of RootSDDL MUST use the syntax defined by the Security Descriptor Definition Language (as specified in [MS-DTYP](#)). This configuration setting MUST be used when processing messages as specified in section [5.1](#).

MaxConcurrentOperations: MUST be the maximum number of concurrent Enumeration operations allowed. The minimum value MUST be 1. The maximum value MUST be 4294967295. This configuration setting MUST be used when processing messages as specified in section [3.1.5.4.1](#).

EnumerationTimeoutms: MUST be the idle timeout in milliseconds between Pull messages. The minimum value MUST be 500. The maximum value MUST be 4294967295. This configuration setting MUST be used when processing messages as specified in section [3.1.6](#).

MaxConnections: MUST be the maximum number of active requests that the service can process simultaneously. The minimum value MUST be 1. The maximum value MUST be 50. This configuration setting MUST be used when processing messages as specified in section [3.1.5.4.1](#).

HTTP: MUST contain an additional element to configure HTTP transport as specified in section [2.1.3.3.3](#).

HTTPS: MUST contain an additional element to configure HTTPS transport as specified in section [2.1.3.3.3](#).

2.1.3.4 Miscellaneous Data Types

2.1.3.4.1 ThisType

ThisType is used by the Web Services Management Protocol Extensions for Windows Server 2003 server to send information about the vendor and version of the protocol stack. The schema MUST be as shown below:

```
<xs:schema xmlns:t="http://schemas.xmlsoap.org/ws/2005/06/management/this" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://schemas.xmlsoap.org/ws/2005/06/management/this" elementFormDefault="qualified">
  <xs:element name="This"/>
  <xs:complexType name="ThisType">
    <xs:sequence>
      <xs:element name="Vendor" type="xs:string"/>
      <xs:element name="Version" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Vendor: MUST be a vendor name. [<8>](#)

Version: MUST be a string that describes the version of the WS-Management Protocol stack. This is intended for diagnostic purposes only and not for version negotiation. [<9>](#)

3 Protocol Details

The following sections specify protocol details with extensions to the message processing [CIM Mapping details](#) and [CIM Binding details](#) for WS-Management Extensions.

3.1 WS-Management Details

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), describes a general SOAP-based protocol for managing systems such as PCs, servers, devices, Web services and other applications, and other **manageable entities**.

This section describes changes to the WS-Management Protocol for Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

3.1.1 Abstract Data Model

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the abstract data model of the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#).

3.1.2 Timers

Web Services Management Protocol Extensions for Windows Server 2003 defines one timer in addition to the timers of the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#).

- The [Enumeration Garbage Collection timer](#) MUST trigger cleanup of the state associated with an enumeration, if a client has not used it recently.

3.1.3 Initialization

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the initialization of the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#).

3.1.4 Higher-Layer Triggered Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the higher-layer events of the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#).

3.1.5 Message Processing

This section describes changes made by Web Services Management Protocol Extensions for Windows Server 2003 to the message processing of the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#).

3.1.5.1 Common Headers

3.1.5.1.1 XML Namespace Definitions

The XML namespace for the WS-Management Protocol is specified in [\[DMTF-DSP0226\]](#) section 1.5, as <http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd>. Web Services Management Protocol Extensions for Windows Server 2003 replaces <http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd> with <http://schemas.xmlsoap.org/wsman/2005/06/management> as the XML namespace for WS-Management Protocol.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST use the <http://schemas.xmlsoap.org/wsman/2005/06/management> prefix instead of the <http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd>.

Note XML namespaces are to be treated as identifiers and they do not point to any location on the Web.

3.1.5.1.2 wsman:SelectorSet

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines **selector** as a resource-relative name and value pair that acts as an instance-level discriminant. SelectorSet MUST be a set of these selectors as specified in [\[DMTF-DSP0226\]](#) section 2.1.2. Rule R2.1.2-1 specifies that the selector names and values MAY be treated as case-insensitive or case-sensitive.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST treat all selector names and values as case insensitive.

3.1.5.1.3 wsa:ReplyTo

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), allows the ReplyTo Header value to be either a valid address for a new connection using any transport supported by the service, or the URI <http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous>, as specified in [\[DMTF-DSP0226\]](#) section 2.5.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST set the ReplyTo Header value to <http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous>.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST return a wsman:UnsupportedFeature fault with a detail code of wsman:faultDetail/AddressingMode if they receive a value other than <http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous> in the ReplyTo header.

3.1.5.1.4 wsa:FaultTo

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), allows a conformant service to require that all faults be delivered to the client over the same transport or connection on which the request arrives. In this case, the URI MUST be <http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous>, as specified in [\[DMTF-DSP0226\]](#) section 2.6.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT set the wsa:FaultTo element.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST return the wsman:UnsupportedFeature fault with a detail code of wsman:faultDetail/AddressingMode if the wsa:Address element within the wsa:FaultTo is not set to <http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous>.

3.1.5.1.5 wsa:MessageID

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), endorses two different MessageID URI formats, as specified in [\[DMTF-DSP0226\]](#) section 2.7.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST use only the uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx MessageID format.

3.1.5.1.6 wsa:Action

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines all the **Action URIs** that indicate the method being invoked against the resource, as specified in [\[DMTF-DSP0226\]](#) section 2.8.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers **MUST** support the following Action URIs for operations related to CIM data:

```
http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
http://schemas.xmlsoap.org/ws/2004/09/transfer/Put
http://schemas.xmlsoap.org/ws/2004/09/transfer/PutResponse
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release
http://schemas.xmlsoap.org/ws/2004/09/enumeration/ReleaseResponse
```

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers **MUST** support the following additional Action URIs for operations related to Web Services Management Protocol Extensions for Windows Server 2003 configuration data:

```
http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse
http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete
http://schemas.xmlsoap.org/ws/2004/09/transfer/DeleteResponse
```

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers **MAY** support the following additional Action URIs for operations related to subscriptions: [<10>](#)

```
http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscribeResponse
http://schemas.xmlsoap.org/ws/2004/08/eventing/Unsubscribe
http://schemas.xmlsoap.org/ws/2004/08/eventing/UnsubscribeResponse
```

3.1.5.1.7 wsa:To

The WS-Management Protocol, as specified in section 2.4 of [\[DMTF-DSP0226\]](#), defines *wsa:To* address as the network address of the service.

3.1.5.1.8 wsa:Relates To

The WS-Management Protocol, as specified in section 2.4 of [\[DMTF-DSP0226\]](#), endorses two different *Relates To* URI formats. For more details, see section 2.7 of [\[DMTF-DSP0226\]](#). Web Services Management Protocol Extensions for Windows Server 2003 clients and servers **MUST** use only the **uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx** *Relates To* format.

3.1.5.1.9 wsman:OperationTimeout

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines the OperationTimeout value to indicate that the clients expect a response or a fault within the specified time, as specified in [\[DMTF-DSP0226\]](#) section 3.1.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST set an OperationTimeout value with the value of the MaxTimeoutms configuration setting defined in section [2.1.3.1](#).

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST set the default timeout value to the value of the MaxTimeoutms configuration setting defined in section [2.1.3.1](#) if no OperationTimeout value is specified by the client or if the OperationTimeout value is more than this setting. <11>

3.1.5.1.10 wsman:MaxEnvelopeSize

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines the MaxEnvelopeSize value to indicate that the clients expect a response to be no larger than the given number of octets, as specified in [\[DMTF-DSP0226\]](#) section 3.2.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST set an MaxEnvelopeSize value with the value of the MaxEnvelopeSizekb configuration setting (defined in section [2.1.3.1](#)) multiplied by 1024.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST set the default MaxEnvelopeSize value to the value of the MaxEnvelopeSizekb configuration setting (defined in section [2.1.3.1](#)) multiplied by 1024 if no MaxEnvelopeSize value is specified by the client or if the MaxEnvelopeSize value is more than this setting.

WS-Management [\[DMTF-DSP0226\]](#) section 3.2 indicates that servers SHOULD return a wsman:EncodingLimit fault if the value of wsman:MaxEnvelopeSize is less than 8192 octets. Web Services Management Protocol Extensions for Windows Server 2003 servers MUST return the indicated fault in this situation. <12>

3.1.5.1.11 wsman:Locale

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines the Locale element, which specifies the language in which the client wants response text to be translated, as specified in [\[DMTF-DSP0226\]](#) section 3.3.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT set the **mustUnderstand** attribute of this element to "true".

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST return a fault of wsman:UnsupportedFeature with a detail code of wsman:faultDetail/Locale if the **mustUnderstand** attribute of this element is "true".

3.1.5.1.12 wsman:OptionSet

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines the OptionSet element as a set of switches to the service to modify or refine the nature of the request, as specified in [\[DMTF-DSP0226\]](#) section 3.4.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT use the OptionSet element.

Web Services Management Protocol Extensions for Windows Server 2003 servers SHOULD ignore the OptionSet element. [<13>](#)

3.1.5.1.13 wsman:RequestEPR

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines the RequestEPR SOAP header, which can be used by clients that need to get the **EPR** in the response, as specified in [\[DMTF-DSP0226\]](#) section 3.5.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT use the RequestEPR header.

Web Services Management Protocol Extensions for Windows Server 2003 servers SHOULD ignore the RequestEPR header. [<14>](#)

3.1.5.2 Enumeration

3.1.5.2.1 wsen:Filter

The WS-Management specification defines the Filter element, which is used to retrieve a subset of the result set, as specified in [\[DMTF-DSP0226\]](#) section 5.2.1.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST fault if the request contains the Filters element. [<15>](#)

3.1.5.2.2 wsen:Pull/wsen:MaxElements

The WS-Management specification defines the MaxElements element, which is used to limit how many items are retrieved in a single message, as specified in [\[DMTF-DSP0226\]](#) section 5.4.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST use the value of MaxBatchItems configuration setting (defined in section [2.1.3.1](#)) as the value of MaxElements when sending Pull requests.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST use the smaller value of the MaxBatchItems configuration setting (defined in section [2.1.3.1](#)) and the value of wsen:MaxElements as the effective value of wsen:MaxElements. So, the maximum number of items that are returned in a single Pull response message is determined by the smaller of the MaxBatchItems configuration setting on the server and the value of the wsen:MaxElements element in the Pull request message.

3.1.5.2.3 wsman:RequestTotalItemsCountEstimate

The WS-Management specification defines the RequestTotalItemsCountEstimate SOAP header to allow a client to request an estimate for the number of items being returned in the result set, as specified in [\[DMTF-DSP0226\]](#) section 5.2.2.

Web Services Management Protocol Extensions for Windows Server 2003 client SHOULD NOT use the RequestTotalItemsCountEstimate header.

Web Services Management Protocol Extensions for Windows Server 2003 servers SHOULD ignore the RequestTotalItemsCountEstimate header. [<16>](#)

3.1.5.2.4 wsman:OptimizeEnumeration

The WS-Management specification defines the optional element OptimizeEnumeration to allow a client to request that initial results be returned in the enumeration response, as specified in [\[DMTF-DSP0226\]](#) section 5.2.3.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT use the OptimizeEnumeration element.

Web Services Management Protocol Extensions for Windows Server 2003 servers SHOULD ignore the OptimizeEnumeration element.[<17>](#)

3.1.5.2.5 wsman:EnumerationMode

The WS-Management specification defines the optional EnumerationMode element to allow a client to specify whether the actual objects, the EPR of the object, or both the EPR and the object should be returned, as specified in [\[DMTF-DSP0226\]](#) section 5.7.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT use the EnumerationMode element.

Web Services Management Protocol Extensions for Windows Server 2003 servers SHOULD ignore the EnumerationMode element.[<18>](#)

3.1.5.2.6 wsman:Filter

The WS-Management specification defines wsman:Filter as an alternative mechanism to specify filters that are different from that of the [wsen:Filter](#), as specified in [\[DMTF-DSP0226\]](#) section 5.3.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT use the wsman:Filter element.

Web Services Management Protocol Extensions for Windows Server 2003 servers SHOULD ignore the wsman:Filter element.[<19>](#)

3.1.5.3 Transfer Get and Put

3.1.5.3.1 wsman:FragmentTransfer

The WS-Management Protocol, as specified in [\[DMTF-DSP0226\]](#), defines the FragmentTransfer SOAP header, which is used to retrieve and update fragments or individual elements of a CIM object, as specified in [\[DMTF-DSP0226\]](#) section 4.8.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT use the FragmentTransfer header.

Web Services Management Protocol Extensions for Windows Server 2003 servers SHOULD ignore the FragmentTransfer header.[<20>](#)

3.1.5.4 Miscellaneous

3.1.5.4.1 Concurrent Operations

A Web Services Management Protocol Extensions for Windows Server 2003 server MUST reject additional requests if it is already processing a number of concurrent requests equal to the MaxConnections configuration setting defined in section [2.1.3.3](#).

A Web Services Management Protocol Extensions for Windows Server 2003 server MUST reject additional Enumerate requests if it has a number of outstanding enumerations equal to the MaxConcurrentOperations configuration setting defined in section [2.1.3.3](#).

A Web Services Management Protocol Extensions for Windows Server 2003 server MUST reject additional requests to a specific provider if the provider is already processing a number of concurrent requests equal to the MaxProviderRequests configuration setting defined in section [2.1.3.3](#).

3.1.5.4.2 Inbound Message Size

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST return an HTTP status of 413 (Request Entity Too Large) without processing the SOAP message if the request packet from the client is larger than the MaxEnvelopeSizekb configuration setting defined in section [2.1.3.1](#).

3.1.5.4.3 Fault Detail

The WS-Management specification allows servers to specify additional fault details as part of SOAP faults that it generates, as specified in [\[DMTF-DSP0226\]](#) section 11. The URI prefix for fault detail is <http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail>.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST use the wsman:faultDetail prefix instead of the <http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail> prefix.

Note XML namespaces are to be treated as identifiers and they do not point to any location on the Web.

3.1.5.4.4 Metadata and Discovery

The WS-Management specification defines a mechanism to determine the existence of a WS-Management service on a server, as specified in [\[DMTF-DSP0226\]](#) section 8.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers SHOULD NOT use the WS-Management discovery mechanism specified in [\[DMTF-DSP0226\]](#) section 8.

To determine if a server supports the Web Services Management Protocol Extensions for Windows Server 2003 Protocol, Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD send a WS-Management Get operation using the wsman:system/2005/06/this resource URIs with no selectors.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST respond to this Get request with a [ThisType](#) response as defined in section [2.1.3.4.1](#), [<21>](#) [<22>](#)

3.1.5.4.5 Binary Attachments

The WS-Management Protocol defines a mechanism to send binary attachments, as specified in [\[DMTF-DSP0226\]](#) section 10.6.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST NOT send binary attachments.

3.1.5.5 Configuration

The remote configuration of Web Services Management Protocol Extensions for Windows Server 2003 clients and servers can be performed through a series of Get, Put, Create, Delete, and Enumeration operations addressed to a set of resource URIs defined in this section. In this section, wherever clients and servers are mentioned, they refer to Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

3.1.5.5.1 wsman:microsoft.com/wsman/2005/06/config

The wsman:microsoft.com/wsman/2005/06/config resource URI MUST be used to retrieve the complete configuration of Web Services Management Protocol Extensions for Windows Server 2003 clients and servers. The configuration is grouped under separate XML elements; further URIs are exposed to allow easier and more finely-grained levels of retrieval and updates.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that is passed as part of the request or response is referenced:

Operation	Input Data Type	Output Data Type
Get	None	cfg:ConfigType
Put	cfg:ConfigType	cfg:ConfigType

3.1.5.5.2 wsman:microsoft.com/wsman/2005/06/config/client

The wsman:microsoft.com/wsman/2005/06/config/client resource URI MUST be used to configure the client.

This URI refers to a singleton instance of data, and so no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that is passed as part of the request or response is referenced:

Operation	Request data type	Response data type
Get	None	cfg:ClientType
Put	cfg:ClientType	cfg:ClientType

3.1.5.5.2.1 wsman:microsoft.com/wsman/2005/06/config/client/http

The wsman:microsoft.com/wsman/2005/06/config/client/http resource URI MUST be used to configure the HTTP-related parameters related to the client configuration defined in section [2.1.3.2.3](#). This includes the protocol and authentication schemes that MAY be used for a client-initiated operation over the HTTP protocol. If a client application tries to use an authentication scheme that is not enabled, the request MUST fail with an error.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that is passed as part of the request or response is referenced:

Operation	Request Data Type	Response Data Type
Get	None	cfg:ClientHTTPType
Put	cfg:ClientHTTPType	cfg:ClientHTTPType

3.1.5.5.2.2

wsman:microsoft.com/wsman/2005/06/config/client/http/unencrypted

The wsman:microsoft.com/wsman/2005/06/config/client/http/unencrypted resource URI MUST be used to configure the authentication mechanisms supported by a client when using HTTP. If a client application tries to use an authentication scheme that is not enabled, the request MUST fail with an error.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced:

Operation	Request Data Type	Response Data Type
Get	None	cfg:ClientUnencryptedType
Put	cfg:ClientUnencryptedType	cfg:ClientUnencryptedType

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST use wsman:secprofile/http/basic as an authentication scheme if and only if the Basic property in the cfg:ClientUnencryptedType is true.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST use wsman:secprofile/http/digest as an authentication scheme if and only if the Digest property in the cfg:ClientUnencryptedType is true.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST use wsman:secprofile/http/spnego-kerberos basic as an authentication scheme if and only if the Negotiate property in the cfg:ClientUnencryptedType is true.

3.1.5.5.2.3 wsman:microsoft.com/wsman/2005/06/config/client/https

The wsman:microsoft.com/wsman/2005/06/config/client/https resource URI MUST be used to configure the authentication mechanisms supported by a client when using HTTPS. If a client application tries to use an authentication scheme that is not enabled, the request MUST fail with an error.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced:

Operation	Request Data Type	Response Data Type
Get	None	cfg:ClientHTTPSType
Put	cfg:ClientHTTPSType	cfg:ClientHTTPSType

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST use wsman:secprofile/https/basic as an authentication scheme if and only if the Basic property in the cfg:ClientHTTPSType is true.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST use wsman:secprofile/https/digest as an authentication scheme if and only if the Digest property in the cfg:ClientHTTPSType is true.

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST support wsman:secprofile/https/spnego-kerberos as an authentication scheme if and only if the Negotiate property in the cfg:ClientHTTPSType is true.

3.1.5.5.3 wsman:microsoft.com/wsman/2005/06/config/service

The wsman:microsoft.com/wsman/2005/06/config/service resource URI MUST be used to configure the server.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced:

Operation	Request Data Type	Response Data Type
Get	None	cfg:ServiceType
Put	cfg:ServiceType	cfg:ServiceType

3.1.5.5.3.1 wsman:microsoft.com/wsman/2005/06/config/service/http

The wsman:microsoft.com/wsman/2005/06/config/service/http resource URI MUST be used to configure the authentication mechanisms supported by a server when using HTTP. If a client application tries to use an authentication scheme that is not enabled on the server, the request MUST fail with an error.

If a client application tries to use the digest authentication scheme, which is not implemented on the server, the request MUST fail with an error.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced:

Operation	Request data type	Response data type
Get	None	cfg:ServiceHTTPType
Put	cfg:ServiceHTTPType	cfg:ServiceHTTPType

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST use wsman:secprofile/http/basic as an authentication scheme if and only if the Basic property in the cfg:ServiceHTTPType is true. [<23>](#)

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST use wsman:secprofile/http/spnego-kerberos as an authentication scheme if and only if the Negotiate property in the cfg:ServiceHTTPType is true.

3.1.5.5.3.2

wsman:microsoft.com/wsman/2005/06/config/service/http/unencrypted

The wsman:microsoft.com/wsman/2005/06/config/service/http/unencrypted resource URI MUST be used to configure the authentication mechanisms supported by a server when using HTTP. If a client application tries to use an authentication scheme that is not enabled on the server, the request MUST fail with an error.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced:

Operation	Request data type	Response data type
Get	None	cfg:ServiceUnencryptedType
Put	cfg:ServiceUnencryptedType	cfg:ServiceUnencryptedType

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST use wsman:secprofile/http/basic as an authentication scheme if and only if the Basic property in the cfg:ServiceHTTPType is true. [<24>](#)

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST use wsman:secprofile/http/spnego-kerberos as an authentication scheme if and only if the Negotiate property in the cfg:ServiceHTTPType is true.

3.1.5.5.3.3 wsman:microsoft.com/wsman/2005/06/config/service/https

The wsman:microsoft.com/wsman/2005/06/config/service/https resource URI MUST be used to configure the authentication mechanisms supported by a server when using HTTPS. If a client application tries to use an authentication scheme that is not enabled on the server, the request MUST fail with an error.

If a client application tries to use the digest authentication scheme, which is not implemented on the server, the request MUST fail with an error.

This URI refers to a singleton instance of data, therefore no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced:

Operation	Request Data Type	Response Data Type
Get	None	cfg:ServiceHTTPSType
Put	cfg:ServiceHTTPSType	cfg:ServiceHTTPSType

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST use wsman:secprofile/https/basic as an authentication scheme if and only if the Basic property in the cfg:ServiceHTTPSType is true. [<25>](#)

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST use wsman:secprofile/https/spnego-kerberos as an authentication scheme if and only if the Negotiate property in the cfg:ServiceHTTPSType is true.

3.1.5.5.4 wsman:microsoft.com/wsman/2005/06/config/listener

The wsman:microsoft.com/wsman/2005/06/config/listener resource URI MUST be used for configuring the server to listen on the network for WS-Management requests. By default, the server is configured with no listeners resulting in no remote configuration of the machine using WS-Management. This means that no remote configuration can be done initially until some form of configuration is performed locally.

3.1.5.5.4.1 Enumeration of Listeners

Enumeration can be used to retrieve all listeners configured on the server. The IP and Port properties that are returned with each of the objects can be used as selectors to address the specific configuration item for updates.

This URI refers to all instances of the listeners, and so no additional selectors are needed to address this resource.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced:

Operation	Request Data Type	Response Data Type
Enumerate	None	None
Pull	None	cfg:ListenerType

3.1.5.5.4.2 Retrieval and Modification of Individual Listeners

To retrieve and configure an individual listener, the listener MUST need to be referred by a selector. The following properties, which are part of cfg:ListenerType, are the selectors.

Selector Name	Description
IP	The IP address that the server is configured to listen on.
Port	The Port that the server is configured to listen on.

The set of operations that Web Services Management Protocol Extensions for Windows Server 2003 servers MUST support on this URI are defined in the following table and, where relevant, the XSD type for the data that MUST be passed as part of the request or response is referenced.

Operation	Request Data Type	Response Data Type
Get	None	cfg:ListenerType
Put	cfg:ListenerType	cfg:ListenerType
Delete	None	None
Create	Not Supported	Not Supported

If and only if the Enabled property is true, Web Services Management Protocol Extensions for Windows Server 2003 servers MUST listen on the network on the port given by the Port property and MUST only process requests sent to a configured destination IP address and addressed to the path given by the URIPrefix property.

When considering destination IP addresses, Web Services Management Protocol Extensions for Windows Server 2003 servers MUST listen on a specific IP address (if the IP address property is a valid IP address) or on all IP addresses associated with the value of the MACAddress property (if the IP address property is "*").

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST return a SOAP fault in response to a Put request if the Transport property is HTTPS and the certificate identified by the CertificateThumbprint property does not exist or the Common Name in the certificate does not match the Hostname property.

3.1.6 Timer Events

Web Services Management Protocol Extensions for Windows Server 2003 defines one timer in addition to the events of the WS-Management Protocol timer as specified in [\[DMTF-DSP0226\]](#).

3.1.6.1 Enumeration Garbage Collection Timer

The Enumeration Garbage Collection timer MUST be started by the Web Services Management Protocol Extensions for Windows Server 2003 server when it sends an EnumerationResponse or a PullResponse message. There MUST be a unique timer for each enumeration. Upon receipt of a Pull or Release request, the Enumeration Garbage Collection timer for that enumeration MUST be canceled.

The Enumeration Garbage Collection timer MUST expire after the number of milliseconds given by the EnumerationTimeoutms configuration setting defined in section [2.1.3.3](#). Upon expiration of this timer, the Web Services Management Protocol Extensions for Windows Server 2003 server MUST return a wsen:InvalidEnumerationContext fault in response to a Pull or Release request that contains the enumeration context value of the last PullResponse message or the EnumerateResponse if no PullResponse messages were sent.

3.1.7 Other Local Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the other local events of the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#).

3.2 WS-CIM Mapping Details

The WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#) defines a normative description of a protocol-independent mapping of CIM models to XML Schema, WSDL fragments, and metadata fragments.

This section describes changes to the WS-CIM Mapping Specification for Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

3.2.1 Abstract Data Model

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the abstract data model of the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#).

3.2.2 Timers

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the timers of the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#).

3.2.3 Initialization

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the initialization of the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#).

3.2.4 Higher-Layer Triggered Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the higher-layer triggered events of the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#).

3.2.5 Message Processing

This section describes changes made by Web Services Management Protocol Extensions for Windows Server 2003 to the message processing of the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#).

3.2.5.1 Embedded Objects

The WS-CIM Mapping Specification defines how one object can be embedded in another object, as specified in [\[DMTF-DSP0230-NM\]](#) section 7.2.5.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST NOT send embedded objects as attachments.

3.2.5.2 CIM XML Namespace

The WS-CIM Mapping Specification defines the CIM XML namespace as <http://schemas.dmtf.org/wbem/wscim/1/common>, as specified in [\[DMTF-DSP0230-NM\]](#) section 4.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST use <http://schemas.dmtf.org/wsman/2005/06/base> as the **CIM namespace** instead of <http://schemas.dmtf.org/wbem/wscim/1/common>.

Note XML namespaces are to be treated as identifiers and they do not point to any location on the Web.

3.2.5.3 Arrays

The WS-CIM Mapping Specification defines specific rules for mapping CIM properties that are arrays, as specified in [\[DMTF-DSP0230-NM\]](#) section 7.2.2.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST NOT send null array elements. Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST indicate null arrays by including the element one time with the xsi:nil attribute set to "true".

3.2.5.4 cim:Location

The WS-CIM Mapping Specification defines rules for representing a CIM object in XML, as specified in [\[DMTF-DSP0230-NM\]](#) section 7.

Web Services Management Protocol Extensions for Windows Server 2003 extends the object XML with an extra child element that describes the EPR of the object.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST add a cim:Location element of type wsa:EndpointReferenceType, as specified in [\[WSAddressing\]](#) section 2.2, specifying the EPR of an object to the instance element when responding to Get requests. Web Services Management Protocol Extensions for Windows Server 2003 servers MUST ignore the cim:Location when processing a Put request.

Web Services Management Protocol Extensions for Windows Server 2003 clients MAY remove the cim:Location element when requesting a Put operation following a Get operation on an object. [<26>](#)

3.2.6 Timer Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the timer events of the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#).

3.2.7 Other Local Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the other local events of the WS-CIM Mapping Specification as specified in [\[DMTF-DSP0230-NM\]](#).

3.3 WS-Management - CIM Binding Details

The WS-Management CIM Binding Specification [\[DMTF-DSP0227-NM\]](#) describes how transformed CIM resources, as specified by the WS-CIM specification, are bound to WS-Management operations and WSDL definitions.

This section describes changes to the WS-Management CIM Binding Specification for Web Services Management Protocol Extensions for Windows Server 2003 clients and servers.

3.3.1 Abstract Data Model

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the abstract data model of the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#).

3.3.2 Timers

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the timers of the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#).

3.3.3 Initialization

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the initialization of the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#).

3.3.4 Higher-Layer Triggered Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the higher-layer triggered events of the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#).

3.3.5 Message Processing

This section describes changes made by Web Services Management Protocol Extensions for Windows Server 2003 to the message processing of the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#).

3.3.5.1 wsmb:PolymorphismMode

A common way to extend CIM classes is to define derivatives of the CIM class. When a client requests objects of the type for CIM_Process, it is possible to return instances that are actually of a derived type such as Vendor_Process. The WS-Management CIM Binding Specification defines details for handling polymorphism in the resultSet, as specified in [\[DMTF-DSP0227-NM\]](#) section 8.1.

Web Services Management Protocol Extensions for Windows Server 2003 clients SHOULD NOT send the PolymorphismMode element.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST ignore the PolymorphismMode element.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST return instances of both base and derived classes by effectively casting derived objects to the base class. Each returned instance MUST contain only the properties of the base class and omit the properties from the derived classes.

3.3.5.2 CIM Namespace

The CIM binding for WS-Management defines Resource URIs to be used in referencing the CIM objects, as specified in [\[DMTF-DSP0227-NM\]](#) section 5.1.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST use the `http://schemas.dmtf.org/wsman/2005/06/cimv2.9/` namespace prefix followed by the class name when accessing DMTF classes whose CIM namespace is `root\hardware`.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST use the following format when accessing all other classes:

`<prefix> <cim namespace> "/" <class name>`

Where `<prefix>` is `http://schemas.microsoft.com/wsman/2005/06/wmi/` and `<cim namespace>` is the CIM namespace in which `"\"` has been converted to `"/"`.

[\[DMTF-DSP0227-NM\]](#) section 5.3 defines the "*cimnamespace*" selector to specify the CIM namespace that the request is associated with.

Web Services Management Protocol Extensions for Windows Server 2003 clients and servers MUST NOT use the "*cimnamespace*" selector.

Note XML namespaces are to be treated as identifiers and they do not point to any location on the Web.

3.3.6 Timer Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the timer events of the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#).

3.3.7 Other Local Events

Web Services Management Protocol Extensions for Windows Server 2003 includes no changes to the other local events of the WS-Management CIM Binding Specification as specified in [\[DMTF-DSP0227-NM\]](#).

4 Protocol Examples

4.1 CIM examples

This section illustrates protocol examples related to CIM.

4.1.1 Retrieving a CIM Instance

This section illustrates an example of a simple CIM class being accessed using the WS-Management Protocol.

MOF representation of the class as specified in [\[DMTF-DSP004\]](#):

```
[abstract]
class Base
{
[key] sint32 id;
};

class MyClass : Base
{
string Data1;
};

instance of MyClass
{
id = 1;
Data1 = "Hello World";
};
```

This is a simple CIM class hierarchy of two classes, a base CIM class, and a derived CIM class called MyClass. These classes are defined in the CIM namespace called root\mycimnamespace.

The request to access an instance of this class using a Get operation is shown below:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/
soap-envelope"
  xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
  xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
  <s:Header>
    <a:To>http://server:80/wsman</a:To>
    <w:ResourceURI s:mustUnderstand="true">
http://schemas.microsoft.com/
wsman/2005/06/wmi/root/mynamespace/myclass
</w:ResourceURI>
    <a:ReplyTo>
    <a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/08/addressing/role/anonymous</a:Address>
</a:ReplyTo>
    <a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
```

```

2004/09/transfer/Get</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200</w:MaxEnvelopeSize>
<a:MessageID>
uuid:5E6FD101-710A-4EEA-A50D-70C0BF863AA3
</a:MessageID>
<w:SelectorSet>
<w:Selector Name="id">1</w:Selector>
</w:SelectorSet>
<w:OperationTimeout>PT60.000S</w:OperationTimeout>
</s:Header>
<s:Body/>
</s:Envelope>

```

Get Response:

```

<s:Envelope xml:lang="en-US" xmlns:
s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/09/transfer/
GetResponse</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:2DAB718A-0103-4E0A-AB17-06C8A5530D2B
</a:MessageID>
<a:To>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous
</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:5E6FD101-710A-4EEA-A50D-70C0BF863AA3
</a:RelatesTo>
</s:Header>
<s:Body>
<p:myclass xmlns:
p="http://schemas.microsoft.com/wsman/2005/06/wmi/
root/mynamespace/myclass">
<p:Data1>Hello World</p:Data1>
<p:id>1</p:id>
<cim:Location xmlns:
cim="http://schemas.dmtf.org/wsman/2005/06/
base" xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<a:Address>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/
anonymous</a:Address>
<a:ReferenceParameters>
<w:ResourceURI>http://schemas.microsoft.com/
wsman/2005/06/wmi/root/
mynamespace/myclass</w:ResourceURI>

```



```

<w:SelectorSet>
<w:Selector Name="id">1</w:Selector>
</w:SelectorSet>
</a:ReferenceParameters>
</cim:Location>
</p:myclass>
</s:Body>
</s:Envelope>

```

4.1.2 Enumeration of Instances

If there are multiple instances of a class, then Enumeration can be used to retrieve all the instances of the CIM class. The example from section [4.1.1](#) can be extended to add another instance as shown below:

[C++]

```

[abstract]
class Base
{
[key] sint32 id;
};

class MyClass : Base
{
string Data1;
};

instance of MyClass
{
id = 1;
Data1 = "Hello World";
};

instance of MyClass
{
id = 2;
Data1 = "Hello Again";
};

```

Enumeration involves multiple requests and response exchanges as shown below:

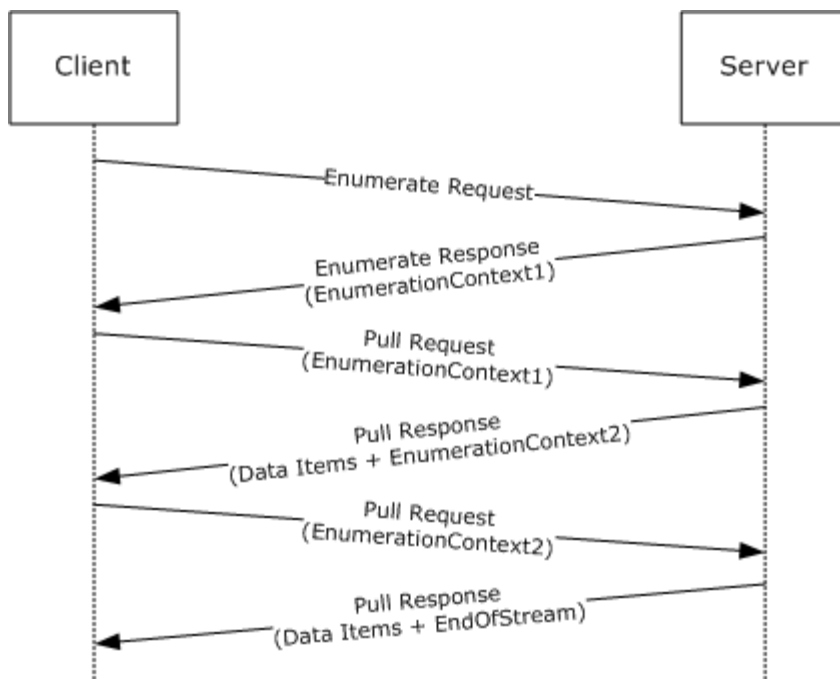


Figure 2: Web Services Management Request-and-Response Enumeration

1. Client sends an [Enumerate request](#) with the Resource URI of the CIM class.
2. Server responds with an [Enumerate Response](#) that contains an Enumeration Context.
3. Client sends a Pull request and includes the Enumeration Context returned in the Enumerate Response.
4. Server responds with one or more instances of the CIM class along with a new Enumeration Context. The number of instances is determined by maxElements specified in the Pull request.
5. Client sends a Pull request and includes the Enumeration Context received in the previous Pull response.
6. Server responds with one or more instances of the CIM class along with a new Enumeration Context.
7. This sequence is repeated until the server sends an EndofSequence, which indicates there are no more instances.
8. Client can send a Release request at any time during the enumeration to stop the exchange.

4.1.2.1 Enumerate Request

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/
soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/
enumeration"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/

```

```

management">
<s:Header>
<a:To>http://server:80/wsman</a:To>
<w:ResourceURI s:mustUnderstand="true">
http://schemas.microsoft.com/
wsman/2005/06/wmi/root/mynamespace/myclass
</w:ResourceURI>
<a:ReplyTo>
<a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/08/addressing/role/anonymous</a:Address>
</a:ReplyTo>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/09/enumeration/Enumerate</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200
</w:MaxEnvelopeSize>
<a:MessageID>
uuid:C61CA1DC-51C0-4353-AE46-3E42ED0DA794
</a:MessageID>
<w:OperationTimeout>PT60.000S</w:OperationTimeout>
</s:Header>
<s:Body>
<n:Enumerate/>
</s:Body>
</s:Envelope>

```

4.1.2.2 Enumerate Response

```

<s:Envelope xml:lang="en-US"
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/
enumeration"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/09/enumeration/
EnumerateResponse
</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:95783CED-6AC4-471B-B773-1CC892FC674B
</a:MessageID>
<a:To>
http://schemas.xmlsoap.org/ws/2004/08/addressing/role/
anonymous</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:C61CA1DC-51C0-4353-AE46-3E42ED0DA794</a:RelatesTo>
</s:Header>
<s:Body>
<n:EnumerateResponse>
<n:EnumerationContext>
uuid:22EB9809-5543-4020-A75C-FD95FF06217B

```

```

</n:EnumerationContext>
</n:EnumerateResponse>
</s:Body>
</s:Envelope>

```

4.1.2.3 First Pull Request

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/
soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/
enumeration"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:To>http://server:80/wsman</a:To>
<w:ResourceURI s:mustUnderstand="true">
http://schemas.microsoft.com/wsman/2005/06/wmi/
root/mynamespace/
myclass</w:ResourceURI>
<a:ReplyTo>
<a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/08/addressing/
role/anonymous</a:Address>
</a:ReplyTo>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200
</w:MaxEnvelopeSize>
<a:MessageID>
uuid:54E3FD6C-A83E-454C-A2F6-0BDABF5F14D7</a:MessageID>
<w:OperationTimeout>PT60.000S</w:OperationTimeout>
</s:Header>
<s:Body>
<n:Pull>
<n:EnumerationContext xmlns:n="
http://schemas.xmlsoap.org/ws/2004/09/enumeration">
uuid:22EB9809-5543-4020-A75C-FD95FF06217B
</n:EnumerationContext>
<n:MaxElements>1</n:MaxElements>
</n:Pull>
</s:Body>
</s:Envelope>

```

4.1.2.4 First Pull Response

```

<s:Envelope xml:lang="en-US"
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"

```

```

xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/
enumeration"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/09/enumeration/
PullResponse
</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:21E59CC8-6D5E-4072-BCA2-7C0DC2BC2504
</a:MessageID>
<a:To>
http://schemas.xmlsoap.org/ws/2004/08/addressing/role/
anonymous</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:54E3FD6C-A83E-454C-A2F6-0BDABF5F14D7</a:RelatesTo>
</s:Header>
<s:Body>
<n:PullResponse>
<n:EnumerationContext>
uuid:2504CA0D-94B9-4F91-B2F7-9F4CD9A2A96C
</n:EnumerationContext>
<n:Items>
<p:myclass
xmlns:p="http://schemas.microsoft.com/wsman/2005/
06/wmi/root/
mynamespace/myclass">
<p:Data1>Test Message</p:Data1>
<p:id>1</p:id>
<cim:Location xmlns:cim="http://schemas.dmtf.org/
wsman/2005/06/base"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<a:Address>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous</a:Address>
<a:ReferenceParameters>
<w:ResourceURI>
http://schemas.microsoft.com/wsman/2005/06/
wmi/root/mynamespace/myclass</w:ResourceURI>
<w:SelectorSet>
<w:Selector Name="id">1</w:Selector>
</w:SelectorSet>
</a:ReferenceParameters>
</cim:Location>
</p:myclass>
</n:Items>
</n:PullResponse>
</s:Body>
</s:Envelope>

```

4.1.2.5 Second Pull Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/
```

```

soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/
enumeration"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:To>http://server:80/wsman</a:To>
<w:ResourceURI s:mustUnderstand="true">
http://schemas.microsoft.com/
wsman/2005/06/wmi/root/mynamespace/myclass
</w:ResourceURI>
<a:ReplyTo>
<a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/08/addressing/role/anonymous</a:Address>
</a:ReplyTo>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/09/enumeration/Pull</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200
</w:MaxEnvelopeSize>
<a:MessageID>uuid:2C2D261E-D2C3-4A5D-80DE-
BB1A48E90BD2</a:MessageID>
<w:OperationTimeout>PT60.000S</w:OperationTimeout>
</s:Header>
<s:Body>
<n:Pull>
<n:EnumerationContext
xmlns:n="http://schemas.xmlsoap.org/ws/2004/
09/enumeration">
uuid:2504CA0D-94B9-4F91-B2F7-9F4CD9A2A96C
</n:EnumerationContext>
<n:MaxElements>1</n:MaxElements>
</n:Pull>
</s:Body>
</s:Envelope>

```

4.1.2.6 Second Pull Response with EndOfSequence

```

<s:Envelope xml:lang="en-US"
xmlns:s="http://www.w3.org/2003/05/
soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/
enumeration"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/09/enumeration/PullResponse</a:Action>
<a:MessageID s:mustUnderstand="true">

```

```

uuid:8820F22A-DB9C-448F-9297-C84519E93753
</a:MessageID>
<a:To>http://schemas.xmlsoap.org/ws/2004/08/
addressing/
role/anonymous</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:2C2D261E-D2C3-4A5D-80DE-BB1A48E90BD2</a:RelatesTo>
</s:Header>
<s:Body>
<n:PullResponse>
<n:Items>
<p:myclass xmlns:p="http://schemas.microsoft.com/
wsman/2005/06/wmi/root/mynamespace/myclass">
<p>Data1>"Hello again"</p>Data1>
<p:id>2</p:id>
<cim:Location xmlns:cim=
"http://schemas.dmtf.org/wsman/2005/06/base"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<a:Address>http://schemas.xmlsoap.org/ws/2004/08/
addressing/
role/anonymous</a:Address>
<a:ReferenceParameters>
<w:ResourceURI>http://schemas.microsoft.com/
wsman/2005/06/wmi/root/mynamespace/myclass
</w:ResourceURI>
<w:SelectorSet>
<w:Selector Name="id">2</w:Selector>
</w:SelectorSet>
</a:ReferenceParameters>
</cim:Location>
</p:myclass>
</n:Items>
<n:EndOfSequence/>
</n:PullResponse>
</s:Body>
</s:Envelope>

```

4.1.3 Modifying an Instance

To modify an instance, a Put request is used. Using the example from section [4.1.1](#), the below exchange shows the Data1 property being modified from "Hello World" to "Test String" in an instance of MyClass:

Put Request:

```

<s:Envelope xmlns:s=
"http://www.w3.org/2003/05/soap-envelope"
xmlns:a=
"http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:w=
"http://schemas.xmlsoap.org/ws/2005/06/management">
<s:Header>

```

```

<a:To>http://server:80/wsman</a:To>
<w:ResourceURI s:mustUnderstand="true">
http://schemas.microsoft.com/
wsman/2005/06/wmi/root/mynamespace/myclass
</w:ResourceURI>
<a:ReplyTo>
<a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/08/addressing/role/anonymous</a:Address>
</a:ReplyTo>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/09/transfer/Put
</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200
</w:MaxEnvelopeSize>
<a:MessageID>
uuid:D1408048-E0F6-4C6D-8B8A-515B9F7B641C
</a:MessageID>
<w:SelectorSet>
<w:Selector Name="id">1</w:Selector>
</w:SelectorSet>
<w:OperationTimeout>PT60.000S</w:OperationTimeout>
</s:Header>
<s:Body>
<p:myclass xmlns:p=
"http://schemas.microsoft.com/wsman/2005/06/wmi/
root/mynamespace/myclass">
<p:Data1>Test Message</p:Data1>
<p:id>1</p:id>
<cim:Location xmlns:cim=
"http://schemas.dmtf.org/wsman/2005/06/
base" xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<a:Address>http://schemas.xmlsoap.org/ws/2004/08/
addressing/
role/anonymous</a:Address>
<a:ReferenceParameters>
<w:ResourceURI>http://schemas.microsoft.com/
wsman/2005/06/wmi/
root/mynamespace/myclass</w:ResourceURI>
<w:SelectorSet>
<w:Selector Name="id">1</w:Selector>
</w:SelectorSet>
</a:ReferenceParameters>
</cim:Location>
</p:myclass>
</s:Body>
</s:Envelope>

```

Put Response:

```

<s:Envelope xml:lang="en-US" xmlns:

```



```

s="http://www.w3.org/2003/05/
soap-envelope" xmlns:a=
"http://schemas.xmlsoap.org/ws/2004/08/
addressing" xmlns:w=
"http://schemas.xmlsoap.org/ws/2005/06/management">
<s:Header>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/
ws/2004/09/transfer/PutResponse</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:92E94D15-B9D2-4DFB-AACF-9952F19B4AFB
</a:MessageID>
<a:To>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:D1408048-E0F6-4C6D-8B8A-515B9F7B641C</a:RelatesTo>
</s:Header>
<s:Body>
<p:myclass xmlns:p="http://schemas.microsoft.com/
wsman/2005/06/wmi/root/mynamespace/myclass">
<p>Data1>Test Message</p>Data1>
<p:id>1</p:id>
<cim:Location xmlns:cim="http://schemas.dmtf.org/
wsman/2005/06/base" xmlns:a=
"http://schemas.xmlsoap.org/ws/2004/
08/addressing" xmlns:w=
"http://schemas.xmlsoap.org/ws/2005/06/management">
<a:Address>
http://schemas.xmlsoap.org/ws/2004/08/addressing/
role/anonymous</a:Address>
<a:ReferenceParameters>
<w:ResourceURI>
http://schemas.microsoft.com/wsman/2005/06/wmi/
root/mynamespace/myclass</w:ResourceURI>
<w:SelectorSet>
<w:Selector Name="id">1</w:Selector>
</w:SelectorSet>
</a:ReferenceParameters>
</cim:Location>
</p:myclass>
</s:Body>
</s:Envelope>

```

4.1.4 Invoking a Method

The WS-Management Protocol can be used to invoke a method on a CIM class or instance. Win32_Process is a CIM class derived from CIM_Process, which has a method called Create that is used to create a process.

Invoke method request:

```

<s:Envelope xmlns:s=
"http://www.w3.org/2003/05/soap-envelope"
xmlns:a=
"http://schemas.xmlsoap.org/ws/2004/08/addressing"

```

```

xmlns:w=
"http://schemas.xmlsoap.org/ws/2005/06/management">
<s:Header>
<a:To>http://server:80/wsman</a:To>
<w:ResourceURI s:mustUnderstand="true">
http://schemas.microsoft.com/
wsman/2005/06/wmi/root/cimv2/Win32_Process
</w:ResourceURI>
<a:ReplyTo>
<a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004
/08/addressing/role/anonymous</a:Address>
</a:ReplyTo>
<a:Action s:mustUnderstand="true">
http://schemas.microsoft.com/wsman/
2005/06/wmi/root/cimv2/Win32_Process/Create
</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200
</w:MaxEnvelopeSize>
<a:MessageID>uuid:9A989269-283B-4624-BAC5-
BC291F72E854</a:MessageID>
<w:OperationTimeout>PT60.000S</w:OperationTimeout>
</s:Header>
<s:Body>
<p>Create_INPUT xmlns:p=
"http://schemas.microsoft.com/wsman/2005/06/
wmi/root/cim
/Win32_Process">
<p:CommandLine>notepad.exe</p:CommandLine>
<p:CurrentDirectory>C:\</p:CurrentDirectory>
</p>Create_INPUT>
</s:Body>
</s:Envelope>

```

Invoke method response:

```

<s:Envelope xml:lang="en-US" xmlns:s=
"http://www.w3.org/2003/05/
soap-envelope" xmlns:a=
"http://schemas.xmlsoap.org/ws/2004/08/
addressing" xmlns:w=
"http://schemas.xmlsoap.org/ws/2005/06/management">
<s:Header>
<a:Action s:mustUnderstand="true">
http://schemas.microsoft.com/
wsman/2005/06/wmi/root/cimv2/Win32_Process/
CreateResponse</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:F0228E67-F37B-4BE3-BAA2-3BB58AA6F911
</a:MessageID>
<a:To>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:9A989269-283B-4624-BAC5-BC291F72E854</a:RelatesTo>

```

```

</s:Header>
<s:Body>
<p>Create_OUTPUT xmlns:p=
"http://schemas.microsoft.com/wsman/2005/06/wmi/root/
cimv2/Win32_Process">
<p:ProcessId>4000</p:ProcessId>
<p:ReturnValue>0</p:ReturnValue>
</p>Create_OUTPUT>
</s:Body>
</s:Envelope>

```

4.2 Configuration Examples

This section illustrates protocol examples related to configuration of a Web Services Management Protocol Extensions for Windows Server 2003 service.

4.2.1 Retrieving Configuration

This section illustrates an example of the entire configuration of Web Services Management Protocol Extensions for Windows Server 2003 accessed using Get.

Get Request:

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/
soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:To>http://localhost:80/wsman</a:To>
<w:ResourceURI s:mustUnderstand="true">
wsman:microsoft.com/wsman/2005/06/config</w:ResourceURI>
<a:ReplyTo>
<a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/
08/addressing/role/anonymous</a:Address>
</a:ReplyTo>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200
</w:MaxEnvelopeSize>
<a:MessageID>uuid:613DCD71-95AF-4ED5-
86E2-1D6AB44ECE66</a:MessageID>
<w:OperationTimeout>
PT60.000S
</w:OperationTimeout>
</s:Header>
<s:Body/>
</s:Envelope>

```

Get Response:

```
<s:Envelope xml:lang="en-US" xmlns:s=
"http://www.w3.org/2003/05/
soap-envelope" xmlns:a=
"http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:w=
"http://schemas.xmlsoap.org/ws/2005/06/management">
<s:Header>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/
09/transfer/GetResponse</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:26ED5937-8016-41D5-9157-C9AD5B1D3C37</a:MessageID>
<a:To>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:613DCD71-95AF-4ED5-86E2-1D6AB44ECE66</a:RelatesTo>
</s:Header>
<s:Body>
<cfg:Config xmlns:cfg="
wsman:microsoft.com/wsman/2005/06/config.xsd">
<cfg:MaxEnvelopeSizekb>50</cfg:MaxEnvelopeSizekb>
<cfg:MaxTimeoutms>60000</cfg:MaxTimeoutms>
<cfg:MaxBatchItems>20</cfg:MaxBatchItems>
<cfg:SoapTraceEnabled>true</cfg:SoapTraceEnabled>
<cfg:MaxProviderRequests>25
</cfg:MaxProviderRequests>
<cfg:Client>
<cfg:NetworkDelays>5000</cfg:NetworkDelays>
<cfg:URLPrefix>wsman</cfg:URLPrefix>
<cfg:HTTP>
<cfg:Port>80</cfg:Port>
<cfg:Unencrypted>
<cfg:Basic>true</cfg:Basic>
<cfg:Digest>false</cfg:Digest>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:Unencrypted>
</cfg:HTTP>
<cfg:HTTPS>
<cfg:Port>443</cfg:Port>
<cfg:Basic>true</cfg:Basic>
<cfg:Digest>true</cfg:Digest>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:HTTPS>
</cfg:Client>
<cfg:Service>
<cfg:RootSDDL>O:NSG:BAD:P(A;;GA;;;BA)S:P
(AU;FA;GA;;;WD)(AU;SA;GWGX;;;WD)</cfg:RootSDDL>
<cfg:MaxConcurrentOperations>100
</cfg:MaxConcurrentOperations>
<cfg:EnumerationTimeoutms>60000
</cfg:EnumerationTimeoutms>
<cfg:MaxClientCertInfoSize>16384
</cfg:MaxClientCertInfoSize>
<cfg:MaxConnections>5</cfg:MaxConnections>
<cfg:HTTP>
<cfg:Unencrypted>
```

```

<cfg:Basic>true</cfg:Basic>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:Unencrypted>
</cfg:HTTP>
<cfg:HTTPS>
<cfg:Basic>true</cfg:Basic>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:HTTPS>
</cfg:Service>
</cfg:Config>
</s:Body>
</s:Envelope>

```

4.2.2 Modify Configuration

To modify a configuration, a Put request is used. In this example, `cfg:MaxBatchItems` is changed from 20 to 10.

Put Request:

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/
soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>
<a:To>http://localhost:80/wsman</a:To>
<w:ResourceURI s:mustUnderstand="true">
wsman:microsoft.com/wsman/2005/06/Config
</w:ResourceURI>
<a:ReplyTo>
<a:Address s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/08/addressing/role/anonymous</a:Address>
</a:ReplyTo>
<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/09/transfer/Put</a:Action>
<w:MaxEnvelopeSize s:mustUnderstand="true">
51200
</w:MaxEnvelopeSize>
<a:MessageID>
uuid:47F4F498-0050-4DCF-BCA1-5611732CF7DE
</a:MessageID>
<w:OperationTimeout>PT60.000S
</w:OperationTimeout>
</s:Header>
<s:Body>
<cfg:Config xmlns:cfg=
"wsman:microsoft.com/wsman/2005/06/config.xsd">
<cfg:MaxEnvelopeSizekb>50</cfg:MaxEnvelopeSizekb>
<cfg:MaxTimeoutms>60000</cfg:MaxTimeoutms>
<cfg:MaxBatchItems>10</cfg:MaxBatchItems>
<cfg:SoapTraceEnabled>true</cfg:SoapTraceEnabled>

```

```

<cfg:MaxProviderRequests>25
</cfg:MaxProviderRequests>
<cfg:Client>
<cfg:NetworkDelays>5000</cfg:NetworkDelays>
<cfg:URLPrefix>wsman</cfg:URLPrefix>
<cfg:HTTP>
<cfg:Port>80</cfg:Port>
<cfg:Unencrypted>
<cfg:Basic>true</cfg:Basic>
<cfg:Digest>false</cfg:Digest>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:Unencrypted>
</cfg:HTTP>
<cfg:HTTPS>
<cfg:Port>443</cfg:Port>
<cfg:Basic>true</cfg:Basic>
<cfg:Digest>true</cfg:Digest>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:HTTPS>
</cfg:Client>
<cfg:Service>
<cfg:RootSDDL>O:NSG:BAD:P(A;;GA;;;BA)
S:P(AU;FA;GA;;;WD)(AU;SA;GWGX;;;WD)</cfg:RootSDDL>
<cfg:MaxConcurrentOperations>100
</cfg:MaxConcurrentOperations>
<cfg:EnumerationTimeouts>60000
</cfg:EnumerationTimeouts>
<cfg:MaxClientCertInfoSize>16384
</cfg:MaxClientCertInfoSize>
<cfg:MaxConnections>5
</cfg:MaxConnections>
<cfg:HTTP>
<cfg:Unencrypted>
<cfg:Basic>true</cfg:Basic>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:Unencrypted>
</cfg:HTTP>
<cfg:HTTPS>
<cfg:Basic>true</cfg:Basic>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:HTTPS>
</cfg:Service>
</cfg:Config>
</s:Body>
</s:Envelope>

```

Put Response:

```

<s:Envelope xml:lang="en-US" xmlns:s=
"http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
<s:Header>

```

```

<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/
2004/09/transfer/PutResponse</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:791085DF-7297-4DA4-B10C-E7AB1EE2C821
</a:MessageID>
<a:To>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:47F4F498-0050-4DCF-BCA1-5611732CF7DE</a:RelatesTo>
</s:Header>
<s:Body>
<cfg:Config xmlns:cfg=
"wsman:microsoft.com/wsman/2005/06/config.xsd">
<cfg:MaxEnvelopeSizekb>50</cfg:MaxEnvelopeSizekb>
<cfg:MaxTimeoutms>60000</cfg:MaxTimeoutms>
<cfg:MaxBatchItems>10</cfg:MaxBatchItems>
<cfg:SoapTraceEnabled>true</cfg:SoapTraceEnabled>
<cfg:MaxProviderRequests>25
</cfg:MaxProviderRequests>
<cfg:Client>
<cfg:NetworkDelays>5000</cfg:NetworkDelays>
<cfg:URLPrefix>wsman</cfg:URLPrefix>
<cfg:HTTP>
<cfg:Port>80</cfg:Port>
<cfg:Unencrypted>
<cfg:Basic>true</cfg:Basic>
<cfg:Digest>false</cfg:Digest>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:Unencrypted>
</cfg:HTTP>
<cfg:HTTPS>
<cfg:Port>443</cfg:Port>
<cfg:Basic>true</cfg:Basic>
<cfg:Digest>true</cfg:Digest>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:HTTPS>
</cfg:Client>
<cfg:Service>
<cfg:RootSDDL>O:NSG:BAD:P(A;;GA;;;BA)S:P
(AU;FA;GA;;;WD)
(AU;SA;GWGX;;;WD)</cfg:RootSDDL>
<cfg:MaxConcurrentOperations>100
</cfg:MaxConcurrentOperations>
<cfg:EnumerationTimeoutms>60000
</cfg:EnumerationTimeoutms>
<cfg:MaxClientCertInfoSize>16384
</cfg:MaxClientCertInfoSize>
<cfg:MaxConnections>5</cfg:MaxConnections>
<cfg:HTTP>
<cfg:Unencrypted>
<cfg:Basic>true</cfg:Basic>
<cfg:Negotiate>true</cfg:Negotiate>
</cfg:Unencrypted>
</cfg:HTTP>
<cfg:HTTPS>
<cfg:Basic>true</cfg:Basic>
<cfg:Negotiate>true</cfg:Negotiate>

```

```

</cfg:HTTPS>
</cfg:Service>
</cfg:Config>
</s:Body>
</s:Envelope>

```

4.3 Fault Detail

In this section, an example of fault detail is shown. A Get request with an invalid Resource URI is sent, resulting in a fault.

Get Request:

```

<s:Envelope xmlns:s="http://www.w3.org/2003/05/
soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
  xmlns:f="http://schemas.xmlsoap.org/ws/2005/06/
wsmanfault">

  <s:Header>
    <a:To>http://localhost:80/wsman</a:To>
    <w:ResourceURI s:mustUnderstand="true">
      http://schemas.microsoft.com/wsman/2005/06/wmi/root/
      cimv2/win32_servic</w:ResourceURI>
    <a:ReplyTo>
      <a:Address s:mustUnderstand="true">
        http://schemas.xmlsoap.org/ws/2004/08/addressing/role/
        anonymous</a:Address>
      </a:ReplyTo>
      <a:Action s:mustUnderstand="true">
        http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
      </a:Action>
      <w:MaxEnvelopeSize s:mustUnderstand="true">
        51200
      </w:MaxEnvelopeSize>
      <a:MessageID>uuid:B2C3F241-1C90-4B91-9D66-
      EEA0DEB81879</a:MessageID>
      <w:OperationTimeout>PT60.000S</w:OperationTimeout>
    </s:Header>
    <s:Body/>
  </s:Envelope>

```

Fault Response:

```

<s:Envelope xml:lang="en-US"
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/
addressing"
xmlns:w="http://schemas.xmlsoap.org/ws/2005/06/
management">
  <s:Header>

```



```

<a:Action s:mustUnderstand="true">
http://schemas.xmlsoap.org/ws/2004/
08/addressing/fault</a:Action>
<a:MessageID s:mustUnderstand="true">
uuid:F6968902-D4EA-4B50-9F6E-DECCFBA1BDFD
</a:MessageID>
<a:To>http://schemas.xmlsoap.org/ws/2004/08/
addressing/role/anonymous
</a:To>
<a:RelatesTo s:mustUnderstand="true">
uuid:B2C3F241-1C90-4B91-9D66-EEA0DEB81879
</a:RelatesTo>
</s:Header>
<s:Body>
<s:Fault>
<s:Code>
<s:Value>s:Sender</s:Value>
<s:Subcode>
<s:Value>a:DestinationUnreachable</s:Value>
</s:Subcode>
</s:Code>
<s:Reason>
<s:Text xml:lang="en-US">The WS-Management service
cannot process the request. The service cannot find
the resource identified by the resource URI and
selectors. </s:Text>
</s:Reason>
<s:Detail>
<w:FaultDetail>wsman:faultDetail/ResourceNotFound
</w:FaultDetail>
<f:WSManFault xmlns:f="
http://schemas.microsoft.com/ws/2005/06/
wsmanfault" Code="32768"
Machine="http://localhost:80/wsman">
<f:Message>
<f:ProviderFault
providerId="D9A2A039-A4B3-4A70-8CB9-8D7714EAD776">
<f:WSManFault xmlns:f=
"http://schemas.microsoft.com/ws/2005/06/
wsmanfault" Code="32768" Machine="server">
<f:Message>The WS-Management service cannot
process the request. The service cannot find the
resource identified by the resource URI and selectors.
</f:Message>
</f:WSManFault>
</f:ProviderFault>
</f:Message>
</f:WSManFault>
</s:Detail>
</s:Fault>
</s:Body>
</s:Envelope>

```

5 Security

The following sections specify security considerations for implementers of the Web Services Management Protocol Extensions for Windows Server 2003.

5.1 Security Considerations for Implementers

Web Services Management Protocol Extensions for Windows Server 2003 uses the WS-Management Security Profiles as described in [\[DMTF-DSP0226\]](#) section 12.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST authenticate the request using one of the configured security profiles. See sections [2.1.3.3](#) and [3.1.5.5.3](#) for more details on configured profiles.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST authorize the request using the RootSDDL configuration setting defined in section [2.1.3.3](#).

Web Services Management Protocol Extensions for Windows Server 2003 clients MUST implement the following security profiles: [<27>](#)

- wsman:secprofile/http/basic described in [\[DMTF-DSP0226\]](#) section 12.5.
- wsman:secprofile/https/basic described in [\[DMTF-DSP0226\]](#) section 12.7.
- wsman:secprofile/https/spnego-kerberos described in [\[DMTF-DSP0226\]](#) section 12.12.
- wsman:secprofile/http/spnego-kerberos described in [\[DMTF-DSP0226\]](#) section 12.14.
- wsman:secprofile/http/digest as defined in [\[DMTF-DSP0226\]](#) section 12.6.
- wsman:secprofile/https/digest as defined in [\[DMTF-DSP0226\]](#) section 12.8.

Web Services Management Protocol Extensions for Windows Server 2003 servers MUST implement the following security profiles:

- wsman:secprofile/http/basic described in [\[DMTF-DSP0226\]](#) section 12.5.
- wsman:secprofile/https/basic described in [\[DMTF-DSP0226\]](#) section 12.7.
- wsman:secprofile/https/spnego-kerberos described in [\[DMTF-DSP0226\]](#) section 12.12.
- wsman:secprofile/http/spnego-kerberos described in [\[DMTF-DSP0226\]](#) section 12.14.

5.2 Index of Security Parameters

None.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Server 2003 R2

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

[<1> Section 1.3:](#) Windows Server 2003 R2 clients and servers implement a subset of the WS-Management Protocol as specified in [\[DMTF-DSP0226\]](#), WS-Management CIM Binding as specified in [\[DMTF-DSP0227-NM\]](#), and WS-CIM Mapping as specified in [\[DMTF-DSP0230-NM\]](#); specifications defined not by the published specifications, but by pre-release drafts of those specifications. Unlike the Windows Server 2003 R2 implementation, the Windows Vista and Windows Server 2008 implementations conform to the published specifications of [\[DMTF-DSP0226\]](#), [\[DMTF-DSP0227-NM\]](#), and [\[DMTF-DSP0230-NM\]](#).

[<2> Section 1.3:](#) Windows Server 2003 R2 servers support retrieval and update of existing CIM instances but do not support creation of new CIM instances or deletion of existing CIM instances.

[<3> Section 1.3:](#) Windows Server 2003 R2 servers support two resource providers:

- The CIM resource provider, which handles CIM-related requests. The GUID of the CIM resource provider is D9A2A039-A4B3-4A70-8CB9-8D7714EAD776.
- The Configuration resource provider, which handles configuration related requests. The GUID of the Configuration resource provider is FCBE098D-64C7-4b07-BB5B-748DBEC256A3.

[<4> Section 1.7:](#) Windows Server 2003 R2 servers do not support Identity requests. Instead, they provide the ability to retrieve the version of the protocol using a Get request on a specific Resource URI, which is described in section [3.1.5.4.4](#) of this specification.

[<5> Section 2.1.2.1:](#) Windows Server 2003 R2 servers do not support negative time stamps, which are cim:cimDateTime values that start with a dash ('-').

[<6> Section 2.1.3.2.2:](#) Web Services Management Protocol Extensions for Windows Server 2003 client and server must support the security profiles mentioned in section [5](#). When using any of these profiles, the client can pre-authenticate to the Web Services Management Protocol Extensions for Windows Server 2003 server. In this case, after the connection between the client and server is established, the client will send the Authentication header to the server. If the server supports the authentication scheme, then the server can respond with a 200 to indicate success

[<7> Section 2.1.3.2.4:](#) SSL mutual authentication is also supported in addition to the https security profiles. In this case, client-side certificates are used.

[<8> Section 2.1.3.4.1:](#) Windows Server 2003 R2 servers use the following string for the Vendor element: Microsoft Corporation.

[<9> Section 2.1.3.4.1:](#) Windows Server 2003 R2 servers use the following format to describe the version: OS: d.d.d SP: d.d Stack: d.d Where d is a 32 digit unsigned integer. For example, OS: 5.2.3790 SP: 1.0 Stack: 1.0.

- **OS:** The major and minor version numbers of the operating system.
- **SP:** The service pack installed on the machine.

- **Stack:** A version number that identifies which version of the WS-Management stack implementation is running.

<10> [Section 3.1.5.1.6:](#) Windows Server 2003 R2 servers do not support subscription operations and return a wsa:ActionNotSupported fault.

<11> [Section 3.1.5.1.9:](#) Windows Server 2003 R2 servers do not support non-zero year and non-zero month values in the OperationTimeout field.

<12> [Section 3.1.5.1.10:](#) Web Services Management Protocol Extensions for Windows Server 2003 servers use a default value for the MaxEnvelopeSize value to the value of the MaxEnvelopeSizekb configuration setting multiplied by 1024 if no MaxEnvelopeSize value is specified by the client or if the MaxEnvelopeSize value is more than this setting.

<13> [Section 3.1.5.1.12:](#) Windows Server 2003 R2 servers ignore the OptionSet element unless mustUnderstand="true", in which case they will return an s:NotUnderstood fault as required by [\[SOAP1.2/1\]](#).

<14> [Section 3.1.5.1.13:](#) Windows Server 2003 R2 servers ignore the RequestEPR header unless mustUnderstand="true", in which case they will return an s:NotUnderstood fault as required by [\[SOAP1.2/1\]](#).

<15> [Section 3.1.5.2.1:](#) Windows Server 2003 R2 clients can be configured to send Enumerate requests with the Filter element. Windows Server 2003 R2 servers do not support any filtering dialects.

<16> [Section 3.1.5.2.3:](#) Windows Server 2003 R2 servers ignore the RequestTotalItemsCountEstimate header.

<17> [Section 3.1.5.2.4:](#) Windows Server 2003 R2 servers ignore the OptimizeEnumeration element.

<18> [Section 3.1.5.2.5:](#) Windows Server 2003 R2 servers ignore the EnumerationMode element and support enumeration of objects only.

<19> [Section 3.1.5.2.6:](#) Windows Server 2003 R2 servers ignore the [wsman:Filter](#) element.

<20> [Section 3.1.5.3.1:](#) Windows Server 2003 R2 servers ignore the FragmentTransfer header unless mustUnderstand="true", in which case they will return an s:NotUnderstood fault as required by [\[SOAP1.2/1\]](#).

<21> [Section 3.1.5.4.4:](#) Windows Server 2003 R2 servers use the following string for the Vendor element: Microsoft Corporation.

<22> [Section 3.1.5.4.4:](#) Windows Server 2003 R2 servers use the following format to describe the version: OS: d.d.d SP: d.d Stack: d.d Where d is a 32-digit unsigned integer. For example, OS: 5.2.3790 SP: 1.0 Stack: 1.0

- OS: The major and minor version numbers of the operating system
- SP: The service pack installed on the machine
- Stack: A version number that identifies which version of the WS-Management stack implementation is running.

<23> [Section 3.1.5.5.3.1:](#) Windows Server 2003 R2 servers only accept user names to local accounts when using Basic.

[<24> Section 3.1.5.5.3.2:](#) Windows Server 2003 R2 servers only accept user names to local accounts when using Basic.

[<25> Section 3.1.5.5.3.3:](#) Windows Server 2003 R2 servers only accept user names to local accounts when using Basic.

[<26> Section 3.2.5.4:](#) Windows Server 2003 R2 clients remove the [cim:Location](#) element when requesting a Put operation following a Get operation on an object.

[<27> Section 5.1:](#) Windows Server 2003 R2 clients remove the cim:Location element when requesting a Put operation following a Get operation on an object.

7 Index

A

Abstract data model
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)
[Action](#)
[Applicability statement](#)
[Arrays](#)
[Attachments - binary](#)

B

[Binary attachments](#)

C

[Capability negotiation](#)
CIM binding
 [abstract data model](#)
 [higher-layer triggered events](#)
 [initialization](#)
 [local events](#)
 [message processing](#)
 [overview](#)
 [timer events](#)
 [timers](#)
[CIM data types](#)
[CIM examples](#)
[CIM namespace](#)
Client
 configuration data types ([section 2.1.3.1](#), [section 2.1.3.2](#))
[ClientHTTPSType](#)
[ClientHTTPType](#)
[ClientType](#)
[ClientUnencryptedType](#)
[Concurrent operations](#)
[ConfigType](#)
[Configuration](#)
[Configuration data types](#)
[Configuration examples](#)

D

Data model - abstract
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)
[Data types](#)
 [configuration](#)
 [miscellaneous](#)
[DateTime](#)
[Discovery](#)

E

[Embedded objects](#)
[Enumeration](#)

[Enumeration Garbage Collection timer](#)
[EnumerationMode](#)

Examples
 [CIM](#)
 [configuration](#)
 [fault detail](#)
 [overview](#)

F

Fault detail ([section 2.1.1.1](#), [section 3.1.5.4.3](#))
[Fault detail example](#)
[FaultTo](#)
[Fields - vendor-extensible](#)
Filter ([section 3.1.5.2.1](#), [section 3.1.5.2.6](#))
[FragmentTransfer](#)

G

[Garbage collection timer](#)
[Glossary](#)

H

[Headers](#)
Higher-layer triggered events
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)

I

[Implementers - security considerations](#)
[Inbound message size](#)
[Informative references](#)
Initialization
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)
[Introduction](#)

L

[ListenerType](#)
Local events
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)
[Locale](#)
[Location](#)

M

[MaxEnvelopeSize](#)
[MaxItems](#)
Message processing
 [CIM binding](#)
 [WS-CIM mapping](#)

[WS-Management](#)
[MessageID](#)
Messages
 [CIM data types](#)
 [configuration data types](#)
 [data types](#)
 [overview](#)
 [size](#)
 [syntax](#)
Metadata
[Miscellaneous data types](#)

N

Namespace
 [CIM](#)
 [CIM XML](#)
 [XML](#)
[Normative references](#)

O

[OperationTimeout](#)
[OptimizeEnumeration](#)
[OptionSet](#)
[Overview \(synopsis\)](#)

P

[Parameters - security](#)
[PolymorphismMode](#)
[Preconditions](#)
[Prerequisites](#)

R

Reference
 [informative](#)
 [normative](#)
[References - overview](#)
[Relationship to other protocols](#)
[ReplyTo](#)
[RequestEPR](#)
[RequestTotalItemsCountEstimate](#)

S

[Security](#)
[SelectorSet](#)
Server
 configuration data types ([section 2.1.3.1](#), [section 2.1.3.3](#), [section 2.1.3.4.1](#))
 [ServiceHTTPSType](#)
 [ServiceHTTPSType](#)
 [ServiceType](#)
 [Standards assignments](#)
 [Syntax - message](#)

T

[ThisType](#)

Timer events
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)
Timers
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)
[Transfer get and put](#)
Triggered events - higher-layer
 [CIM binding](#)
 [WS-CIM mapping](#)
 [WS-Management](#)

V

[Vendor-extensible fields](#)
[Versioning](#)

W

[Windows behavior](#)
WS-CIM mapping
 [abstract data model](#)
 [higher-layer triggered events](#)
 [initialization](#)
 [local events](#)
 [message processing](#)
 [overview](#)
 [timer events](#)
 [timers](#)
[wsman:microsoft.com/wsman/2005/06/config](#)
[wsman:microsoft.com/wsman/2005/06/config/client](#)
[wsman:microsoft.com/wsman/2005/06/config/listener](#)
[wsman:microsoft.com/wsman/2005/06/config/service](#)
WS-Management
 [abstract data model](#)
 [configuration](#)
 [enumeration](#)
 [headers](#)
 [higher-layer triggered events](#)
 [initialization](#)
 [local events](#)
 [message processing](#)
 [overview](#)
 [timer events](#)
 [timers](#)

X

XML namespace ([section 3.1.5.1.1](#), [section 3.2.5.2](#))