BS ISO/IEC 17963:2013

BSI Standards Publication

Web Services for Management (WS-Management) Specification

bsi.

...making excellence a habit.™

Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO/IEC 17963:2013.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2013.
Published by BSI Standards Limited 2013

ISBN 978 0 580 77532 1

ICS 35.020

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 28 February 2013.

Amendments issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 17963:2013

BS ISO/IEC 17963:2013

INTERNATIONAL STANDARD

ISO/IEC 17963

First edition
2013-02-01

Web Services for Management (WS-Management) Specification

Spécification des services Web pour le management (WS-Management)

Reference number
ISO/IEC 17963:2013(E)

Documents
Document Sharing Hub

© ISO/IEC 2013

Full document details: BS ISO/IEC 17963:2013

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2013

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 17963:2013

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 17963 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 38, *Distributed application platforms and services (DAPS).*

Documents
Document Sharing Hub

© ISO/IEC 2013 – All rights reserved

Full document details: BS ISO/IEC 17963:2013

Documents
Document Sharing Hub

1

2 **Document Number: DSP0226**

3 **Date: 2012-08-28**

4 **Version: 1.1.1**

5 Web Services for Management (WS-
6 Management) Specification

7 **Document Type: Specification**

8 **Document Status: DMTF Standard**

9 **Document Language: en-US**

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

Copyright Notice

Copyright © 2006–2012 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

Documents
Document Sharing Hub

© ISO/IEC 2013 – All rights reserved
Full document details: BS ISO/IEC 17963:2013

CONTENTS

Foreword 7
1 Scope 10
2 Normative References 10
3 Terms and Definitions 12
4 Symbols and Abbreviated Terms 15
5 Addressing 16
5.1 Management Addressing 16
5.2 Versions of Addressing 25
5.3 Requirements for Compatibility 25
5.4 Use of Addressing in WS-Management 27
6 WS-Management Control Headers 44
6.1 wsman:OperationTimeout 44
6.2 wsman:MaxEnvelopeSize 45
6.3 wsman:Locale 46
6.4 wsman:OptionSet 47
6.5 wsman:RequestEPR 50
7 Resource Access 51
7.1 General 51
7.2 Addressing Uniformity 53
7.3 Get 54
7.4 Put 55
7.5 Delete 59
7.6 Create 61
7.7 Fragment-Level Access 64
7.8 Fragment-Level Get 66
7.9 Fragment-Level Put 67
7.10 Fragment-Level Delete 70
7.11 Fragment-Level Create 71
8 Enumeration of Datasets 73
8.1 General 73
8.2 Enumerate 75
8.3 Filter Interpretation 82
8.4 Pull 84
8.5 Release 88
8.6 Ad-Hoc Queries and Fragment-Level Enumerations 90
8.7 Enumeration of EPRs 90
8.8 Renew 92
8.9 GetStatus 94
8.10 EnumerationEnd 94
9 Custom Actions (Methods) 95
10 Notifications (Eventing) 96
10.1 General 96
10.2 Subscribe 97
10.3 GetStatus 117
10.4 Unsubscribe 118
10.5 Renew 119
10.6 SubscriptionEnd 120
10.7 Acknowledgement of Delivery 122
10.8 Refusal of Delivery 123
10.9 Dropped Events 124
10.10 Access Control 125

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

10.11 Implementation Considerations 126
10.12 Advertisement of Notifications 126
11 Metadata and Discovery 126
12 Security 129
12.1 General 129
12.2 Security Profiles 130
12.3 Security Considerations for Event Subscriptions 130
12.4 Including Credentials with a Subscription 131
12.5 Correlating Events with a Subscription 132
12.6 Transport-Level Authentication Failure 132
12.7 Security Implications of Third-Party Subscriptions 132
13 Transports and Message Encoding 133
13.1 SOAP 133
13.2 Lack of Response 134
13.3 Replay of Messages 134
13.4 Encoding Limits 134
13.5 Binary Attachments 135
13.6 Case-Sensitivity 135
14 Faults 136
14.1 Introduction 136
14.2 Fault Encoding 136
14.3 NotUnderstood Faults 137
14.4 Degenerate Faults 138
14.5 Fault Extensibility 138
14.6 Master Faults 139
ANNEX A (informative) Notational Conventions 160
A.1 XML Namespaces 160
ANNEX B (normative) Conformance 162
ANNEX C (normative) HTTP(S) Transport and Security Profile 163
C.1 General 163
C.2 HTTP(S) Binding 163
C.3 HTTP(S) Security Profiles 165
C.4 IPSec and HTTP 170
ANNEX D (informative) XPath Support 171
D.1 General 171
D.2 Level 1 172
D.3 Level 2 174
ANNEX E (normative) Selector Filter Dialect 177
ANNEX F (informative) Identify XML Schema 179
ANNEX G (informative) Resource Access Operations XML Schema and WSDL 182
ANNEX H (informative) Enumeration Operations XML Schema and WSDL 187
ANNEX I (informative) Notification OperationsXML Schema and WSDL 196
ANNEX J (informative) Addressing XML Schema 204
ANNEX K (informative) WS-Management XML Schema 207
ANNEX L (informative) Change Log 217

© ISO/IEC 2013 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

Figures

Figure 1 – Message Information Header Blocks 20

Tables

Table 1 – Relationship Type 21
Table 2 – Interoperability Requirements 25
Table 3 – WSA Versions in Exchanges 26
Table 4 – wsa:Action URI Descriptions 42
Table 5 – wsman:AccessDenied 139
Table 6 – wsa:ActionNotSupported 140
Table 7 – wsman:AlreadyExists 140
Table 8 – wsmen:CannotProcessFilter 141
Table 9 – wsman:CannotProcessFilter 141
Table 10 – wsman:Concurrency 142
Table 11 – wsme:DeliveryModeRequestedUnavailable 142
Table 12 – wsman:DeliveryRefused 143
Table 13 – wsa:DestinationUnreachable 143
Table 14 – wsman:EncodingLimit 144
Table 15 – wsa:EndpointUnavailable 145
Table 16 – wsman:EventDeliverToUnusable 145
Table 17 – wsme:EventSourceUnableToProcess 146
Table 18 – wsmen:FilterDialectRequestedUnavailable 146
Table 19 – wsme:FilteringNotSupported 146
Table 20 – wsmen:FilteringNotSupported 147
Table 21 – wsme:FilteringRequestedUnavailable 147
Table 22 – wsman:FragmentDialectNotSupported 148
Table 23 – wsman:InternalError 148
Table 24 – wsman:InvalidBookmark 149
Table 25 – wsmen:InvalidEnumerationContext 149
Table 26 – wsme:InvalidExpirationTime 150
Table 27 – wsmen:InvalidExpirationTime 150
Table 28 – wsme:InvalidMessage 151
Table 29 – wsa:InvalidMessageInformationHeader 151
Table 30 – wsman:InvalidOptions 152
Table 31 – wsman:InvalidParameter 152
Table 32 – wsmt:InvalidRepresentation 153
Table 33 – wsman:InvalidSelectors 153
Table 34 – wsa:MessageInformationHeaderRequired 154
Table 35 – wsman:NoAck 154
Table 36 – wsman:QuotaLimit 154
Table 37 – wsman:SchemaValidationError 155

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

172 Table 38 – wsmen:TimedOut 155
173 Table 39 – wsman:TimedOut 155
174 Table 40 – wsme:UnableToRenew 156
175 Table 41 – wsme:UnsupportedExpirationType 156
176 Table 42 – wsmen:UnsupportedExpirationType 156
177 Table 43 – wsman:UnsupportedFeature 157
178 Table 44 – wsme:UnsupportedExpirationType 158
179 Table 45 – wsmen:UnableToRenew 158
180 Table 46 – wsa:InvalidMessage 158
181 Table 47 – wsme:CannotProcessFilter 159
182 Table A-1 – Prefixes and XML Namespaces Used in This Specification 161
183 Table C-1 – Basic Authentication Sequence 165
184 Table C-2 – Digest Authentication Sequence 166
185 Table C-3 – Basic Authentication over HTTPS Sequence 166
186 Table C-4 – Digest Authentication over HTTPS Sequence 167
187 Table C-5 – HTTPS with Client Certificate Sequence 167
188 Table C-6 – Basic Authentication over HTTPS with Client Certificate Sequence 168
189 Table C-7 – SPNEGO Authentication over HTTPS Sequence 169
190 Table C-8 – SPNEGO Authentication over HTTPS with Client Certificate Sequence 169
191 Table D-1 – XPath Level 1 Terminals 173
192 Table D-2 – XPath Level 2 Terminals 175
193

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

Foreword

The *Web Services for Management (WS-Management) Specification* (DSP0226) was prepared by the WS-Management sub-group of the WBEM Infrastructure & Protocols Working Group.

This International Standard makes use of functionality similar to the following W3C Recommendations:

- Web Services Eventing (WS-Eventing)
- Web Services Transfer (WS-Transfer)
- Web Services Enumeration (WS-Enumeration)

These W3C Recommendations were not available at the time WS-Management was defined, and similar functionality was incorporated directly into provisions of the WS-Management specification. Future revisions of WS-Management might incorporate these functions by External Reference to these W3C Recommendations

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Acknowledgements

The authors wish to acknowledge the following people.

Chairpersons:

- Josh Cohen – Microsoft
- Larry Lamers (Vice-Chairman) – VMware

Editors:

- Nathan Burkhart – Microsoft
- Doug Davis – IBM
- Raymond McCollum – Microsoft
- Bryan Murray – HP
- Brian Reistad – Microsoft

Authors:

- Akhil Arora – Sun Microsystems
- Vince Brunssen – IBM
- Mark Carlson – Sun Microsystems
- Jim Davis – WBEM Solutions
- Tony Dicenzo – Oracle
- Mike Dutch – Symantec
- Zulah Eckert – BEA Systems
- George Ericson – EMC
- Wassim Fayed – Microsoft
- Chris Ferris – IBM
- Bob Freund – Hitachi Ltd.
- Eugene Golovinsky – BMC Software
- Yasuhiro Hagiwara – NEC

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

- Steve Hand – Olocity
- Jackson He – Intel
- David Hines – Intel
- Reiji Inohara – NEC
- Christane Kämpfe – Fujitsu-Siemens Computers
- Paul Knight – Nortel Networks
- Vincent Kowalski – BMC Software
- Heather Kreger – IBM
- Vishwa Kumbalimutt – Microsoft
- Sunil Kunisetty – Oracle
- Richard Landau – Dell
- Paul Lipton – CA
- James Martin – Intel
- Milan Milenkovic – Intel
- Jeff Mischkinsky – Oracle
- Paul Montgomery – AMD
- Jishnu Mukurji – HP
- Alexander Nosov – Microsoft
- Abhay Padlia – Novell
- Gilbert Pilz – Oracle
- Roger Reich – Symantec
- Larry Russon – Novell
- Tom Rutt – Fujitsu Ltd.
- Jeffrey Schlimmer – Microsoft
- Dr. Hemal Shah – Broadcom
- Sharon Smith – Intel
- Enoch Suen – Dell
- Vijay Tewari – Intel
- William Vambenepe – HP
- Andrea Westerinen – CA, Inc.
- Kirk Wilson – CA, Inc.
- Dr. Jerry Xie – Intel

Contributors:

- Paul C. Allen – Microsoft
- Rodrigo Bomfim – Microsoft
- Don Box – Microsoft
- Jerry Duke – Intel
- David Filani – Intel
- Kirill Gavrylyuk – Microsoft
- Omri Gazitt – Microsoft
- Frank Gorishek – AMD
- Lawson Guthrie – Intel
- Arvind Kumar – Intel
- Brad Lovering – Microsoft

© ISO/IEC 2013 – All rights reserved

Full document details: BS ISO/IEC 17963:2013

- Pat Maynard – Intel
- Steve Millet – Microsoft
- Matthew Senft – Microsoft
- Barry Shilmover – Microsoft
- Tom Slaight – Intel
- Marvin Theimer – Microsoft
- Dave Tobias – AMD
- John Tollefsrud – Sun
- Anders Vinberg – Microsoft
- Megan Wallent – Microsoft

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

Web Services for Management (WS-Management) Specification

1 Scope

The *Web Services for Management (WS-Management) Specification* describes a Web services protocol based on SOAP for use in management-specific domains. These domains include the management of entities such as PCs, servers, devices, Web services and other applications, and other manageable entities. Services can expose only a WS-Management interface or compose the WS-Management service interface with some of the many other Web service specifications.

A crucial application for these services is in the area of systems management. To promote interoperability between management applications and managed resources, this specification identifies a core set of Web service specifications and usage requirements that expose a common set of operations central to all systems management. This includes the ability to do the following:

- Get, put (update), create, and delete individual resource instances, such as settings and dynamic values
- Enumerate the contents of containers and collections, such as large tables and logs
- Subscribe to events emitted by managed resources
- Execute specific management methods with strongly typed input and output parameters

In each of these areas of scope, this specification defines minimal implementation requirements for conformant Web service implementations. An implementation is free to extend beyond this set of operations, and to choose not to support one or more of the preceding areas of functionality if that functionality is not appropriate to the target device or system.

This specification intends to meet the following requirements:

- Constrain Web services protocols and formats so that Web services can be implemented with a small footprint in both hardware and software management services.
- Define minimum requirements for compliance without constraining richer implementations.
- Ensure backward compatibility and interoperability with WS-Management version 1.0.
- Ensure composability with other Web services specifications.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IETF RFC 2616, R. Fielding et al, *Hypertext Transfer Protocol (HTTP 1.1)*, June 1999, http://www.ietf.org/rfc/rfc2616.txt

IETF RFC 2818, E. Rescorla, *HTTP over TLS (HTTPS)*, May 2000, http://www.ietf.org/rfc/rfc2818.txt

IETF, RFC 3986, T. Berners-Lee et al, *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998, http://www.ietf.org/rfc/rfc3986.txt

© ISO/IEC 2013 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

IETF, RFC 4122, P. Leach et al, *A Universally Unique Identifier (UUID) URN Namespace*, July 2005, http://www.ietf.org/rfc/rfc4122.txt

IETF RFC 4178, L. Zhu et al, *The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism*, October 2005, http://www.ietf.org/rfc/rfc4178.txt

IETF, RFC 4559, K. Jaganathan et al, *SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows*, June 2006, http://www.ietf.org/rfc/rfc4559.txt

IETF RFC 5646, A. Phillips et al, *Tags for Identifying Languages*, September 2009, http://tools.ietf.org/rfc/rfc5646.txt

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*, http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

OASIS, A. Nadalin et al, *Web Services Security Username Token Profile 1.0*, March 2004, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

OASIS, A. Nadalin et al, *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)*, March 2004, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

OASIS, S. Anderson et al, *Web Services Trust Language (WS-Trust)*, December 2005, http://schemas.xmlsoap.org/ws/2005/02/trust

The Unicode Consortium, *The Unicode Standard Version 3.0*, January 2000, http://www.unicode.org/book/u2.html

The Unicode Consortium, *Byte Order Mark (BOM) FAQ*, http://www.unicode.org/faq/utf_bom.html#BOM

W3C, M. Gudgin, et al, *SOAP Version 1.2 Part 1: Messaging Framework*, June 2003, http://www.w3.org/TR/soap12-part1/

W3C, M. Gudgin, et al, *SOAP Version 1.2 Part 2: Adjuncts*, June 2003, http://www.w3.org/TR/2003/REC-soap12-part2-20030624

W3C, M. Gudgin, et al, *SOAP Message Transmission Optimization Mechanism (MTOM)*, November 2004, http://www.w3.org/TR/2004/PR-soap12-mtom-20041116/

W3C, J. Clark et al, *XML Path Language Version 1.0 (XPath 1.0)*, November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116

W3C, J. Cowan et al, *XML Information Set Second Edition (XML Infoset)*, February 2004, http://www.w3.org/TR/2004/REC-xml-infoset-20040204/

W3C, H. Thompson et al, *XML Schema Part 1: Structures (XML Schema 1)*, May 2001, http://www.w3.org/TR/xmlschema-1/

W3C, P. Biron et al, *XML Schema Part 2: Datatypes (XML Schema 2)*, May 2001, http://www.w3.org/TR/xmlschema-2/

W3C, *Web Services Addressing 1.0 – Core*, W3C Recommendation, May 2006, http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/

W3C, *Web Services Addressing 1.0 – SOAP Binding*, W3C Recommendation, May 2006, http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/

W3C, *Web Services Addressing 1.0 – Metadata*, W3C Recommendation, September 2007, http://www.w3.org/TR/2007/REC-ws-addr-metadata-20070904/

W3C, *Extensible Markup Language (XML) 1.0*, W3C Recommendation, October 2000, http://www.w3.org/TR/2000/REC-xml-20001006

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

W3C, *Namespaces in XML*, W3C Recommendation, January 1999, http://www.w3.org/TR/1999/REC-xml-names-19990114/

W3C, E. Christensen et al, *Web Services Description Language Version 1.1 (WSDL/1.1)*, March 2001, http://www.w3.org/TR/wsdl

W3C, S. Boag et al, XQuery 1.0: An XML Query Language (XQuery 1.0), January 2007, http://www.w3.org/TR/2007/REC-xquery-20070123/

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. The fact that a normative term such as "shall", “shall not”, “should”, “should not”, “may”, or “need not” may be used in text which does not have an associated rule number does not mean that the text is not normative.

3.1
can
used for statements of possibility and capability, whether material, physical, or causal

3.2
cannot
used for statements of possibility and capability, whether material, physical, or causal

3.3
conditional
indicates requirements to be followed strictly to conform to the document when the specified conditions are met

3.4
mandatory
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.5
may
indicates a course of action permissible within the limits of the document

3.6
need not
indicates a course of action permissible within the limits of the document

3.7
optional
indicates a course of action permissible within the limits of the document

3.8
shall
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

3.9
shall not
indicates requirements to be followed strictly to conform to the document and from which no deviation is permitted

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

3.10
should
indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.11
should not
indicates that a certain possibility or course of action is deprecated but not prohibited

3.12
client
the application that uses the Web services defined in this document to access the management service

3.13
consumer
the Web service that is requesting the data enumeration from the data source

3.14
data source
a Web service that supports traversal using enumeration contexts via the Enumerate operation defined in this specification

3.15
delivery mode
the mechanism by which notification messages are delivered from the source to the sink

3.16
enumeration context
a session context that represents a specific traversal through a logical sequence of XML element information items using the Pull operation defined in this specification

3.17
event sink
a Web service that receives notifications

3.18
event source
a Web service that sends notifications and accepts requests to create subscriptions

3.19
managed resource
an entity that can be of interest to an administrator
It may be a physical object, such as a laptop computer or a printer, or an abstract entity, such as a service.

3.20
notification
a message sent to indicate that an event has occurred

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

3.21
push mode
a delivery mechanism where the source sends event messages to the sink as individual, unsolicited SOAP messages

3.22
resource
a Web service that is addressable by an endpoint reference and accessed using the operations defined in this specification. This resource can be represented by an XML document. The XML document may be a representation of managed resource

3.23
resource class
an abstract representation (type) of a managed resource
A resource class defines the representation of management-related operations and properties. An example of a resource class is the description of operations and properties for a set of laptop computers.

3.24
resource factory
a Web service that is capable of creating new resources using the Create operation defined in this specification

3.25
resource instance
an instantiation of a resource class
An example is the set of management-related operations and property values for a specific laptop computer.

3.26
selector
a resource-relative name and value pair that acts as an instance-level discriminant when used with the WS-Management default addressing model
A selector is essentially a filter or "key" that identifies the desired instance of the resource. A selector may not be present when service-specific addressing models are used.

The relationship of services to resource classes and instances is as follows:

- A service consists of one or more resource classes.
- A resource class may contain zero or more instances.

If more than one instance for a resource class exists, they are isolated or identified through parts of the SOAP address for the resource, such as the ResourceURI and SelectorSet fields in the default addressing model.

3.27
service
an application that provides management services to clients by exposing the Web services defined in this document
Typically, a service is equivalent to the network "listener," is associated with a physical transport address, and is essentially a type of manageability access point.

3.28
subscriber
a Web service that sends requests to create, renew, and/or delete subscriptions

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

3.29
subscription manager
a Web service that accepts requests to manage, get the status of, renew, and/or delete subscriptions on behalf of an event source

4 Symbols and Abbreviated Terms

The following symbols and abbreviations are used in this document.

4.1
BNF
Backus-Naur Form (http://foldoc.org/foldoc/?Backus-Naur+Form)

4.2
BOM
byte-order mark

4.3
CQL
CIM Query Language

4.4
EPR
Endpoint Reference

4.5
GSSAPI
Generic Security Services Application Program Interface

4.6
SOAP
Simple Object Access Protocol

4.7
SPNEGO
Simple and Protected GSSAPI Negotiation Mechanism

4.8
SQL
Structured Query Language

4.9
URI
Uniform Resource Identifier

4.10
URL
Uniform Resource Locator

4.11
UTF
UCS Transformation Format

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

4.12
UUID
Universally Unique Identifier

4.13
WSDL
Web Services Description Language

4.14
WS-Man
Web Services Management

5 Addressing

WS-Management relies on a SOAP-based addressing mechanism (like the one defined in 5.1) to define references to other Web service endpoints and to define some of the headers used in SOAP messages. This addressing mechanism is semantically equivalent and fully wire-compatible with the version of WS-Addressing referenced in WS-Management 1.0. Therefore, this change to WS-Management is fully backward compatible with existing WS-Management implementations.

Clause 5.2 specifies how more than one addressing version may be used with WS-Management, such as the version defined in 5.1 or the W3C Recommendation version of addressing. In this specification, unless explicitly referring to a particular version, the term "Addressing" refers generically to either version of addressing as defined in 5.2.

Multiple addressing models may be used with any of the addressing versions described in 5.2. Implementations may implement any of the following addressing models:

- basic addressing as defined in 5.1
- the Default Addressing Model as defined in 5.4.2
- new addressing models that are not defined in this specification. These addressing models may impose additional restrictions or requirements for addressing.

5.1 Management Addressing

The features defined in this clause provide a transport-neutral mechanism to address Web services and messages. Specifically, this clause defines XML elements to identify Web service endpoints and to secure end-to-end endpoint identification in messages. This enables messaging systems to support message transmission through networks that include processing nodes such as endpoint managers, firewalls, and gateways in a transport-neutral manner.

5.1.1 Introduction

This clause defines two interoperable constructs, endpoint references and message information headers, that convey information that is typically provided by transport protocols and messaging systems. These constructs normalize this underlying information into a uniform format that can be processed independently of transport or application.

A Web service endpoint is an entity, processor, or resource that can be referenced and can be targeted for Web service messages. Endpoint references convey the information needed to identify and reference a Web service endpoint, and they may be used in several different ways:

- Endpoint references are suitable for conveying the information needed to access a Web service endpoint.

© ISO/IEC 2013 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

- Endpoint references are also used to provide addresses for individual messages sent to and from Web services.

To deal with the latter use case, this clause defines a family of message information headers that allows uniform addressing of messages independent of underlying transport. These message information headers convey end-to-end message characteristics including addressing for source and destination endpoints as well as message identity.

EXAMPLE: The following example illustrates the use of these mechanisms in a SOAP 1.2 message being sent from http://business456.example/client1 to http://fabrikam123.example/Purchasing.

Lines (002) to (014) represent the header of the SOAP message where the mechanisms defined in this clause are used. The body is represented by lines (015) to (017).

Lines (003) to (013) contain the message information header blocks. Specifically, lines (003) to (005) specify the identifier for this message, lines (006) to (008) specify the endpoint from where the message originated, and lines (009) to (011) specify the endpoint to which replies to this message should be sent as an Endpoint Reference. Line (012) specifies the address URI of the ultimate receiver of this message. Line (013) specifies an Action URI identifying expected semantics.

```
(001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
                  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
(002)   <S:Header>
(003)    <wsa:MessageID>
(004)      uuid:6B29FC40-CA47-1067-B31D-00DD010662DA
(005)    </wsa:MessageID>
(006)    <wsa:From>
(007)      <wsa:Address>http://business456.example/client1</wsa:Address>
(008)    </wsa:From>
(009)    <wsa:ReplyTo>
(010)      <wsa:Address>http://business456.example/client1</wsa:Address>
(011)    </wsa:ReplyTo>
(012)    <wsa:To>http://fabrikam123.example/Purchasing</wsa:To>
(013)    <wsa:Action>http://fabrikam123.example/SubmitPO</wsa:Action>
(014)   </S:Header>
(015)   <S:Body>
(016)     ...
(017)   </S:Body>
(018) </S:Envelope>
```

5.1.2 Endpoint References

This clause defines the syntax of an Endpoint Reference (EPR).

5.1.2.1 Format of Endpoint References

This clause defines an XML representation for an endpoint reference as both an XML type (wsa:EndpointReferenceType) and as an XML element (<wsa:EndpointReference>).

The wsa:EndpointReferenceType type is used wherever a Web service endpoint is referenced. The following describes the contents of this type:

```
<wsa:EndpointReference>
    <wsa:Address>xs:anyURI</wsa:Address>
    <wsa:ReferenceProperties>... </wsa:ReferenceProperties> ?
    <wsa:ReferenceParameters>... </wsa:ReferenceParameters> ?
    <wsa:PortType>xs:QName</wsa:PortType> ?
    <wsa:ServiceName PortName="xs:NCName"?>xs:QName</wsa:ServiceName> ?
    <wsp:Policy> ... </wsp:Policy>*
</wsa:EndpointReference>
```


© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

The following describes the attributes and elements listed in the preceding schema overview:

wsa:EndpointReference

This represents some element of type wsa:EndpointReferenceType. This example uses the predefined <wsa:EndpointReference> element, but any element of type wsa:EndpointReferenceType may be used.

wsa:EndpointReference/wsa:Address

This required element (of type xs:anyURI) specifies the address URI that identifies the endpoint. This address may be a logical address or identifier for the service endpoint.

wsa:EndpointReference/wsa:ReferenceProperties/

This optional element contains any number of individual reference properties that are associated with the endpoint to facilitate a particular interaction. Reference properties are XML elements that are required to properly interact with the endpoint. Reference properties are provided by the issuer of the endpoint reference and are otherwise assumed to be opaque to consuming applications.

NOTE: The use of reference properties is deprecated; reference parameters should be used instead.

wsa:EndpointReference/wsa:ReferenceProperties/{any}

Each child element of ReferenceProperties represents an individual reference property.

wsa:EndpointReference/wsa:ReferenceParameters/

This optional element contains any number of individual parameters that are associated with the endpoint to facilitate a particular interaction. Reference parameters are XML elements that are required to properly interact with the endpoint. Reference parameters are also provided by the issuer of the endpoint reference and are otherwise assumed to be opaque to consuming applications.

See 5.4 for some WS-Management-specific reference parameters.

wsa:EndpointReference/wsa:ReferenceParameters/{any}

Each child element of ReferenceParameters represents an individual reference parameter.

wsa:EndpointReference/wsa:PortType

This optional element (of type xs:QName) specifies the value of the primary portType of the endpoint being conveyed.

NOTE: The use of wsa:PortType is deprecated.

wsa:EndpointReference/wsa:ServiceName

This optional element (of type xs:QName) specifies the <wsdl:service> definition that contains a WSDL description of the endpoint being referenced. The service name provides a link to a full description of the service endpoint. An optional non-qualified name identifies the specific port in the service that corresponds to the endpoint.

NOTE: The use of wsa:ServiceName is deprecated.

wsa:EndpointReference/wsa:ServiceName/@PortName

This optional attribute (of type xs:NCName) specifies the name of the <wsdl:port> definition that corresponds to the endpoint being referenced.

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

wsa:EndpointReference/wsp:Policy

This optional element specifies a policy that is relevant to the interaction with the endpoint.

NOTE: The use of wsp:Policy is deprecated.

wsa:EndpointReference/{any}

This is an extensibility mechanism to allow additional elements to be specified.

wsa:EndpointReference/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

EXAMPLE: The following example illustrates an endpoint reference. This element references the URI "http://www.fabrikam123.example/acct":

```
<wsa:EndpointReference xmlns:wsa="..." xmlns:fabrikam="...">
   <wsa:Address>http://www.fabrikam123.example/acct</wsa:Address>
</wsa:EndpointReference>
```

5.1.2.2 Binding Endpoint References

When a message needs to be addressed to the endpoint, the information contained in the endpoint reference is mapped to the message according to a transformation that is dependent on the protocol and data representation used to send the message. Protocol-specific mappings (or bindings) define how the information in the endpoint reference is copied to message and protocol fields. This clause defines the SOAP binding for endpoint references. This mapping may be explicitly replaced by other bindings (defined as WSDL bindings or as policies); however, in the absence of an applicable policy stating that a different mapping is to be used, the SOAP binding defined here is assumed to apply. To ensure interoperability with a broad range of devices, all conformant implementations shall support the SOAP binding.

The SOAP binding for endpoint references is defined by the following two rules:

R5.1.2.2-1: The wsa:Address element in the endpoint reference shall be copied in the wsa:To header field of the SOAP message.

R5.1.2.2-2: Each Reference Property and Reference Parameter element becomes a header block in the SOAP message. The elements of each Reference Property or Reference Parameter (including all of its child elements, attributes, and in-scope namespaces) shall be added as a header block in the new message.

EXAMPLE: The following example shows how the default SOAP binding for endpoint references is used to construct a message addressed to the endpoint:

```
<wsa:EndpointReference xmlns:wsa="..." xmlns:fabrikam="...">
   <wsa:Address>http://www.fabrikam123.example/acct</wsa:Address>
   <wsa:ReferenceParameters>
       <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
       <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
   </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

According to the mapping rules stated before, the address value is copied in the "To" header and the "CustomerKey" element should be copied literally as a header in a SOAP message addressed to this endpoint. The SOAP message would look as follows:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
         xmlns:wsa="..." xmlns:fabrikam="... ">
   <S:Header>
```


© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

```
      ...
    <wsa:To>http://www.fabrikam123.example/acct</wsa:To>
    <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
    <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
      ...
   </S:Header>
   <S:Body>
      ...
   </S:Body>
</S:Envelope>
```

5.1.3 Message Information Headers

This clause defines the syntax of a message information header.

The message information headers collectively augment a message with the headers shown in Figure 1. These headers enable the identification and location of the endpoints involved in an interaction. The basic interaction pattern from which all others are composed is "one way". In this pattern a source sends a message to a destination without any further definition of the interaction.

"Request Reply" is a common interaction pattern that consists of an initial message sent by a source endpoint (the request) and a subsequent message sent from the destination of the request back to the source (the reply). A reply can be an application message, a fault, or any other message.

The message information header blocks provide end-to-end characteristics of a message that can be easily secured as a unit. The information in these headers is immutable and not intended to be modified along the message path.

Figure 1 shows the contents of the message information header blocks:

```
   <wsa:MessageID> xs:anyURI </wsa:MessageID>
   <wsa:RelatesTo RelationshipType="..."?>xs:anyURI</wsa:RelatesTo>
   <wsa:To>xs:anyURI</wsa:To>
   <wsa:Action>xs:anyURI</wsa:Action>
   <wsa:From>endpoint-reference</wsa:From>
   <wsa:ReplyTo>endpoint-reference</wsa:ReplyTo>
   <wsa:FaultTo>endpoint-reference</wsa:FaultTo>
```

Figure 1 – Message Information Header Blocks

The following describes the attributes and elements listed in Figure 1:

wsa:MessageID

This optional element (of type xs:anyURI) uniquely identifies this message in time and space. This element shall be present if wsa:ReplyTo or wsa:FaultTo is present. No two messages with a distinct application intent may share a wsa:MessageID value. A message may be retransmitted for any purpose (including communications failure) and may use the same wsa:MessageID value. The value of this header is an opaque URI whose interpretation beyond equivalence is not defined in this specification. If a reply is expected, this property shall be present.

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

wsa:RelatesTo

This optional (repeating) element indicates how this message relates to another message, in the form of a URI-QName pair. The child of this element (which is of type xs:anyURI) contains the wsa:MessageID of the related message or the following well-known URI that means "unspecified message":

```
http://schemas.xmlsoap.org/ws/2004/08/addressing/id/unspecified
```

A reply message shall contain a wsa:RelatesTo header consisting of wsa:Reply and the wsa:MessageID value of the request message.

wsa:RelatesTo/@RelationshipType

This optional attribute (of type xs:QName) conveys the relationship type as a QName. When absent, the implied value of this attribute is wsa:Reply.

This specification has one predefined relationship type, as shown in Table 1:

Table 1 – Relationship Type

QName	Description
wsa:Reply	Indicates that this is a reply to the message identified by the URI.

wsa:ReplyTo

This optional element (of type wsa:EndpointReferenceType) provides an endpoint reference that identifies the intended receiver for replies to this message. This element shall be present if a reply is expected. If this element is present, wsa:MessageID shall be present. If a reply is expected, a message shall contain a wsa:ReplyTo header. The sender shall use the contents of the wsa:ReplyTo to formulate the reply message as defined in 5.1.3.1. If the wsa:ReplyTo header is absent, the contents of the wsa:From header may be used to formulate a message to the source. This header may be absent if the message has no meaningful reply.

wsa:From

This optional element (of type wsa:EndpointReferenceType) provides a reference to the endpoint where the message originated.

wsa:FaultTo

This optional element (of type wsa:EndpointReferenceType) provides an endpoint reference that identifies the intended receiver for faults related to this message. If this element is present, wsa:MessageID shall be present. When formulating a fault message as defined in 5.1.3.1, the sender shall use the contents of this header to formulate the fault message. If this header is absent, the sender should use the contents of the wsa:ReplyTo header to formulate the fault message. If both the wsa:FaultTo and wsa:ReplyTo header are absent, the sender may use the contents of the wsa:From header to formulate the fault message.

wsa:To

This required element (of type xs:anyURI) provides the address of the intended receiver of this message.

wsa:Action

This required element (of type xs:anyURI) uniquely identifies the semantics implied by this message. It is recommended that the value of this header be a URI identifying an input, output, or fault message within a WSDL port type. An action may be explicitly or implicitly associated with

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

the corresponding WSDL definition. Finally, if in addition to the wsa:Action header, a SOAP Action URI is encoded in a request, the URI of the SOAP Action shall either be the same as the one specified by the wsa:Action header, or set to "".

The dispatching of incoming messages is based on two message properties. The mandatory wsa:To and wsa:Action header identify the target processing location and the verb or intent of the message.

Due to the range of network technologies currently in wide-spread use (for example, NAT, DHCP, and firewalls), many deployments cannot assign a meaningful global URI to a given endpoint. To allow these "anonymous" endpoints to initiate message exchange patterns and receive replies, Addressing defines the following well-known URI for use by endpoints that cannot have a stable, resolvable URI:

```
http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
```

Requests whose wsa:ReplyTo, wsa:From and/or wsa:FaultTo headers use this address shall provide some out-of-band mechanism for delivering replies or faults (for example, returning the reply on the same transport connection). This mechanism may be a simple request/reply transport protocol (for example, HTTP GET or POST). This URI may be used as the wsa:To header for reply messages and should not be used as the wsa:To header in other circumstances.

5.1.3.1 Formulating a Reply Message

The reply to an Addressing compliant request message shall be constructed according to the rules defined in this clause.

EXAMPLE 1: The following example illustrates a request message using message information header blocks in a SOAP 1.2 message:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:f123="http://www.fabrikam123.example/svc53">
  <S:Header>
  <wsa:MessageID>uuid:aaaabbbb-cccc-dddd-eeee-ffffffffffff
    </wsa:MessageID>
    <wsa:ReplyTo>
    <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:To S:mustUnderstand="1">mailto:joe@fabrikam123.example</wsa:To>
    <wsa:Action>http://fabrikam123.example/mail/Delete</wsa:Action>
  </S:Header>
  <S:Body>
    <f123:Delete>
       <maxCount>42</maxCount>
    </f123:Delete>
  </S:Body>
</S:Envelope>
```

EXAMPLE 2: The following example illustrates a reply message using message information header blocks in a SOAP 1.2 message:

```
<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:f123="http://www.fabrikam123.example/svc53">
  <S:Header>
    <wsa:MessageID>
      uuid:aaaabbbb-cccc-dddd-eeee-wwwwwwwwwww
    </wsa:MessageID>
    <wsa:RelatesTo>
```


© ISO/IEC 2013 – All rights reserved

Full document details: BS ISO/IEC 17963:2013

```
      uuid:aaaabbbb-cccc-dddd-eeee-ffffffffffff
    </wsa:RelatesTo>
    <wsa:To>
      http://business456.example/client1
    </wsa:To>
    <wsa:Action>http://fabrikam123.example/mail/DeleteAck</wsa:Action>
  </S:Header>
  <S:Body>
    <f123:DeleteAck/>
  </S:Body>
</S:Envelope>
```

5.1.3.2 Associating Action with WSDL Operations

Addressing defines two mechanisms, explicit association and default action pattern, to associate an action with input, output, and fault elements within a WSDL port type.

5.1.3.2.1 Explicit Association

The action may be explicitly associated using the wsa:Action attribute.

EXAMPLE: Consider the following WSDL excerpt:

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetTradePricesInput"
            wsa:Action="http://example.com/GetQuote"/>
      <output message="tns:GetTradePricesOutput"
            wsa:Action="http://example.com/Quote"/>
    </operation>
  </portType>
  ...
</definitions>
```

The action for the input of the GetLastTradePrice operation within the StockQuotePortType is explicitly defined to be http://example.com/GetQuote. The action for the output of this same operation is http://example.com/Quote.

5.1.3.2.2 Default Action Pattern

In the absence of the wsa:Action attribute, the following pattern is used to construct a default action for inputs and outputs. The general form of an action URI is as follows:

```
targetNamespace/portTypeName/(inputName|outputNname)
```

The "/" is a literal character to be included in the action. The values of the properties are as follows:

- *targetNamespace* is the target namespace (/definition/@targetNamespace). If *target namespace* ends with a "/" an additional "/" is not added.
- *portTypeName* is the name of the port type (/definition/portType/@name).
- (*inputName*|*outputName*) is the name of the element as defined in Section 2.4.5 of WSDL 1.1.

For fault messages, this pattern is not applied. Instead, the following URI is the default action URI for fault messages:

```
http://schemas.xmlsoap.org/ws/2004/08/addressing/fault
```


© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

EXAMPLE: Consider the following WSDL excerpt:

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetTradePricesInput" name="GetQuote"/>
      <output message="tns:GetTradePricesOutput" name="Quote"/>
    </operation>
  </portType>
  ...
</definitions>
```

targetNamespace = http://example.com/stockquote

portTypeName = StockQuotePortType

inputName = GetQuote

outputName = Quote

Applying the preceding pattern with these values produces the following:

input action = http://example.com/stockquote/StockQuotePortType/GetQuote

output action = http://example.com/stockquote/StockQuotePortType/Quote

WSDL defines rules for a default input or output name if the name attribute is not present. Consider the following example:

EXAMPLE: The following is a WSDL excerpt:

```
<definitions targetNamespace="http://example.com/stockquote" ...>
  ...
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetTradePricesInput"/>
      <output message="tns:GetTradePricesOutput"/>
    </operation>
  </portType>
  ...
</definitions>
```

targetNamespace = http://example.com/stockquote

portTypeName = StockQuotePortType

According to the rules defined in 2.4.5 of WSDL, if the name attribute is absent for the input of a request response operation, the default value is the name of the operation with "Request" appended.

inputName = GetLastTradePriceRequest

Likewise, the output defaults to the operation name with "Response" appended.

outputName = GetLastTradePriceResponse

Applying the previous pattern with these values produces the following:

input action = http://example.com/stockquote/StockQuotePortType/GetLastTradePriceRequest

output action = http://example.com/stockquote/StockQuotePortType/GetLastTradePriceResponse

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

5.2 Versions of Addressing

To maintain compatibility with implementations of previous versions of WS-Management, this protocol accommodates messages formatted by those previous versions. However, WS-Management 1.1 also allows for the optional use of the WS-Addressing W3C Recommendation.

The following abbreviations are used for clarity and brevity.

- "WSMA" refers to the version of Management Addressing as specified in 5.1.
- "WSA-Rec" refers to the WS-Addressing W3C Recommendation.
- "WS-Man 1.0" refers to the *WS-Management Specification* 1.0 and implementations compatible with that specification.
- "WS-Man 1.1" refers to this specification and implementations compatible with this specification.
- “Addressing Anonymous URI” refers to the anonymous URI that is defined by the version of Addressing currently in use. The anonymous URI defined by WSA-Rec is http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous. The anonymous URI defined by WSMA is http://www.w3.org/2005/08/addressing/anonymous.

NOTE: Some information in this clause is implementation advice to clients on algorithms for efficient communication with unknown services. This informative advice should not be construed to place normative requirements on the behavior of compliant clients or services.

5.2.1 Technical Differences

The WSMA and WSA-Rec specifications reference different XML namespaces. An endpoint sending Web service messages shall use, for the Addressing SOAP headers, one namespace or the other; a receiving endpoint may recognize one namespace or both namespaces. Existing implementations of WS-Man 1.0 are limited to recognizing only the WSMA namespace. Interactions between WS-Man 1.0 and WS-Man 1.1 implementations will have to allow for these limitations.

5.3 Requirements for Compatibility

To maximize interoperability of WS-Management implementations, WS-Man 1.0 and WS-Man 1.1 clients and services need to be able to exchange messages. These requirements are summarized in Table 2.

Table 2 – Interoperability Requirements

Interoperability Requirements between WS-Management Versions	WS-Man 1.0 Service	WS-Man 1.1 Service
WS-Man 1.0 client	It works.	WS-Man 1.0 client needs to be able to access WS-Man 1.1 service, but some negotiation might be needed.
WS-Man 1.1 client	WSMan 1.1 client needs to be able to access 1.0 service.	It works, but some negotiations might be needed.

Homogeneous pairings of compliant clients and services (that is, a version 1.0 client with a version 1.0 service, or a version 1.1 client with a version 1.1 service) can exchange messages in accordance with their respective specifications. To ensure reliable communications, heterogeneous pairings need to meet certain requirements and implement certain sequencing strategies.

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

In particular, clients and services that implement WS-Man 1.0 can use only WSMA in any exchanges; therefore, all exchanges with version 1.0 endpoints use only WSMA. This conclusion is summarized in Table 3.

Table 3 – WSA Versions in Exchanges

Interoperable Version of Addressing	WS-Man 1.0 Service	WS-Man 1.1 Service
WS-Man 1.0 client	WSMA	WSMA
WS-Man 1.1 client	WSMA	WSMA or WSA-Rec

5.3.1 Discovery or Negotiation

If it is possible for a client to determine the capabilities of the service with respect to WSA, such discovery is more efficient than negotiating the WSA version. For instance, if a service supports Identify, then a client can determine in advance the WS-Man protocol, as well as an Addressing version or versions supported by the service. For this reason, support of Identify is mandatory in this specification when WSA-Rec is used.

Identify would be used as follows:

- The client sends the service an Identify message.
- If the service does not support Identify, the client can conclude that the service is a WS-Man 1.0 implementation and only supports WSMA.
- If the service successfully processes the Identify message, the client examines the versions of Addressing by looking at the AddressingVersionURI element (as defined in clause 11), if present, and can choose the appropriate version.
- If the Identify response message does not contain any Addressing versions, then there is no way for the client to know which version of Addressing to use and it would need to use one of the strategies described in 5.3.2.

In any case, to avoid unnecessary re-discovery or re-negotiation, a WS-Man 1.1 client should retain information about the capabilities of service endpoints where practical.

5.3.2 Client Negotiation Strategies

A compliant WS-Man 1.0 client will use only WSMA in message exchanges. A WS-Man 1.1 client, however, may use either WSMA or WSA-Rec in message exchanges. If a WS-Man 1.1 client does not know the WSA version capabilities of a service, it may use different strategies when initially contacting the service. The client may begin a message exchange with either version of WSA, using WSA-Rec or WSMA in the request message. The message exchange would proceed as follows:

- Strategy type 1: A client sends the request using WSA-Rec. The WSA-Rec SOAP headers need to be marked with a mustUnderstand="1" attribute to ensure that a fault will be generated if the receiver does not support the WSA-Rec version of Addressing. The client can then retry the operation using WSMA.
- Strategy type 2: A client sends the request using WSMA. Both WS-Man 1.0 services and WS-Man 1.1 services respond to the request using WSMA.

5.3.3 Initiating Message Exchanges

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

Outgoing messages initiated by a WS-Man implementation need to use the same version of Addressing that was used in the Endpoint Reference to which those messages are being sent. For example, if a Subscribe request message uses WSA-Rec in the SOAP headers (for example, for the wsa:To and wsa:ReplyTo), but uses WSMA for the NotifyTo EPR, then the Subscribe response will be sent using WSA-Rec, but the events will be sent using WSMA.

5.3.4 Normative Rules

R5.3.4-1: If a WS-Man 1.1 service supports WSA-Rec, then it shall also support the Identify operation.

R5.3.4-2: A WS-Man 1.1 service shall support WSMA and should support WSA-Rec.

R5.3.4-3: A WS-Man 1.1 implementation shall send messages to endpoints using the same version of Addressing used in the Endpoint Reference of the destination endpoint (see 5.2).

R5.3.4-4: Within a single SOAP message, a WS-Man 1.1 implementation shall use the same version of Addressing for all Addressing SOAP headers.

Because WS-Man 1.1 allows for either version of Addressing to be used, R5.3.4-4 removes the possibility of mixing the two versions for the WSA SOAP headers, but it does not disallow Endpoint References that might appear elsewhere in the message to be of a different version.

In order to provide a migration path from the WSMA to WSA-Rec, the schema of certain messages allows for either version's EndpointReferenceType to be used. While the schema itself is written in a very generic way (that is, using an xs:any) allowing any arbitrary XML to appear, implementations shall restrict the contents of this element to one of the EndpointReference Types.

NOTE: This allows existing WS-Man 1.0 implementations to be compliant, while providing newer implementations a migration path. In this spirit, newer implementations are strongly encouraged to support both versions of Addressing.

5.4 Use of Addressing in WS-Management

This clause describes the use of Endpoint References regardless of whether an implementation uses WS-Management Addressing (see 5.1) or the W3C Recommendation version of WS-Addressing.

Addressing (either addressing type) endpoint references (EPRs) are used to convey information needed to address a Web service endpoint. WS-Management defines a default addressing model that can optionally be used in EPRs.

5.4.1 Use of Endpoint References

WS-Management uses EPRs as the addressing mechanism for individual resource instances. WS-Management also defines a default addressing model for use in addressing resources. In cases where this default addressing model is not appropriate, such as in systems with well-established addressing models or with EPRs retrieved from a discovery service, services may use those service-specific addressing models if they are based on either addressing version supported by WS-Management.

R5.4.1-1: All messages that are addressed to a resource class or instance that is referenced by an EPR must follow the Addressing rules for representing content from the EPR (the address and reference parameters) in the SOAP message. This rule also applies to continuation messages such as Pull or Release, which continue an operation begun in a previous message. Even though such messages contain contextual information that binds them to a previous operation, the information from the EPR is still required in the message to help route it to the correct handler.

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

Rule R5.4.1-1 clarifies that messages such as Pull or Renew still require a full EPR. For Pull, for example, this EPR would be the same as the original Enumerate, even though EnumerateResponse returns a context object that would seem to obviate the need for the EPR. The EPR is still required to route the message properly. Similarly, the Renew request uses the SubscriptionManager EPR received in the SubscribeResponse.

When a service includes an EPR in a response message, it must be willing to accept subsequent request messages targeted to that EPR for the same individual managed resource. Clients are not required to process or enhance EPRs given to them by the service before using them to address a managed resource.

R5.4.1-2: An EPR returned by a service shall be acceptable to that service to refer to the same managed resource.

R5.4.1-3: All EPRs returned by a service, whether expressed using the WS-Management default addressing model (see 5.4.2) or any other addressing model, shall be valid as long as the managed resource exists.

5.4.2 WS-Management Default Addressing Model

WS-Management defines a default addressing model for resources. A service is not required to use this addressing model, but it is suitable for many new implementations and can increase the chances of successful interoperation between clients and services.

This document uses examples of this addressing model that contain its component parts, the ResourceURI and SelectorSet SOAP headers. This specification is independent of the actual data model and does not define the structure of the ResourceURI or the set of values for selectors for a given resource. These may be vendor specific or defined by other specifications.

Description and use of this addressing model in this specification do not indicate that support for this addressing model is a requirement for a conformant service.

All of the normative text, examples, and conformance rules in 5.4.2 and 5.4.2.2 presume that the service is based on the default addressing model. In cases where this addressing model is not in use, these rules do not apply.

The default addressing model uses a representation of an EPR that is a tuple of the following SOAP headers:

- **wsa:To** (required): the transport address of the service
- **wsman:ResourceURI** (required if the default addressing model is used): the URI of the resource class representation or instance representation
- **wsman:SelectorSet** (optional): a header that identifies or "selects" the resource instance to be accessed if more than one instance of a resource class exists

The wsman:ResourceURI value needs to be marked with an s:mustUnderstand attribute set to "true" in all messages that use the default addressing model. Otherwise, a service that does not understand this addressing model might inadvertently return a resource that was not requested by the client.

The WS-Management default addressing model is defined in the following XML outline for an EPR:

```
(1)  <wsa:EndpointReference>
(2)    <wsa:Address>
(3)      Network address
(4)    </wsa:Address>
(5)    <wsa:ReferenceParameters>
(6)      <wsman:ResourceURI> resource URI </wsman:ResourceURI>
(7)      <wsman:SelectorSet>
```


© ISO/IEC 2013 – All rights reserved

Full document details: BS ISO/IEC 17963:2013

```
(8)       <wsman:Selector Name="selector-name"> *
(9)         Selector-value
(10)      </wsman:Selector>
(11)    </wsman:SelectorSet> ?
(12)  </wsa:ReferenceParameters>
(13) </wsa:EndpointReference>
```

The following definitions provide additional, normative constraints on the preceding outline:

wsa:Address

the URI of the transport address

wsa:ReferenceParameters/wsman:ResourceURI

the URI of the resource class or instance to be accessed

Typically, this URI represents the resource class, but it may represent the instance. The combination of this URI and the wsa:To URI form the full address of the resource class or instance.

wsa:ReferenceParameters/wsman:SelectorSet:

the optional set of selectors as described in 5.4.2.2

These values are used to select an instance if the ResourceURI identifies a multi-instanced target.

When the default addressing model is used in a SOAP message, Addressing specifies that translations take place and the headers are flattened out.

EXAMPLE: The following is an example EPR definition:

```
(1)  <wsa:EndpointReference>
(2)    <wsa:Address> Address </wsa:Address>
(3)    <wsa:ReferenceParameters xmlns:wsman="...">
(4)      <wsman:ResourceURI>resURI</wsman:ResourceURI>
(5)      <wsman:SelectorSet>
(6)        <wsman:Selector Name="Selector-name">
(7)          Selector-value
(8)        </wsman:Selector>
(9)      </wsman:SelectorSet>
(10)   </wsa:ReferenceParameters>
(11) </wsa:EndpointReference>
```

This address definition is translated as follows when used in a SOAP message. wsa:Address becomes wsa:To, and the reference parameters are unwrapped and juxtaposed. The following example shows a sample SOAP message using WSMA:

```
(1) <s:Envelope xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
(2)  <s:Header>
(3)    <wsa:To> Address </wsa:To>
(4)    <wsa:Action> Action URI </wsa:Action>
(5)    <wsman:ResourceURI s:mustUnderstand="true">resURI</wsman:ResourceURI>
(6)    <wsman:SelectorSet>
(7)      <wsman:Selector Name="Selector-name">
(8)        Selector-value
(9)      </wsman:Selector>
(10)   </wsman:SelectorSet>
(11)   ...
(12)  </s:Header>
(13)  <s:Body> ... </s:Body>
(14) </s:Envelope>
```


© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

The following message shows a sample SOAP message using WS-Rec:

```
(1) <s:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing ">
(2)   <s:Header>
(3)     <wsa:To s:mustUnderstand="true"> Address </wsa:To>
(4)     <wsa:Action s:mustUnderstand="true"> Action URI </wsa:Action>
(5)     <wsman:ResourceURI s:mustUnderstand="true"
(6)        wsa:isReferenceParameter="true">resURI</wsman:ResourceURI>
(7)     <wsman:SelectorSet wsa:isReferenceParameter="true">
(8)       <wsman:Selector Name="Selector-name">
(9)         Selector-value
(10)        </wsman:Selector>
(11)    </wsman:SelectorSet>
(12)    ...
(13)   </s:Header>
(14)   <s:Body> ... </s:Body>
(15) </s:Envelope>
```

In both cases, the wsa:To, wsman:ResourceURI, and wsman:SelectorSet elements work together to *reference* the resource instance to be managed, but the actual *method* or *operation* to be executed against this resource is indicated by the wsa:Action header.

EXAMPLE: The following is an example of Addressing headers based on the default addressing model in an actual message:

```
(1)  <s:Envelope
(2)    xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(3)    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(4)    xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
(5)    <s:Header>
(6)      ...
(7)      <wsa:To>http://123.99.222.36/wsman</wsa:To>
(8)      <wsman:ResourceURI s:mustUnderstand="true">
(9)        http://example.org/hardware/2005/02/storage/physDisk
(10)     </wsman:ResourceURI>
(11)     <wsman:SelectorSet>
(12)       <wsman:Selector Name="LUN"> 2 </wsman:Selector>
(13)     </wsman:SelectorSet>
(14)     <wsa:Action> http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
           </wsa:Action>
(15)     <wsa:MessageID> urn:uuid:d9726315-bc91-430b-9ed8-ce5ffb858a91
           </wsa:MessageID>
(16)     ...
(17)   </s:Header>
(18)   <s:Body> ... </s:Body>
(19) </s:Envelope>
```

The following definitions apply to the preceding message example:

wsa:To
: the network (or transport-level) address of the service

wsman:ResourceURI
: the ResourceURI of the resource class or resource instance to be accessed

wsman:SelectorSet
: a wrapper for the selectors

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

wsman:SelectorSet/wsman:Selector

identifies or selects the resource instance to be accessed, if more than one instance of the resource exists

In this case, the selector is "LUN" (logical unit number), and the selected device is unit number "2".

wsa:Action

identifies which operation is to be carried out against the resource (in this case, a "Get")

wsa:MessageID

identifies this specific message uniquely for tracking and correlation purposes

The format defined in RFC 4122 is often used in the examples in this specification, but it is not required.

5.4.2.1 ResourceURI

The ResourceURI is used to indicate the class resource or instance.

R5.4.2.1-1: The format of the wsman:ResourceURI is unconstrained provided that it meets RFC 3986 requirements.

The format and syntax of the ResourceURI is any valid URI according to RFC 3986. Although there is no default scheme, http: and urn: are common defaults. If http: is used, users may expect to find Web-based documentation of the resource at that address. The wsa:To and the wsman:ResourceURI elements work together to define the actual resource being targeted.

R5.4.2.1-2: Vendor-specific or organization-specific URIs should contain the Internet domain name in the first token sequence after the scheme, such as "example.org" in ResourceURI in the following example.

EXAMPLE:

```
(20)   <s:Header>
(21)     <wsa:To> http://123.15.166.67/wsman </wsa:To>
(22)     <wsman:ResourceURI>
(23)       http//schemas.example.org/2005/02/hardware/physDisk
(24)     </wsman:ResourceURI>
(25)     ...
(26)   </s:Header>
```

R5.4.2.1-3: When the default addressing model is used, the wsman:ResourceURI reference parameter is required in messages with the following wsa:Action URIs:

http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
http://schemas.xmlsoap.org/ws/2004/09/transfer/Put
http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Renew
http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatus
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release
http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

The following messages require the EPR to be returned in the SubscriptionManager element of the SubscribeResponse message. The format of the EPR is determined by the service and might or might not include the ResourceURI:

http://schemas.xmlsoap.org/ws/2004/08/eventing/Renew
http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatus

While the ResourceURI SOAP header is required when the WS-Management default addressing mode is used, it may be short and of a very simple form, such as http://example.com/* or http://example.com/resource.

R5.4.2.1-4: For the request message of custom actions (methods), the ResourceURI header may be present in the message to help route the message to the correct handler.

R5.4.2.1-5: The ResourceURI element should not appear in other messages, such as responses or events, unless the associated EPR includes it in its ReferenceParameters.

In practice, the wsman:ResourceURI element is required only in requests to reference the targeted resource class. Responses are not addressed to a management resource, so the wsman:ResourceURI has no meaning in that context.

R5.4.2.1-6: When the default addressing model is used and the wsman:ResourceURI element is missing or in an incorrect form, the service shall issue a wsa:DestinationUnreachable fault with a detail code of

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/InvalidResourceURI

R5.4.2.1-7: The wsman:ResourceURI element shall be used to indicate only the identity of a resource, and it may not be used to indicate the action being applied to that resource, which is properly expressed using the wsa:Action URI.

Custom WSDL-based methods have both a ResourceURI identity from the perspective of addressing and a wsa:Action URI from the perspective of execution. In many cases, the ResourceURI is simply a pseudonym for the WSDL identity and Port, and the wsa:Action URI is the specific method within that port (or interface) definition.

Although a single URI could theoretically be used alone to define an instance of a multi-instance resource, it is recommended that the wsa:To element be used to locate the WS-Management service, that the wsman:ResourceURI element be used to identify the resource class, and that the wsman:SelectorSet element be used to reference the resource instance. If the resource consists of only a single instance, then the wsman:ResourceURI element alone refers to the single instance.

This usage is not a strict requirement, just a guideline. The service can use distinct selectors for any given operation, even against the same resource class, and may allow or require selectors for the Enumerate operation.

See the recommendations in 7.2 regarding addressing uniformity.

Custom actions have two distinct identities: the ResourceURI, which can identify the WSDL and port (or interface), and the wsa:Action URI, which identifies the specific method. If only one method exists in the interface, in a sense the ResourceURI and wsa:Action URI are identical.

It is not an error to use the wsa:Action URI for the ResourceURI of a custom method, but both are still required in the message for uniform processing on both clients and servers.

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

EXAMPLE 1: The following action to reset a network card might have the following EPR usage:

```
(1)  <s:Header>
(2)    <wsa:To>
(3)      http://1.2.3.4/wsman/
(4)    </wsa:To>
(5)    <wsman:ResourceURI>http://example.org/2005/02/networkcards/reset
         </wsman:ResourceURI>
(6)    <wsa:Action>
(7)      http://example.org/2005/02/networkcards/reset
(8)    </wsa:Action>
(9)    ...
(10) </s:Header>
```

In many cases, the ResourceURI is equivalent to a WSDL name and port, and the wsa:Action URI contains an additional token as a suffix, as in the following example.

EXAMPLE 2:

```
(1)  <s:Header>
(2)    <wsa:To>
(3)      http://1.2.3.4/wsman
(4)    </wsa:To>
(5)    <wsman:ResourceURI>http://example.org/2005/02/networkcards
         </wsman:ResourceURI>
(6)    <wsa:Action>
(7)      http://example.org/2005/02/networkcards/reset
(8)    </wsa:Action>
(9)    ...
(10) </s:Header>
```

Finally, the ResourceURI may be completely unrelated to the wsa:Action URI, as in the following example.

EXAMPLE 3:

```
(1)  <s:Header>
(2)    <wsa:To>http://1.2.3.4/wsman</wsa:To>
(3)    <wsman:ResourceURI>
(4)      http://example.org/products/management/networkcards
(5)    </wsman:ResourceURI>
(6)    <wsa:Action>
(7)      http://example.org/2005/02/netcards/reset
(8)    </wsa:Action>
(9)    ...
(10) </s:Header>
```

All of these uses are legal.

When used with subscriptions, the EPR described by wsa:Address and wsman:ResourceURI (and optionally the wsman:SelectorSet values) identifies the event source to which the subscription is directed. In many cases, the ResourceURI identifies a real or virtual event log, and the subscription is intended to provide real-time notifications of any new entries added to the log. In many cases, the wsman:SelectorSet element might not be used as part of the EPR.

5.4.2.2 Selectors

In the WS-Management default addressing model, selectors are optional elements used to identify instances within a resource class. For operations such as Get or Put, the selectors are used to identify a single instance of the resource class referenced by the ResourceURI.

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

In practice, because the ResourceURI often acts as a table or a "class," the SelectorSet element is a discriminant used to identify a specific "row" or "instance." If only one instance of a resource class is implied by the ResourceURI, the SelectorSet can be omitted because the ResourceURI is acting as the full identity of the resource. If more than one selector value is required, the entire set of selectors is interpreted by the service in order to reference the specific instance. The selectors are interpreted as being separated by implied logical AND operators.

In some information domains, the values referenced by the selectors are "keys" that are part of the resource content itself, whereas in other domains the selectors are part of a logical or physical directory system or search space. In these cases, the selectors are used to identify the resource, but are not part of the representation.

R5.4.2.2-1: If a resource has more than one instance, a wsman:SelectorSet element may be used to distinguish which instance is targeted if the WS-Management default addressing model is in use. Any number of wsman:Selector values may appear with the wsman:SelectorSet element, as required to identify the precise instance of the resource class. The service may consider the case of selector names and values (see 13.6), as required by the underlying execution environment.

If the client needs to discover the policy on how the case of selector values is interpreted, the service can provide metadata documents that describe this policy. The format of such metadata is beyond the scope of this specification.

R5.4.2.2-2: All content within the SelectorSet element is to be treated as a single reference parameter with a scope relative to the ResourceURI.

R5.4.2.2-3: A service using the WS-Management default addressing model shall examine all selectors in the message and process them as if they were logically joined by AND. If the set of selectors is incorrect for the targeted resource instance, a wsman:InvalidSelectors fault should be returned to the client with the following detail codes:

- if selectors are missing:

  http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/InsufficientSelectors

- if selector values are the wrong types:

  http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/TypeMismatch

- if the selector value is of the correct type from the standpoint of XML types, but out of range or otherwise illegal in the specific information domain:

  http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/InvalidValue

- if the name is not a recognized selector name

  http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/UnexpectedSelectors

R5.4.2.2-4: The Selector Name attribute shall not be duplicated at the same level of nesting. If this occurs, the service should return a wsman:InvalidSelectors fault with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/DuplicateSelectors

This specification does not mandate the use of selectors. Some implementations may decide to use complex URI schemes in which the ResourceURI itself implicitly identifies the instance.

© ISO/IEC 2013 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

The format of the SelectorSet element is as follows:

```
(1)  <s:Envelope
(2)     xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(3)     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(4)     xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
(5)    <s:Header>
(6)     ...
(7)     <wsa:To>  service transport address </wsa:To>
(8)     <wsman:ResourceURI> ResourceURI </wsman:ResourceURI>
(9)     <wsman:SelectorSet>
(10)      <wsman:Selector Name="name"> value </wsman:Selector> +
(11)    </wsman:SelectorSet> ?
(12)    ...
(13)   </s:Header>
(14)   <s:Body> ... </s:Body>
(15) </s:Envelope>
```

The following definitions provide additional, normative constraints on the preceding outline:

wsman:SelectorSet

the wrapper for one or more Selector elements required to reference the instance

wsman:SelectorSet/wsman:Selector

used to describe the selector and its value

If more than one selector is required, one Selector element exists for each part of the overall selector. The value of this element is the Selector value.

wsman:SelectorSet/wsman:Selector/@Name

the name of the selector (to be treated in a case-insensitive manner)

The value of a selector may be a nested EPR.

EXAMPLE: In the following example, the selector on line 9 is a part of a SelectorSet that contains a nested EPR (lines 10–18) with its own Address, ResourceURI, and SelectorSet elements:

```
(1)  <s:Envelope
(2)     xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(3)     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(4)     xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
(5)    <s:Header>
(6)     ...
(7)     <wsman:SelectorSet>
(8)       <wsman:Selector Name="Primary"> 123 </wsman:Selector>
(9)       <wsman:Selector Name="EPR">
(10)        <wsa:EndpointReference>
(11)          <wsa:Address> address </wsa:Address>
(12)          <wsa:ReferenceParameters>
(13)            <wsman:ResourceURI> resource URI </wsman:ResourceURI>
(14)            <wsman:SelectorSet>
(15)              <wsman:Selector Name="name"> value </wsman:Selector>
(16)            </wsman:SelectorSet>
(17)          </wsa:ReferenceParameters>
(18)        </wsa:EndpointReference>
(19)      </wsman:Selector>
(20)    </wsman:SelectorSet>
(21)    ...
(22)   </s:Header>
(23)   <s:Body> ... </s:Body>
(24) </s:Envelope>
```


© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

R5.4.2.2-5: For those services using the WS-Management default addressing model, the value of a wsman:Selector shall be one of the following values:

- a simple type as defined in the XML schema namespace http://www.w3.org/2001/XMLSchema
- a nested wsa:EndpointReference using the WS-Management default addressing model

A service may fault selector usage with wsman:InvalidSelectors if the selector is not a simple type or an EPR.

R5.4.2.2-6: A conformant service may reject any selector or nested selector with a nested EPR whose wsa:Address value is not the same as the primary wsa:To value or is not the Addressing Anonymous URI.

The primary purpose for this nesting mechanism is to allow resources that can answer questions about other resources.

R5.4.2.2-7: A service may fail to process a selector name of more than 2048 characters.

R5.4.2.2-8: A service may fail to process a selector value of more than 4096 characters, including any embedded selectors, and may fail to process a message that contains more than 8096 characters of content in the root SelectorSet element.

5.4.2.3 Faults for Default Addressing Model

When faults related to the information in the addressing model based on the default format are generated, they may contain specific fault detail codes. These detail codes are called out separately in 14.6 and do not apply when service-specific addressing is used.

5.4.3 Service-Specific Endpoint References

Although WS-Management specifies a default addressing model, in some cases this model is not available or appropriate.

R5.4.3-1: A conformant service may not understand the header values used by the WS-Management default addressing model. If this is the case, and if the client marks the wsman:ResourceURI with mustUnderstand="true", the service shall return an s:NotUnderstood fault.

R5.4.3-2: A conformant service may require additional header values to be present that are beyond the scope of this specification.

Services can thus use alternative addressing models for referencing resources with WS-Management. These addressing models might or might not use ResourceURI or SelectorSet elements and still be valid addressing models if they conform to the rules of Addressing.

In addition to a defined alternative addressing model, a service might not explicitly define any addressing model at all and instead use an opaque EPR generated at run-time, which is handled according to the standard rules of Addressing.

When such addressing models are used, the client application has to understand and interoperate with discovery methods for acquiring EPRs that are beyond the scope of this specification.

5.4.4 mustUnderstand

This clause describes the use of the mustUnderstand attribute, regardless of whether an implementation uses WS-Management Addressing (see 5.1) or the W3C Recommendation type of WS-Addressing.

© ISO/IEC 2013 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

The mustUnderstand attribute for SOAP headers is to be interpreted as a "must comply" instruction in WS-Management. For example, if a SOAP header that is listed as being optional in this specification is tagged with mustUnderstand="true", the service is required to comply or return a fault. To ensure that the service treats a header as optional, the mustUnderstand attribute can be omitted.

If the wsa:Action URI is not understood, the implementation might not know how to process the message. So, for the following elements, the omission or inclusion of mustUnderstand="true" has no real effect on the message in practice, because mustUnderstand is implied:

- wsa:To
- wsa:MessageID
- wsa:RelatesTo
- wsa:Action
- wsa:ReplyTo
- wsa:FaultTo

R5.4.4-1: A conformant service shall process any of the preceding elements identically regardless of whether mustUnderstand="true" is present.

As a corollary, clients can omit mustUnderstand="true" from any of the preceding elements with no change in meaning.

R5.4.4-2: If a service cannot comply with a header marked with mustUnderstand="true", it shall issue an s:NotUnderstood fault.

The goal is for the service to be tolerant of inconsistent mustUnderstand usage by clients when the request is not likely to be misinterpreted.

It is important that clients using the WS-Management default addressing model (ResourceURI and SelectorSet) use mustUnderstand="true" on the wsman:ResourceURI element to ensure that the service is compliant with that addressing model. Implementations that use service-specific addressing models will otherwise potentially ignore these header values and behave inconsistently with the intentions of the client.

5.4.5 wsa:To

This clause describes the use of the Addressing wsa:To header regardless of whether an implementation uses WS-Management Addressing (see 5.1) or the W3C Recommendation version of WS-Addressing.

In request messages, the wsa:To address contains the transport address of the service. In some cases, this address is sufficient to locate the resource. In other cases, the service is a dispatching agent for multiple resources. In these cases, the message typically contains additional headers to allow the service to identify a resource within its scope. For example, when the default addressing model is in use, these additional headers will be the ResourceURI and SelectorSet elements.

NOTE: WS-Management does not preclude multiple listener services from coexisting on the same physical system. Such services would be discovered and distinguished using mechanisms beyond the scope of this specification.

R5.4.5-1: The wsa:To header shall be present in all messages, whether requests, responses, or events. In the absence of other requirements, it is recommended that the network address for resources that require authentication be suffixed by the token sequence */wsman*. If */wsman* is used, unauthenticated access should not be allowed.

```
(1) <wsa:To> http://123.15.166.67/wsman </wsa:To>
```


© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

R5.4.5-2: In the absence of other requirements, it is recommended that the network address for resources that do not require authentication be suffixed by the token sequence */wsman-anon*. If */wsman-anon* is used, authenticated access shall not be required.

```
(1) <wsa:To> http://123.15.166.67/wsman-anon </wsa:To>
```

Including the network transport address in the SOAP message may seem redundant because the network connection would already be established by the client. However, in cases where the message is routed through intermediaries, the network transport address is required so that the intermediaries can examine the message and make the connection to the actual endpoint.

The wsa:To header may encompass any number of tokens required to locate the service and a group of resources within that service.

R5.4.5-3: The service should generate a fault when the wsa:To address cannot be processed due to the following situations::

- If the resource is offline, a wsa:EndpointUnavailable fault is returned with the following detail code:

  http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/ResourceOffline

- If the resource cannot be located ("not found"), a wsa:DestinationUnreachable fault is returned.
- If the resource is valid, but internal errors occur, a wsman:InternalError fault is returned.
- If the resource cannot be accessed for security reasons, a wsman:AccessDenied fault is returned.

5.4.6 Other Addressing Headers

This clause describes the use of other Addressing headers, regardless of whether an implementation uses WS-Management Addressing (see 5.1) or the W3C Recommendation version of WS-Addressing.

WS-Management depends on Addressing to describe the rules for use of other Addressing headers.

5.4.6.1 Processing Addressing Headers

The following additional addressing-related header blocks occur in WS-Management messages.

R5.4.6.1-1: A conformant service shall recognize and process the following Addressing header blocks.

- **wsa:To**
- **wsa:ReplyTo** (required when a response is expected)
- **wsa:FaultTo** (optional)
- **wsa:MessageID** (required)
- **wsa:Action** (required)
- **wsa:RelatesTo** (required in responses)

The use of these header blocks is discussed in subsequent clauses.

© ISO/IEC 2013 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

5.4.6.2 wsa:ReplyTo

WS-Management requires the following usage of wsa:ReplyTo in addressing:

R5.4.6.2-1: A wsa:ReplyTo header shall be present in all request messages when a reply is required. This address shall be either a valid address for a new connection using any transport supported by the service or the Addressing Anonymous URI, which indicates that the reply is to be delivered over the same connection on which the request arrived. If the wsa:ReplyTo header is missing, a wsa:MessageInformationHeaderRequired fault is returned.

Some messages, such as event deliveries, SubscriptionEnd, and so on, do not require a response and may omit a wsa:ReplyTo element.

R5.4.6.2-2: A conformant service may require that all responses be delivered over the same connection on which the request arrives. In this case, the URI discussed in R5.4.6.2-1 shall indicate this. Otherwise, the service shall return a wsman:UnsupportedFeature fault with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/AddressingMode

R5.4.6.2-3: When delivering events for which acknowledgement of delivery is required, the sender of the event shall include a wsa:ReplyTo element and observe the usage in 10.8 of this specification.

R5.4.6.2-4: This rule intentionally left blank.

R5.4.6.2-5: This rule intentionally left blank.

Addressing allows clients to include client-defined reference parameters in wsa:ReplyTo headers. Addressing requires that these reference parameters be extracted from requests and placed in the responses by removing the ReferenceParameters wrapper and placing all of the values as top-level SOAP headers in the response, as discussed in 5.1. This allows clients to better correlate responses with the original requests. This step cannot be omitted.

EXAMPLE: In the following example, the header x:someHeader is included in the reply message:

```
(1)  <s:Envelope
(2)    xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(3)    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(4)    xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
(5)    <s:Header>
(6)      ...
(7)      <wsa:To> http://1.2.3.4/wsman </wsa:To>
(8)      <wsa:ReplyTo>
(9)        <wsa:Address>
(10)         http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
(11)       </wsa:Address>
(12)       <wsa:ReferenceParameters>
(13)         <x:someHeader xmlns:x="..."> user-defined content </x:someHeader>
(14)       </wsa:ReferenceParameters>
(15)     </wsa:ReplyTo>
(16)     ...
(17)   </s:Header>
(18)   <s:Body> ... </s:Body>
(19) </s:Envelope>
```

R5.4.6.2-6: If the wsa:ReplyTo address is not usable or is missing, the service should not reply to the request and it should close or terminate the connection according to the rules of the current network transport. In these cases, the service should locally log some type of entry to help locate the client defect later.

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

5.4.6.3 wsa:FaultTo

WS-Management qualifies the use of wsa:FaultTo as indicated in this clause.

R5.4.6.3-1: A conformant service may support a wsa:FaultTo address that is distinct from the wsa:ReplyTo address. If such a request is made and is not supported by the service, a wsman:UnsupportedFeature fault shall be returned with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/AddressingMode

If both the wsa:FaultTo and wsa:ReplyTo headers are omitted from a request, transport-level mechanisms are typically used to fail the request because the address to which the fault is to be sent is uncertain. In such a case, it is not an error for the service to simply shut down the connection.

R5.4.6.3-2: If wsa:FaultTo is omitted, the service shall return the fault to the wsa:ReplyTo address if a fault occurs.

R5.4.6.3-3: A conformant service may 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 shall be the Addressing Anonymous URI. If services do not support separately addressed fault delivery and the wsa:FaultTo is any other address, a wsman:UnsupportedFeature fault shall be returned with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/AddressingMode

NOTE: This specification does not restrict richer implementations from fully supporting wsa:FaultTo.

R5.4.6.3-4: This rule intentionally left blank.

EXAMPLE: In the following example, the header x:someHeader is included in fault messages if they occur:

```
(1)  <s:Envelope
(2)      xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(3)      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(4)      xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
(5)    <s:Header>
(6)      ...
(7)      <wsa:To> http://1.2.3.4/wsman </wsa:To>
(8)      <wsa:FaultTo>
(9)        <wsa:Address>
(10)         http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
(11)       </wsa:Address>
(12)       <wsa:ReferenceParameters>
(13)         <x:someHeader xmlns:x="..."> user-defined content </x:someHeader>
(14)       </wsa:ReferenceParameters>
(15)     </wsa:FaultTo>
(16)     ...
(17)   </s:Header>
(18)   <s:Body> ... </s:Body>
(19) </s:Envelope>
```

R5.4.6.3-5: If the wsa:FaultTo address is not usable, the service should not reply to the request. Similarly, if according to WS-Addressing processing rules there is no suitable address to send a fault to, it should not reply and should close the network connection. In these cases, the service should locally log some type of entry to help locate the client defect later.

R5.4.6.3-6: The service shall properly duplicate the wsa:Address of the wsa:FaultTo element in the wsa:To of the reply, even if some of the information is not understood by the service.

© ISO/IEC 2013 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

This rule applies in cases where the client includes private content suffixes on the HTTP or HTTPS address that the service does not understand. If the service removes this information when constructing the address, the subsequent message might not be correctly processed.

5.4.6.4 wsa:MessageID and wsa:RelatesTo

WS-Management qualifies the use of wsa:MessageID and wsa:RelatesTo as follows:

R5.4.6.4-1: The MessageID and RelatesTo URIs may be of any format, as long as they are valid URIs according to RFC 3986. Two URIs are considered different even if the characters in the URIs differ only by case.

The following two formats are endorsed by this specification. The first is considered a best practice because it is backed by RFC 4122:

urn:uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
or
uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

In these formats, each *x* is an uppercase or lowercase hexadecimal digit (lowercase is required by RFC 4122); there are no spaces or other tokens. The value may be a DCE-style universally unique identifier (UUID) with provable uniqueness properties in this format, however, it is not necessary to have provable uniqueness properties in the URIs used in the wsa:MessageID and wsa:RelatesTo headers.

Regardless of format, the URI should not exceed the maximum defined in R13.1-6.

UUIDs have a numeric meaning as well as a string meaning, and this can lead to confusion. A UUID in lowercase is a different URI from the same UUID in uppercase. This is because URIs are case-sensitive. If a UUID is converted to its decimal equivalent the case of the original characters is lost. WS-Management works with the URI value itself, not the underlying decimal equivalent representation. Services are free to *interpret* the URI in any way, but are not allowed to alter the case usage when repeating the message or any of the MessageID values in subsequent messages.

The RFC 4122 requires the digits to be lowercase, which is the responsibility of the client. The service simply processes the values as URI values and is not required to analyze the URI for correctness or compliance. The service replicates the client usage in the wsa:RelatesTo reply header and is not allowed to alter the case usage.

R5.4.6.4-2: The MessageID should be generated according to any algorithm that ensures that no two MessageIDs are repeated. Because the value is treated as case-sensitive (R5.4.6.4-1), confusion can arise if the same value is reused differing only in case. As a result, the service shall not create or employ MessageID values that differ only in case. For any message transmitted by the service, the MessageID shall not be reused.

The client ensures that MessageID values are not reused in requests. Although services and clients can issue different MessageIDs that differ only in case, the service is not required to detect this difference, nor is it required to analyze the URI for syntactic correctness or repeated use.

R5.4.6.4-3: The RelatesTo element shall be present in all response messages and faults, shall contain the MessageID of the associated request message, and shall match the original in case, being treated as a URI value and not as a binary UUID value.

R5.4.6.4-4: If the MessageID is not parsable or is missing, a wsa:InvalidMessageInformationHeader fault should be returned.

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

EXAMPLE: The following examples show wsa:MessageID usage:

```
(20)  <wsa:MessageID>
(21)    uuid:d9726315-bc91-430b-9ed8-ce5ffb858a91
(22)  </wsa:MessageID>
(23)
(24)  <wsa:MessageID>
(25)    anotherScheme:ID/12310/1231/16607/25
(26)  </wsa:MessageID>
```

5.4.6.5 wsa:Action

The wsa:Action URI indicates the "operation" being invoked against the resource.

R5.4.6.5-1: The wsa:Action URI shall not be used to identify the specific resource class or instance, but only to identify the operation to use against that resource.

R5.4.6.5-2: For all resource endpoints, a service shall return a wsa:ActionNotSupported fault if a requested action is not supported by the service for the specified resource.

In other words, to model the "Get" of item "Disk", the wsa:Action URI contains the "Get". The wsa:To, and potentially other SOAP headers, indicate *what* is being accessed. When the default addressing model is used, for example, the ResourceURI typically contains the reference to the "Disk" and the SelectorSet identifies which disk. Other service-specific addressing models can factor the identity of the resource in different ways.

Implementations are free to support additional custom methods that combine the notion of "Get" and "Disk" into a single "GetDisk" action if they strive to support the separated form to maximize interoperation. One of the main points behind WS-Management is to unify common methods wherever possible.

R5.4.6.5-3: If a service exposes any of the following types of capabilities, a conformant service shall at least expose that capability using the definitions in Table 4 according to the rules of this specification. The service may optionally expose additional similar functionality using a distinct wsa:Action URI.

Table 4 – wsa:Action URI Descriptions

Action URI	Description
http://schemas.xmlsoap.org/ws/2004/09/transfer/Get	Models any simple single item retrieval
http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse	Response to "Get"
http://schemas.xmlsoap.org/ws/2004/09/transfer/Put	Models an update of an entire item
http://schemas.xmlsoap.org/ws/2004/09/transfer/PutResponse	Response to "Put"
http://schemas.xmlsoap.org/ws/2004/09/transfer/Create	Models creation of a new item
http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse	Response to "Create"
http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete	Models the deletion of an item
http://schemas.xmlsoap.org/ws/2004/09/transfer/DeleteResponse	Response to "Delete"
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate	Begins an enumeration or query
http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerateResponse	Response to "Enumerate"
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull	Retrieves the next batch of results from enumeration
http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse	Response to "Pull"

© ISO/IEC 2013 – All rights reserved

Full document details: BS ISO/IEC 17963:2013

Action URI	Description
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Renew	Renews an enumerator that may have timed out (not required in WS-Management)
http://schemas.xmlsoap.org/ws/2004/09/enumeration/RenewResponse	Response to "Renew" (not required in WS-Management)
http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatus	Gets the status of the enumerator (not required in WS-Management)
http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatusResponse	Response to "GetStatus" (not required in WS-Management)
http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release	Releases an active enumerator
http://schemas.xmlsoap.org/ws/2004/09/enumeration/ReleaseResponse	Response to "Release"
http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerationEnd	Notifies that an enumerator has terminated (not required in WS-Management)
http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe	Models a subscription to an event source
http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscribeResponse	Response to "Subscribe"
http://schemas.xmlsoap.org/ws/2004/08/eventing/Renew	Renews a subscription prior to its expiration
http://schemas.xmlsoap.org/ws/2004/08/eventing/RenewResponse	Response to "Renew"
http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatus	Requests the status of a subscription
http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatusResponse	Response to "GetStatus"
http://schemas.xmlsoap.org/ws/2004/08/eventing/Unsubscribe	Removes an active subscription
http://schemas.xmlsoap.org/ws/2004/08/eventing/UnsubscribeResponse	Response to "Unsubscribe"
http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscriptionEnd	Delivers a message to indicate that a subscription has terminated
http://schemas.dmtf.org/wbem/wsman/1/wsman/Events	Delivers batched events based on a subscription
http://schemas.dmtf.org/wbem/wsman/1/wsman/Heartbeat	A pseudo-event that models a heartbeat of an active subscription; delivered when no real events are available, but used to indicate that the event subscription and delivery mechanism is still active
http://schemas.dmtf.org/wbem/wsman/1/wsman/DroppedEvents	A pseudo-event that indicates that the real event was dropped
http://schemas.dmtf.org/wbem/wsman/1/wsman/Ack	Used by event subscribers to acknowledge receipt of events; allows event streams to be strictly sequenced
http://schemas.dmtf.org/wbem/wsman/1/wsman/Event	Used for a singleton event that does not define its own action

1677 **R5.4.6.5-4**: A custom action may be supported if the operation is a custom method whose
1678 semantic meaning is not present in the table.

1679 **R5.4.6.5-5**: All notifications shall contain a unique action URI that identifies the type of the event
1680 delivery. For singleton notifications with only one event per message (the delivery mode
1681 http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push), the wsa:Action URI

© ISO/IEC 2013 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 17963:2013

defines the event type. For other delivery modes, the Action varies, as described in clause 10.2.7 of this specification.

5.4.6.6 wsa:From

The wsa:From header can be used in any messages, responses, or events to indicate the source. When the same connection is used for both request and reply, this header provides no useful information, but can be useful in cases where the response arrives on a different connection.

R5.4.6.6-1: A conformant service may include a wsa:From address in the message. A conformant service should process any incoming message that has a wsa:From element.

R5.4.6.6-2: A conformant service should not fault any message with a wsa:From element, regardless of whether the mustUnderstand attribute is included.

NOTE: Processing the wsa:From header is trivial because it has no effect on the meaning of the message. The *From* address is primarily for auditing and logging purposes.

6 WS-Management Control Headers

WS-Management defines several SOAP headers that can be used with any operation.

6.1 wsman:OperationTimeout

Most management operations are time-critical due to quality-of-service constraints and obligations. If operations cannot be completed in a specified time, the service returns a fault so that a client can comply with its obligations. The following header value can be supplied with any WS-Management message to indicate that the client expects a response or a fault within the specified time:

```
(1) <wsman:OperationTimeout> xs:duration </wsman:OperationTimeout>
```

R6.1-1: All request messages may contain a wsman:OperationTimeout header element that indicates the maximum amount of time the client is willing to wait for the service to issue a response. The service should interpret the timeout countdown as beginning from the point the message is processed until a response is generated.

R6.1-2: The service should *immediately* issue a wsman:TimedOut fault if the countdown time is exceeded and the operation is not yet complete. If the OperationTimeout value is not valid, a wsa:InvalidMessageInformationHeader fault should be returned.

R6.1-3: If the service does not support user-defined timeouts, a wsman:UnsupportedFeature fault should be returned with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/OperationTimeout

R6.1-4: If the wsman:OperationTimeout element is omitted, the service may interpret this omission as an instruction to block indefinitely until a response is available, or it may impose a default timeout.

These rules do not preclude services from supporting infinite or very long timeouts. Because network connections seldom block indefinitely with no traffic occurring, some type of transport timeout is likely. Also the countdown is initiated from the time the message is received, so network latency is not included. If a client needs to discover the range of valid timeouts or defaults, metadata can be retrieved, but the format of such metadata is beyond the scope of this specification.

If the timeout occurs in such a manner that the service has already performed some of the work associated with the request, the service state reaches an anomalous condition. This specification does not attempt to address behavior in this situation. Clearly, services can attempt to undo the

© ISO/IEC 2013 – All rights reserved

Full document details: BS ISO/IEC 17963:2013