



Devices Profile for Web Services Version 1.1

Committee Specification 01

14 May 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/ws-dd/dpws/1.1/cs-01/wsdd-dpws-1.1-spec-cs-01.html>
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cs-01/wsdd-dpws-1.1-spec-cs-01.docx> (Authoritative Format)
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cs-01/wsdd-dpws-1.1-spec-cs-01.pdf>

Previous Version:

<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-04/wsdd-dpws-1.1-spec-cd-04.html>
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-04/wsdd-dpws-1.1-spec-cd-04.docx>
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-04/wsdd-dpws-1.1-spec-cd-04.pdf>

Latest Version:

<http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.html>
<http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.docx>
<http://docs.oasis-open.org/ws-dd/dpws/wsdd-dpws-1.1-spec.pdf>

Technical Committee:

[OASIS Web Services Discovery and Web Services Devices Profile \(WS-DD\) TC](#)

Chair(s):

Toby Nixon (Microsoft Corporation)
Alain Regnier (Ricoh Company Limited)

Editor(s):

Dan Driscoll (Microsoft Corporation)
Antoine Mensch

Declared XML Namespace(s):

<http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01>

Abstract:

This profile defines a minimal set of implementation constraints to enable secure Web service messaging, discovery, description, and eventing on resource-constrained endpoints.

Status:

This document was last revised or approved by the OASIS Web Services Discovery and Web Services Devices Profile (WS-DD) TC on the above date. The level of approval is also listed above. Check the “Latest Version” or “Latest Approved Version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at <http://www.oasis-open.org/committees/ws-dd/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/ws-dd/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/ws-dd/>.

Notices

Copyright © OASIS® 2009. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction	5
1.1	Requirements	5
1.2	Terminology	5
1.3	XML Namespaces	7
1.4	XSD File	7
1.5	Normative References	7
1.6	Non-Normative References	9
2	Messaging	10
2.1	URI	10
2.2	UDP	10
2.3	HTTP	10
2.4	SOAP Envelope	11
2.5	WS-Addressing	11
2.6	Attachments	12
3	Discovery	13
4	Description	15
4.1	Characteristics	15
4.2	Hosting	18
4.3	WSDL	21
4.4	WS-Policy	23
5	Eventing	25
5.1	Subscription	25
5.2	Subscription Duration and Renewal	27
6	Security	28
6.1	Terminology	28
6.2	Model	28
6.3	Endpoint Reference and xAddr	29
6.4	Credentials	29
6.5	Discovery	30
6.6	Secure Channel	30
6.7	Authentication	32
6.8	Integrity	33
6.9	Confidentiality	33
7	Conformance	34
Appendix A.	Acknowledgements	35
Appendix B.	Constants	37
Appendix C.	Declaring Discovery Types in WSDL	38
Appendix D.	Example x.509.v3 Certificate	39
Appendix E.	Revision History	40

1 Introduction

The Web services architecture includes a suite of specifications that define rich functions and that may be composed to meet varied service requirements. To promote both interoperability between resource-constrained Web service implementations and interoperability with more flexible client implementations, this profile identifies a core set of Web service specifications in the following areas:

- Sending secure messages to and from a Web service
- Dynamically discovering a Web service
- Describing a Web service
- Subscribing to, and receiving events from, a Web service

In each of these areas of scope, this profile defines minimal implementation requirements for compliant Web service implementations.

1.1 Requirements

This profile intends to meet the following requirements:

- Identify a minimal set of Web service specifications needed to enable secure messaging, dynamic discovery, description, and eventing.
- Constrain Web services protocols and formats so Web services can be implemented on peripheral-class and consumer electronics-class hardware.
- Define minimum requirements for compliance without constraining richer implementations.

1.2 Terminology

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

1.2.1 Notational Conventions

This specification uses the following syntax to define normative outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of literal values.
- Characters are appended to elements and attributes to indicate cardinality:
 - "?" (0 or 1)
 - "*" (0 or more)
 - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "(" and ")" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- The characters "[" and "]" are used to call out references and property names.
- Ellipses (i.e., "...") indicate points of extensibility. Additional children and/or attributes MAY be added at the indicated extension points but MUST NOT contradict the semantics of the parent and/or owner, respectively. By default, if a receiver does not recognize an extension, the receiver SHOULD ignore the extension; exceptions to this processing rule, if any, are clearly indicated below.
- XML namespace prefixes (see [Table 1](#)) are used to indicate the namespace of the element being defined.

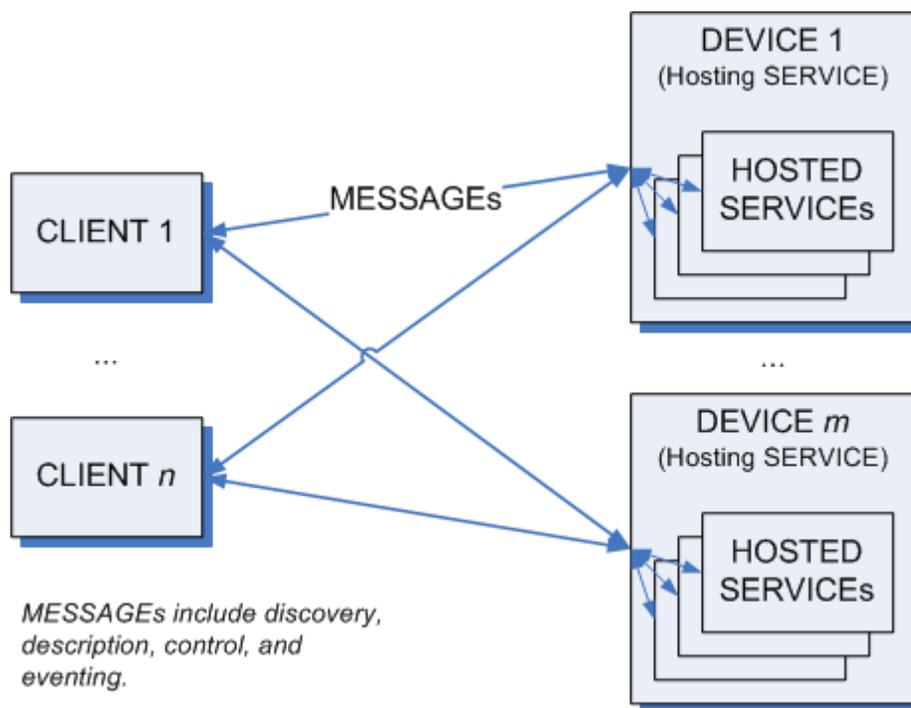
42 This specification uses the **[action]** and Fault properties [WS-Addressing] to define faults.

43 Normative statements in this profile are called out explicitly as follows:

44 *Rnnn: Normative statement text goes here.*

45 where "nnnn" is replaced by the statement number. Each statement contains exactly one requirement
46 level keyword (e.g., "MUST") and one conformance target keyword (e.g., "MESSAGE").

47 1.2.2 Terms and Definitions



48
49 **Figure 1: Arrangement of clients and devices**

50 MESSAGE

51 Protocol elements that are exchanged, usually over a network, to affect a Web service. Always
52 includes a SOAP ENVELOPE. Typically also includes transport framing information such as
53 HTTP headers, TCP headers, and IP headers.

54 SOAP ENVELOPE

55 An XML Infoset that consists of a document information item [XML Infoset] with exactly one
56 member in its [children] property, which MUST be the SOAP Envelope [SOAP 1.2] element
57 information item.

58 MIME SOAP ENVELOPE

59 A SOAP ENVELOPE serialized using MIME Multipart Serialization [MTOM].

60 TEXT SOAP ENVELOPE

61 A SOAP ENVELOPE serialized as application/soap+xml.

62 CLIENT

63 A network endpoint that sends MESSAGES to and/or receives MESSAGES from a SERVICE.

64 SERVICE

65 A software system that exposes its capabilities by receiving and/or sending MESSAGES on one
66 or several network endpoints.

67 DEVICE

68 A distinguished type of SERVICE that hosts other SERVICES and sends and/or receives one or
69 more specific types of MESSAGES.

70 HOSTED SERVICE

71 A distinguished type of SERVICE that is hosted by another SERVICE. The lifetime of the
72 HOSTED SERVICE is a subset of the lifetime of its host. The HOSTED SERVICE is visible (not
73 encapsulated) and is addressed separately from its host. Each HOSTED SERVICE has exactly
74 one host. (The relationship is not transitive.)

75 SENDER

76 A CLIENT or SERVICE that sends a MESSAGE.

77 RECEIVER

78 A CLIENT or SERVICE that receives a MESSAGE.

79 1.3 XML Namespaces

80 The XML namespace URI that MUST be used by implementations of this specification is:

81 `http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01`

82 Table 1 lists XML namespaces that are used in this specification. The choice of any namespace prefix is
83 arbitrary and not semantically significant.

84 **Table 1: Prefixes and XML namespaces used in this specification.**

Prefix	XML Namespace	Specification(s)
soap	http://www.w3.org/2003/05/soap-envelope	[SOAP 1.2]
wsa	http://www.w3.org/2005/08/addressing	[WS-Addressing]
wsd	http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01	[WS-Discovery]
dpws	http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01	This profile
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL 1.1]
wse	http://schemas.xmlsoap.org/ws/2004/08/eventing	[WS-Eventing]
wsp	http://www.w3.org/ns/ws-policy	[WS-Policy, WS-PolicyAttachment]
wsx	http://schemas.xmlsoap.org/ws/2004/09/mex	[WS-MetadataExchange]

85 1.4 XSD File

86 Dereferencing the XML namespace defined in Section 0

87 XML Namespaces will produce the Resource Directory Description Language (RDDL) [RDDL] document
88 that describes this namespace, including the XML Schema [XML Schema Part 1, 2] declarations
89 associated with this specification.

90 1.5 Normative References

91 [RFC 2119]

92 S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
93 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

94 [AES/TLS]

95 P.Chown, *Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)*,
96 <http://www.ietf.org/rfc/rfc3268.txt>, IETF RFC 3268, June 2004.

97 **[BP 1.1, Section 4]**
98 K. Ballinger, et al, *Basic Profile Version 1.1, Section 4: Service Description*, <http://www.ws->
99 [i.org/Profiles/BasicProfile-1.1-2004-08-24.html#description](http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#description), August 2004.

100 **[HTTP/1.1]**
101 R.Fielding, et al, *Hypertext Transfer Protocol -- HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>, IETF
102 RFC 2616, June 1999.

103 **[HTTP Authentication]**
104 J. Franks, et al, *HTTP Authentication: Basic and Digest Access Authentication*,
105 <http://www.ietf.org/rfc/rfc2617.txt>, IETF RFC 2617, June 1999.

106 **[MIME]**
107 N. Freed, et al, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet*
108 *Message Bodies*, <http://www.ietf.org/rfc/rfc2045.txt>, IETF RFC 2045, November 1996.

109 **[MTOM]**
110 N. Mendelsohn, et al, *SOAP Message Transmission Optimization Mechanism*,
111 <http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/>, January 2005.

112 **[RDDL]**
113 Jonathan Borden, et al, *Resource Directory Description Language (RDDL) 2.0*,
114 <http://www.openhealth.org/RDDL/20040118/rddl-20040118.html>, 18 January 2004.

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

118 **[SHA]**
119 *Secure Hash Standard*, http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf, October
120 2008.

121 **[SOAP 1.2, Part 1]**
122 M. Gudgin, et al, *SOAP Version 1.2 Part 1: Messaging Framework*,
123 <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>, April 2007.

124 **[SOAP 1.2, Part 2]**
125 M. Gudgin, et al, *SOAP Version 1.2 Part 2: Adjuncts, Section 7: SOAP HTTP Binding*,
126 <http://www.w3.org/TR/2007/REC-soap12-part2-20070427/#soapinhttp>, April 2007.

127 **[SOAP-over-UDP]**
128 OASIS Public Review Draft 01, *SOAP-over-UDP*, [http://docs.oasis-open.org/ws-](http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/pr-01/wsdd-soapoverudp-1.1-spec-pr-01.docx)
129 [dd/soapoverudp/1.1/pr-01/wsdd-soapoverudp-1.1-spec-pr-01.docx](http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/pr-01/wsdd-soapoverudp-1.1-spec-pr-01.docx), 30 January 2009.

130 **[TLS]**
131 T. Dierks, et al, *The TLS Protocol, Version 1.0*, <http://www.ietf.org/rfc/rfc2246.txt>, IETF RFC 2246,
132 January 1999.

133 **[WS-Addressing]**
134 W3C Recommendation, *Web Services Addressing 1.0 - Core*, [http://www.w3.org/TR/2006/REC-](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509)
135 [ws-addr-core-20060509](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509), 9 May 2006.

136 **[WS-Addressing SOAP Binding]**
137 W3C Recommendation, *Web Services Addressing 1.0 - SOAP Binding*,
138 <http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509>, 9 May 2006.

139 **[WS-Discovery]**
140 OASIS Public Review Draft 01, *Web Services Dynamic Discovery (WS-Discovery)*,
141 <http://docs.oasis-open.org/ws-dd/discovery/1.1/pr-01/wsdd-discovery-1.1-spec-pr-01.docx>, 30
142 January 2009.

143 **[WSDL 1.1]**
144 E. Christensen, et al, *Web Services Description Language (WSDL) 1.1*,
145 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, March 2001.

146 **[WSDL Binding for SOAP 1.2]**

- 147 K. Ballinger, et al, *WSDL 1.1 Binding Extension for SOAP 1.2*,
148 <http://www.w3.org/Submission/2006/SUBM-wsdl11soap12-20060405/>, 5 April 2006.
- 149 **[WS-Eventing]**
- 150 D. Box, et al, *Web Services Eventing (WS-Eventing)*, [http://www.w3.org/Submission/2006/SUBM-](http://www.w3.org/Submission/2006/SUBM-WS-Eventing-20060315/)
151 [WS-Eventing-20060315/](http://www.w3.org/Submission/2006/SUBM-WS-Eventing-20060315/), 15 March 2006.
- 152 **[WS-MetadataExchange]**
- 153 K. Ballinger, et al, *Web Services Metadata Exchange 1.1 (WS-MetadataExchange)*,
154 <http://www.w3.org/Submission/2008/SUBM-WS-MetadataExchange-20080813/>, 13 August 2008.
- 155 **[WS-Policy]**
- 156 W3C Recommendation, *Web Services Policy 1.5 - Framework*, [http://www.w3.org/TR/2007/REC-](http://www.w3.org/TR/2007/REC-ws-policy-20070904/)
157 [ws-policy-20070904/](http://www.w3.org/TR/2007/REC-ws-policy-20070904/), 4 September 2007.
- 158 **[WS-PolicyAttachment]**
- 159 W3C Recommendation, *Web Services Policy 1.5 - Attachment*, [http://www.w3.org/TR/2007/REC-](http://www.w3.org/TR/2007/REC-ws-policy-attach-20070904/)
160 [ws-policy-attach-20070904/](http://www.w3.org/TR/2007/REC-ws-policy-attach-20070904/), 4 September 2007.
- 161 **[WS-Transfer]**
- 162 J. Alexander, et al, *Web Service Transfer (WS-Transfer)*,
163 <http://www.w3.org/Submission/2006/SUBM-WS-Transfer-20060927/>, 27 September 2006.
- 164 **[X.509.v3]**
- 165 *ITU-T X.509.v3 Information technology - Open Systems Interconnection - The Directory: Public-*
166 *key and attribute certificate frameworks (ISO/IEC/ITU 9594-8)*
- 167 **[XML Schema, Part 1]**
- 168 W3C Recommendation, *XML Schema Part 1: Structures Second Edition*,
169 <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>, 28 October 2004.
- 170 **[XML Schema, Part 2]**
- 171 W3C Recommendation, *XML Schema Part 2: Datatypes Second Edition*,
172 <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>, 28 October 2004.
- 173 **1.6 Non-Normative References**
- 174 **[IPv6 Autoconfig]**
- 175 S. Thomson, et al, *IPv6 Stateless Address Autoconfiguration*, <http://www.ietf.org/rfc/2462.txt>,
176 IETF RFC 2462, December 1998.
- 177 **[DHCP]**
- 178 R. Droms, et al, *Dynamic Host Configuration Protocol*, <http://www.ietf.org/rfc/2131.txt>, IETF RFC
179 2131, March 1997.
- 180 **[XML Infoset]**
- 181 J. Cowan, et al, *XML Information Set (Second Edition)*, [http://www.w3.org/TR/2004/REC-xml-](http://www.w3.org/TR/2004/REC-xml-infoset/20040204/)
182 [infoset/20040204/](http://www.w3.org/TR/2004/REC-xml-infoset/20040204/), February 2004.
- 183 **[WS-Security]**
- 184 OASIS Standard Specification, *Web Services Security: SOAP Message Security 1.1 (WS-*
185 *Security 2004)*, [http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-](http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf)
186 [SOAPMessageSecurity.pdf](http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf), 1 February 2006.

187 2 Messaging

188 The scope of this section is the following set of Web services specifications. All of the requirements in
189 these specifications are included by reference except where superseded by normative statements herein:

- 190 • [SOAP 1.2, Part 1]
- 191 • [SOAP 1.2, Part 2]
- 192 • [SOAP-over-UDP]
- 193 • [HTTP/1.1]
- 194 • [WS-Addressing]
- 195 • [RFC 4122]
- 196 • [MTOM]

197 It is assumed that a DEVICE has obtained valid IPv4 and/or IPv6 addresses that do not conflict with other
198 addresses on the network. Mechanisms for obtaining IP addresses are out of the scope of this profile. For
199 more information, see [DHCP] and [IPv6 Autoconfig].

200 2.1 URI

201 *R0025: A SERVICE MAY fail to process any URI with more than MAX_URI_SIZE octets.*

202 *R0027: A SERVICE SHOULD NOT generate a URI with more than MAX_URI_SIZE octets.*

203 The constant MAX_URI_SIZE is defined in [Appendix B -- Constants](#).

204 2.2 UDP

205 *R0029: A SERVICE SHOULD NOT send a SOAP ENVELOPE that has more octets than the MTU over*
206 *UDP.*

207 To improve reliability, a SERVICE should minimize the size of SOAP ENVELOPEs sent over UDP.
208 However, some SOAP ENVELOPEs are larger than an MTU; for example, a signed Hello SOAP
209 ENVELOPE. If a SOAP ENVELOPE is larger than an MTU, the underlying IP network stacks fragment
210 and reassemble the UDP packet.

211 *R5018: A SERVICE MAY reject a SOAP ENVELOPE received over UDP that has more than*
212 *MAX_UDP_ENVELOPE_SIZE octets.*

213 *R5019: A CLIENT MAY reject a SOAP ENVELOPE received over UDP that has more than*
214 *MAX_UDP_ENVELOPE_SIZE octets.*

215 Unlike TCP or HTTP messages, UDP datagrams are received in one chunk, which may lead to excessive
216 resource requirements when receiving large datagrams on small embedded systems. The constant
217 MAX_UDP_ENVELOPE_SIZE is defined in [Appendix B -- Constants](#).

218 2.3 HTTP

219 *R0001: A SERVICE MUST support transfer-coding = "chunked".*

220 *R0012: A SERVICE MUST at least support the SOAP HTTP Binding.*

221 *R5000: A CLIENT MUST at least support the SOAP HTTP Binding.*

222 *R0013: A SERVICE MUST at least implement the Responding SOAP Node of the SOAP Request-*
223 *Response Message Exchange Pattern (<http://www.w3.org/2003/05/soap/mep/request-response/>).*

224 R0014: A SERVICE MAY choose not to implement the Responding SOAP Node of the SOAP Response
225 Message Exchange Pattern (<http://www.w3.org/2003/05/soap/mep/soap-response/>).

226 R0015: A SERVICE MAY choose not to support the SOAP Web Method Feature.

227 R0014 and R0015 relax requirements in [\[SOAP 1.2\]](#).

228 R0030: A SERVICE MUST at least implement the Responding SOAP Node of an HTTP one-way
229 Message Exchange Pattern where the SOAP ENVELOPE is carried in the HTTP Request and
230 the HTTP Response has a Status Code of 202 Accepted and an empty Entity Body (no SOAP
231 ENVELOPE).

232 R0017: A SERVICE MUST at least support Request Message SOAP ENVELOPEs and one-way SOAP
233 ENVELOPEs that are delivered using HTTP POST.

234 2.4 SOAP Envelope

235 R0034: A SERVICE MUST at least receive and send SOAP 1.2 [\[SOAP 1.2\]](#) SOAP ENVELOPEs.

236 R0003: A SERVICE MAY reject a TEXT SOAP ENVELOPE with more than MAX_ENVELOPE_SIZE
237 octets.

238 R0026: A SERVICE SHOULD NOT send a TEXT SOAP ENVELOPE with more than
239 MAX_ENVELOPE_SIZE octets.

240 Large SOAP ENVELOPEs are expected to be serialized using attachments.

241 R5001: A SERVICE MUST at least support SOAP ENVELOPEs with UTF-8 encoding.

242 R5002: A SERVICE MAY choose not to accept SOAP ENVELOPEs with UTF-16 encoding.

243 2.5 WS-Addressing

244 R5005: A SERVICE MUST at least support WS-Addressing 1.0 [\[WS-Addressing\]](#).

245 R5006: A SERVICE MAY reject messages using other versions of WS-Addressing.

246 Some underlying specifications (e.g., WS-Transfer [\[WS-Transfer\]](#)) explicitly allow other versions of WS-
247 Addressing. DPWS applications should rely solely on WS-Addressing 1.0.

248 R0004: A DEVICE SHOULD use a urn:uuid scheme IRI as the [address] property of its Endpoint
249 Reference.

250 R0005: A DEVICE MUST use a stable, globally unique identifier that is constant across re-initializations of
251 the device, and constant across network interfaces and IPv4/v6 addresses as the [address]
252 property of its Endpoint Reference.

253 R0006: A DEVICE MUST persist the [address] property of its Endpoint Reference across re-initialization
254 and changes in the metadata of the DEVICE and any SERVICES it hosts.

255 Because the [address] property of an Endpoint Reference [\[WS-Addressing\]](#) is a SOAP-layer address,
256 there is no requirement to use anything other than a UUID for the [address] property.

257 R0042: A HOSTED SERVICE SHOULD use an HTTP transport address as the [address] property of its
258 Endpoint References.

259 Use of other possible values of [address] by a HOSTED SERVICE is out of scope of this profile.

260 R0031: A SERVICE MUST NOT generate a `wsa:InvalidAddressingHeader` SOAP Fault [\[WS-Addressing](#)
261 [SOAP Binding\]](#) if the [address] of the [reply endpoint] of an HTTP Request Message SOAP
262 ENVELOPE is "<http://www.w3.org/2005/08/addressing/anonymous>".

263 R0041: If an HTTP Request Message SOAP ENVELOPE generates a SOAP Fault, a SERVICE MAY
264 discard the SOAP Fault if the [address] of the [fault endpoint] of the HTTP Request Message is
265 not "<http://www.w3.org/2005/08/addressing/anonymous>".

266 R0031 and R0041 ensure that messages with non-anonymous address in both the [reply endpoint] and
267 the [fault endpoint] do not result in a fault being sent.

268 The SOAP HTTP Binding requires the Response Message SOAP ENVELOPE to be transmitted as the
269 HTTP Response of the corresponding Request Message SOAP ENVELOPE.

270 *R0019: A SERVICE MUST include a Message Information Header representing a [relationship] property*
271 *of type wsa:Reply in each Response Message SOAP ENVELOPE the service generates.*

272 Per WS-Addressing [WS-Addressing], a response SOAP ENVELOPE must include a wsa:RelatesTo
273 SOAP ENVELOPE header block. Since "http://www.w3.org/2005/08/addressing/reply" is the default value
274 for the [relationship] property, the RelationshipType attribute should be omitted from the wsa:RelatesTo
275 SOAP ENVELOPE header block.

276 *R0040: A SERVICE MUST include a Message Information Header representing a [relationship] property*
277 *of "http://www.w3.org/2005/08/addressing/reply" in each SOAP Fault SOAP ENVELOPE the*
278 *service generates.*

279 2.6 Attachments

280 *R0022: If a SERVICE supports attachments, the SERVICE MUST support the HTTP Transmission*
281 *Optimization Feature.*

282 The HTTP Transmission Optimization Feature implies support for the Optimized MIME Multipart
283 Serialization and Abstract Transmission Optimization features.

284 *R0036: A SERVICE MAY reject a MIME SOAP ENVELOPE if the Content-Transfer-Encoding header field*
285 *mechanism of any MIME part is not "binary".*

286 *R0037: A SERVICE MUST NOT send a MIME SOAP ENVELOPE unless the Content-Transfer-Encoding*
287 *header field mechanism of every MIME part is "binary".*

288 Even for the SOAP Envelope, the "binary" Content-Transfer-Encoding mechanism is more appropriate
289 than the "8bit" mechanism which is suitable only for data that may be represented as relatively short lines
290 of at most 998 octets [MIME].

291 While DPWS-compliant services are required to support binary encoded MIME parts at a minimum,
292 R0036 allows for them to support others (non-DPWS compliant clients) if they choose. While a service
293 might choose to support more than what is required in DPWS, a DPWS-compliant client cannot assume
294 that the service it is interacting with supports anything beyond binary MIME parts.

295 *R0038: A SERVICE MAY reject a MIME SOAP ENVELOPE if the root part is not the first body part in the*
296 *Multipart/Related entity.*

297 *R0039: A SERVICE MUST NOT send a MIME SOAP ENVELOPE unless root part is the first body part in*
298 *the Multipart/Related entity.*

299 Per MTOM, the root part of the MIME SOAP ENVELOPE contains an XML representation of the modified
300 SOAP Envelope, with additional parts that contain binary representations of each attachment. This root
301 part must be the first part so a RECEIVER does not have to buffer attachments.

3 Discovery

The scope of this section is the following set of Web services specifications. All of the requirements in these specifications are included by reference except where superseded by normative statements herein:

- [WS-Discovery]

If a CLIENT and a SERVICE are not on the same subnet, the CLIENT may not be able to discover the SERVICE. However, if a CLIENT has an Endpoint Reference and transport address for a SERVICE through some other means, the CLIENT and SERVICE should be able to communicate within the scope of this profile.

R1013: A DEVICE MUST be a compliant WS-Discovery [WS-Discovery] Target Service.

R1001: A HOSTED SERVICE SHOULD NOT be a Target Service.

If each SERVICE were to participate in WS-Discovery, the network traffic generated by a relatively small number of DEVICES hosting a relatively small number of HOSTED SERVICES could overwhelm a bandwidth-limited network. Therefore, only DEVICES act as Target Services.

R1019: A DEVICE MUST at least support the "http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01/rfc3986" and "http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01/strcmp0" Scope matching rules.

R1020: If a DEVICE includes Types in a Hello, Probe Match, or Resolve Match SOAP ENVELOPE, it MUST include the dpws:Device Type.

Including the dpws:Device Type indicates a DEVICE supports the Devices Profile, and indicates a CLIENT may retrieve metadata about the DEVICE and its relationship to any HOSTED SERVICES using Get [WS-Transfer].

R1009: A DEVICE MUST at least support receiving Probe and Resolve SOAP ENVELOPEs and sending Hello and Bye SOAP ENVELOPEs over multicast UDP.

R1016: A DEVICE MUST at least support sending Probe Match and Resolve Match SOAP ENVELOPEs over unicast UDP.

R1018: A DEVICE MAY ignore a multicast UDP Probe or Resolve SOAP ENVELOPE if the [address] of the [reply endpoint] is not "http://www.w3.org/2005/08/addressing/anonymous".

WS-Discovery acknowledges that a CLIENT may include reply information in UDP Probe and Resolve SOAP ENVELOPEs to specify a transport other than SOAP over UDP. However, to establish a baseline for interoperability, DEVICES are required only to support UDP responses.

R1015: A DEVICE MUST support receiving a Probe SOAP ENVELOPE as an HTTP Request at any HTTP transport address where the DEVICE endpoint is available.

R5021: A DEVICE MAY reject a unicast Probe SOAP ENVELOPE received as an HTTP Request if the [address] property of the [destination] is not "urn:docs-oasis-open:ws-dd:ns:discovery:2009:01".

To support the scenario where a DEVICE has a known HTTP transport address, a CLIENT may send an ad-hoc Probe over HTTP to that address and expect to receive a ProbeMatches response, using the same message pattern as defined by the ProbeOp operation of the DiscoveryProxy portType in [WS-Discovery]. This requirement does not imply that the DEVICE must perform as a Discovery Proxy.

How the client obtains the DEVICE HTTP address is not defined in this specification, and this HTTP address does not necessarily relate to HOSTED SERVICE addresses.

A DEVICE MAY also listen for Directed Probes at http://<host address>:3702/.

R1021: If a DEVICE matches a Probe SOAP ENVELOPE received as an HTTP Request, it MUST send a Probe Matches SOAP ENVELOPE response containing a Probe Match section representing the DEVICE.

346 *R1022: If a DEVICE does not match a Probe SOAP ENVELOPE received as an HTTP Request, it MUST*
347 *send a Probe Matches SOAP ENVELOPE response with no Probe Match sections.*

348 *R5022: If a DEVICE includes a Probe Match section as an HTTP Response as described in [R1021](#), it*
349 *MUST include all of its Types and Scopes in the Probe Match section.*

350 DEVICES MAY omit their Types and Scopes in their UDP WS-Discovery messages to reduce message
351 size and prevent fragmentation. However, they are obligated to return all Types and Scopes in their
352 HTTP ProbeMatches messages as increased risk of packet loss due to fragmentation is not a
353 consideration.

354 4 Description

355 The scope of this section is the following set of Web services specifications. All of the requirements in
356 these specifications are included by reference except where superseded by normative statements herein:

- 357 • [\[XML Schema Part 1, Part 2\]](#)
- 358 • [\[WSDL 1.1\]](#)
- 359 • [\[BP 1.1, Section 4\]](#)
- 360 • [\[WSDL Binding for SOAP 1.2\]](#)
- 361 • [\[WS-MetadataExchange\]](#)
- 362 • [\[WS-Policy\]](#)
- 363 • [\[WS-PolicyAttachment\]](#)
- 364 • [\[WS-Transfer\]](#)

365 A DEVICE acts primarily as a metadata resource for device-wide data, and for the HOSTED SERVICES
366 on the device. A CLIENT retrieves the XML representation of these characteristics by sending a WS-
367 Transfer Get SOAP ENVELOPE to the DEVICE. The resulting metadata contains characteristics of the
368 device and topology information relating the DEVICE to its HOSTED SERVICES. WS-Transfer Get is
369 used here because the device-wide metadata is the XML representation of the DEVICE.

370 CLIENTs may also retrieve metadata for individual HOSTED SERVICES by sending a WS-
371 MetadataExchange GetMetadata SOAP ENVELOPE to the HOSTED SERVICE. The resulting metadata
372 contains limited topology information about the HOSTED SERVICE, its hosting DEVICE, its WSDL, and
373 any additional sections specific to the type of service. GetMetadata is used here because the XML
374 representation of the HOSTED SERVICE (possibly accessible with WS-Transfer Get) is not defined.

375 Through WSDL, this description also conveys the MESSAGES a HOSTED SERVICE is capable of
376 receiving and sending. Through WS-Policy, description conveys the capabilities and requirements of a
377 HOSTED SERVICE, particularly the transports over which it may be reached and its security capabilities.

378 *R5007: A DEVICE MUST support receiving a WS-Transfer Get SOAP ENVELOPE using the HTTP*
379 *binding defined in this profile.*

380 *R2044: In a Get Response SOAP ENVELOPE, a DEVICE MUST include only a wsx:Metadata element in*
381 *the SOAP ENVELOPE Body.*

382 All metadata from the device should be contained in the wsx:Metadata element in the Get Response.

383 *R2045: A DEVICE MAY generate a wsa:ActionNotSupported SOAP Fault in response to a Put, Delete, or*
384 *Create SOAP ENVELOPE.*

385 A DEVICE is not required to support all of the operations defined in [\[WS-Transfer\]](#).

386 *R5008: A HOSTED SERVICE MUST support receiving a WS-MetadataExchange GetMetadata SOAP*
387 *ENVELOPE using the HTTP binding defined in this profile.*

388 4.1 Characteristics

389 To express DEVICE characteristics that are typically fixed across all DEVICES of the same model by their
390 manufacturer, this profile defines extensible ThisModel metadata as follows:

```
391 <dpws:ThisModel ...>  
392   <dpws:Manufacturer xml:lang="..."? >xs:string</dpws:Manufacturer>+  
393   <dpws:ManufacturerUrl>xs:anyURI</dpws:ManufacturerUrl?>  
394   <dpws:ModelName xml:lang="..."? >xs:string</dpws:ModelName>+  
395   <dpws:ModelNumber>xs:string</dpws:ModelNumber?>  
396   <dpws:ModelUrl>xs:anyURI</dpws:ModelUrl?>  
397   <dpws:PresentationUrl>xs:anyURI</dpws:PresentationUrl?>
```

398 ...
 399 </dpws:ThisModel>
 400 The following describes additional, normative constraints on the outline above:
 401 dpws:ThisModel/ dpws:Manufacturer
 402 Name of the manufacturer of the DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode
 403 characters, SHOULD be localized, and SHOULD be repeated for each supported locale.
 404 dpws:ThisModel/ dpws:ManufacturerUrl
 405 URL to a Web site for the manufacturer of the DEVICE. It MUST have fewer than
 406 MAX_URI_SIZE octets.
 407 dpws:ThisModel/ dpws:ModelName
 408 User-friendly name for this model of device chosen by the manufacturer. It MUST have fewer
 409 than MAX_FIELD_SIZE Unicode characters, SHOULD be localized, and SHOULD be repeated
 410 for each supported locale.
 411 dpws:ThisModel/ dpws:ModelNumber
 412 Model number for this model of DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode
 413 characters.
 414 dpws:ThisModel/ dpws:ModelUrl
 415 URL to a Web site for this model of DEVICE. It MUST have fewer than MAX_URI_SIZE octets.
 416 dpws:ThisModel/ dpws:PresentationUrl
 417 URL to a presentation resource for this DEVICE. It MAY be relative to the HTTP transport
 418 address over which metadata was retrieved, and MUST have fewer than MAX_URI_SIZE octets.
 419 If PresentationUrl is specified, the DEVICE MAY provide the resource in multiple formats, but
 420 MUST at least provide an HTML page. CLIENTs and DEVICEs MAY use HTTP content
 421 negotiation [HTTP/1.1] to determine the format and content of the presentation resource.
 422 DEVICEs that use a relative URL MAY use HTTP Redirection 3xx codes [HTTP/1.1] to direct
 423 CLIENTs to a dedicated web server running on another port.

424 CORRECT:

```

425 <dpws:ThisModel
426    xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
427    <dpws:Manufacturer>ACME Manufacturing</dpws:Manufacturer>
428    <dpws:ModelName xml:lang="en-GB" >ColourBeam 9</dpws:ModelName>
429    <dpws:ModelName xml:lang="en-US" >ColorBeam 9</dpws:ModelName>
430 </dpws:ThisModel>
  
```

431 A Dialect [WS-MetadataExchange] equal to "http://docs.oasis-open.org/ws-
 432 dd/ns/dpws/2009/01/ThisModel" indicates an instance of the ThisModel metadata format.
 433 No Identifier [WS-MetadataExchange] is defined for instances of the ThisModel metadata format.

434 *R2038: A DEVICE MUST have one Metadata Section with Dialect equal to "http://docs.oasis-
 435 open.org/ws-dd/ns/dpws/2009/01/ThisModel" for its ThisModel metadata.*

436 *R2012: In any Get Response SOAP ENVELOPE, a DEVICE MUST include the Metadata Section with
 437 Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisModel".*

438 Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve the resource representation data
 439 for a DEVICE – which includes the ThisModel metadata for a DEVICE. A DEVICE MAY also provide other
 440 means for a CLIENT to retrieve its ThisModel metadata.

441 *R2001: If a DEVICE changes any of its ThisModel metadata, it MUST increment the Metadata Version
 442 exposed in Hello, Probe Match, and Resolve Match SOAP ENVELOPEs as
 443 wsd:MetadataVersion.*

444 Caching for the ThisModel metadata is controlled by the wsd:MetadataVersion construct [WS-Discovery].

445 To express DEVICE characteristics that typically vary from one DEVICE to another of the same kind, this
446 profile defines extensible ThisDevice metadata as follows:

```
447 <dpws:ThisDevice ...>
448   <dpws:FriendlyName xml:lang="..."? >xs:string</dpws:FriendlyName>+
449   <dpws:FirmwareVersion>xs:string</dpws:FirmwareVersion>?
450   <dpws:SerialNumber>xs:string</dpws:SerialNumber>?
451   ...
452 </dpws:ThisDevice>
```

453 The following describes additional, normative constraints on the outline above:

454 dpws:ThisDevice/dpws:FriendlyName

455 User-friendly name for this DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode
456 characters, SHOULD be localized, and SHOULD be repeated for each supported locale.

457 dpws:ThisDevice/dpws:FirmwareVersion

458 Firmware version for this DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode
459 characters.

460 dpws:ThisDevice/dpws:SerialNumber

461 Manufacturer-assigned serial number for this DEVICE. It MUST have fewer than
462 MAX_FIELD_SIZE Unicode characters.

463 CORRECT:

```
464 <dpws:ThisDevice
465   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
466   <dpws:FriendlyName xml:lang="en-GB" >
467     ACME ColourBeam Printer
468   </dpws:FriendlyName>
469   <dpws:FriendlyName xml:lang="en-US" >
470     ACME ColorBeam Printer
471   </dpws:FriendlyName>
472 </dpws:ThisDevice>
```

473 A Dialect [[WS-MetadataExchange](#)] equal to "http://docs.oasis-open.org/ws-
474 dd/ns/dpws/2009/01/ThisDevice" indicates an instance of the ThisDevice metadata format.

475 No Identifier [[WS-MetadataExchange](#)] is defined for instances of the ThisDevice metadata format.

476 *R2039: A DEVICE MUST have a Metadata Section with Dialect equal to "http://docs.oasis-open.org/ws-
477 dd/ns/dpws/2009/01/ThisDevice" for its ThisDevice metadata.*

478 *R2014: In any Get Response SOAP ENVELOPE, a DEVICE MUST include the Metadata Section with
479 Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisDevice".*

480 CORRECT:

```
481 <soap:Envelope
482   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
483   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
484   xmlns:wsm="http://schemas.xmlsoap.org/ws/2004/09/mex"
485   xmlns:wsa="http://www.w3.org/2005/08/addressing" >
486   <soap:Header>
487     <wsa:Action>
488       http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
489     </wsa:Action>
490     <wsa:RelatesTo>
491       urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
492     </wsa:RelatesTo>
493     <wsa:To>
494       http://www.w3.org/2005/08/addressing/anonymous
495     </wsa:To>
```

```

496 </soap:Header>
497 <soap:Body>
498   <wsx:Metadata>
499     <wsx:MetadataSection
500       Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisModel"
501       >
502         <dpws:ThisModel>
503           <dpws:Manufacturer>ACME Manufacturing</dpws:Manufacturer>
504           <dpws:ModelName xml:lang="en-GB" >
505             ColourBeam 9
506           </dpws:ModelName>
507           <dpws:ModelName xml:lang="en-US" >
508             ColorBeam 9
509           </dpws:ModelName>
510         </dpws:ThisModel>
511       </wsx:MetadataSection>
512     <wsx:MetadataSection
513       Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisDevice"
514       >
515       <dpws:ThisDevice>
516         <dpws:FriendlyName xml:lang="en-GB" >
517           ACME ColourBeam Printer
518         </dpws:FriendlyName>
519         <dpws:FriendlyName xml:lang="en-US" >
520           ACME ColorBeam Printer
521         </dpws:FriendlyName>
522       </dpws:ThisDevice>
523     </wsx:MetadataSection>
524
525     <!-- Other Metadata Sections omitted for brevity. -->
526
527   </wsx:Metadata>
528 </soap:Body>
529 </soap:Envelope>

```

530 Get [[WS-Transfer](#)] is the interoperable means for a CLIENT to retrieve the resource representation data
531 for a DEVICE – which includes the ThisDevice metadata for a DEVICE. A DEVICE MAY also provide
532 other means for a CLIENT to retrieve its ThisDevice metadata.

533 *R2002: If a DEVICE changes any of its ThisDevice metadata, it MUST increment the Metadata Version*
534 *exposed in Hello, Probe Match, and Resolve Match SOAP ENVELOPES as*
535 *wsd:MetadataVersion.*

536 Caching for the ThisDevice metadata is controlled by the wsd:MetadataVersion construct [[WS-Discovery](#)].

537 4.2 Hosting

538 To express the relationship between a HOSTED SERVICE and its hosting DEVICE, this profile defines
539 relationship metadata as follows:

```

540 <dpws:Relationship Type="xs:anyURI" ... >
541   (<dpws:Host>
542     <wsa:EndpointReference>endpoint-reference</wsa:EndpointReference>
543     <dpws:Types>list of xs:QName</dpws:Types>?
544     ...
545   </dpws:Host>)?
546   (<dpws:Hosted>
547     <wsa:EndpointReference>endpoint-reference</wsa:EndpointReference>+
548     <dpws:Types>list of xs:QName</dpws:Types>
549     <dpws:ServiceId>xs:anyURI</dpws:ServiceId>

```

```
550     ...
551     </dpws:Hosted>)*
552     ...
553 </dpws:Relationship>
```

554 The following describes additional, normative constraints on the outline above:

555 dpws:Relationship

556 This is a general mechanism for defining a relationship between two or more SERVICES.

557 dpws:Relationship/@Type

558 The type of the relationship. The nature of the relationship and the content of the
559 dpws:Relationship element are determined by this value. This value should be compared directly,
560 as a case-sensitive string, with no attempt to make a relative URI into an absolute URI, to
561 unescape, or to otherwise canonicalize it.

562 dpws:Relationship/@Type = "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host"

563 This is a specific, hosting relationship type to indicate the relationship between a HOSTED
564 SERVICE and its hosting DEVICE. This relationship type defines the following additional content:

565 dpws:Relationship/dpws:Host

566 This is a section describing a hosting DEVICE. At least one of ./dpws:Host or ./dpws:Hosted
567 MUST be included.

568 dpws:Relationship/dpws:Host/wsa:EndpointReference

569 Endpoint Reference for the host, which includes the stable identifier for the host which MUST be
570 persisted across re-initialization (see also R0005 and R0006). If ./dpws:Host is omitted, implied
571 value is the Endpoint Reference of the DEVICE that returned this metadata in a Get Response
572 SOAP ENVELOPE.

573 dpws:Relationship/dpws:Host/dpws:Types

574 Unordered set of Types implemented by the host. (See [WS-Discovery].) If omitted or ./dpws:Host
575 is omitted, no implied value.

576 dpws:Relationship/dpws:Hosted

577 This is a section describing a HOSTED SERVICE. . It MUST be included by a DEVICE for each
578 of its HOSTED SERVICES. It MUST be included by a HOSTED SERVICE for itself. For the
579 hosting relationship type, if a host has more than one HOSTED SERVICE, including one
580 relationship that lists all HOSTED SERVICES is equivalent to including multiple relationships that
581 each list some subset of the HOSTED SERVICES.

582 dpws:Relationship/dpws:Hosted/wsa:EndpointReference

583 Endpoint References for a HOSTED SERVICE.

584 dpws:Relationship/dpws:Hosted/dpws:Types

585 Unordered set of Types implemented by a HOSTED SERVICE. All implemented Types SHOULD
586 be included.

587 dpws:Relationship/dpws:Hosted/dpws:ServiceId

588 Identifier for a HOSTED SERVICE which MUST be persisted across re-initialization and MUST
589 NOT be shared across multiple Hosted elements. ServiceId MUST be unique within a DEVICE.

590 This value should be compared directly, as a case-sensitive string, with no attempt to make a
591 relative URI into an absolute URI, to unescape, or to otherwise canonicalize it.

592 CORRECT:

```
593 <dpws:Relationship
594   Type="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host"
595   xmlns:img="http://printer.example.org/imaging"
596   xmlns:wsa="http://www.w3.org/2005/08/addressing"
597   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
598 <dpws:Hosted>
```

```

599 <wsa:EndpointReference>
600   <wsa:Address>http://172.30.184.244/print</wsa:Address>
601 </wsa:EndpointReference>
602 <dpws:Types>
603   img:PrintBasicPortType img:PrintAdvancedPortType
604 </dpws:Types>
605 <dpws:ServiceId>
606   http://printer.example.org/imaging/PrintService
607 </dpws:ServiceId>
608 </dpws:Hosted>
609 </dpws:Relationship>

```

610 A Dialect [[WS-MetadataExchange](#)] equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship" indicates an instance of the Relationship metadata format.

612 No Identifier [[WS-MetadataExchange](#)] is defined for instances of the Relationship metadata format.

613 *R2040: If a DEVICE has any HOSTED SERVICES, it MUST have at least one Metadata Section with*
614 *Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship" for its*
615 *Relationship metadata.*

616 *R2029: In any Get Response SOAP ENVELOPE, a DEVICE MUST include any Metadata Section(s) with*
617 *Dialect equal to "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship".*

618 Get [[WS-Transfer](#)] is the interoperable means for a CLIENT to retrieve the resource representation data
619 for a DEVICE – which includes the relationship metadata for itself and HOSTED SERVICES.

620 *R5020: A HOSTED SERVICE MUST have one Metadata Section with http://docs.oasis-open.org/ws-*
621 *dd/ns/dpws/2009/01/Relationship".*

622 GetMetadata [[WS-MetadataExchange](#)] is the interoperable means for a CLIENT to retrieve metadata for
623 a HOSTED SERVICE – which includes the relationship metadata for itself and its hosting DEVICE.

624 A DEVICE or HOSTED SERVICE MAY provide other means for a CLIENT to retrieve its relationship
625 metadata.

626 CORRECT:

```

627 <soap:Envelope
628   xmlns:gen="http://example.org/general"
629   xmlns:img="http://printer.example.org/imaging"
630   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
631   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
632   xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
633   xmlns:wsa="http://www.w3.org/2005/08/addressing" >
634 <soap:Header>
635   <wsa:Action>
636     http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
637   </wsa:Action>
638   <wsa:RelatesTo>
639     urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
640   </wsa:RelatesTo>
641   <wsa:To>
642     http://www.w3.org/2005/08/addressing/anonymous
643   </wsa:To>
644 </soap:Header>
645 <soap:Body>
646   <wsx:Metadata>
647     <wsx:MetadataSection
648       Dialect
649       ="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship"
650     >
651   </dpws:Relationship

```

```

652     Type="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host" >
653     <dpws:Hosted>
654         <wsa:EndpointReference>
655             <wsa:Address>http://172.30.184.244/print</wsa:Address>
656         </wsa:EndpointReference>
657         <wsa:EndpointReference>
658             <wsa:Address>http://[fdaa:23]/print1</wsa:Address>
659         </wsa:EndpointReference>
660         <dpws:Types>
661             img:PrintBasicPortType img:PrintAdvancedPortType
662         </dpws:Types>
663         <dpws:ServiceId>
664             http://printer.example.org/imaging/PrintService
665         </dpws:ServiceId>
666     </dpws:Hosted>
667     <dpws:Hosted>
668         <wsa:EndpointReference>
669             <wsa:Address>http://172.30.184.244/scan</wsa:Address>
670         </wsa:EndpointReference>
671         <wsa:EndpointReference>
672             <wsa:Address>http://[fdaa:24]/scan</wsa:Address>
673         </wsa:EndpointReference>
674         <dpws:Types>img:ScanBasicPortType</dpws:Types>
675         <dpws:ServiceId>
676             http://printer.example.org/imaging/ScanService
677         </dpws:ServiceId>
678     </dpws:Hosted>
679 </dpws:Relationship>
680 </wsx:MetadataSection>
681
682 <!-- Other Metadata Sections omitted for brevity. -->
683
684 </wsx:Metadata>
685 </soap:Body>
686 </soap:Envelope>

```

687 *R2030: If a DEVICE changes any of its relationship metadata, it MUST increment the Metadata Version*
688 *exposed in Hello, Probe Match, and Resolve Match SOAP ENVELOPES as*
689 *wsd:MetadataVersion.*

690 Caching for relationship metadata is controlled by the wsd:MetadataVersion construct [[WS-Discovery](#)].

691 *R2042: A DEVICE MUST NOT change its relationship metadata based on temporary changes in the*
692 *network availability of the SERVICES described by the metadata.*

693 Relationship metadata is intended to model fairly static relationships and should not change if a SERVICE
694 becomes temporarily unavailable. As in the general case, any CLIENT attempting to contact such a
695 SERVICE will need to deal with an Endpoint Unavailable Fault [[WS-Addressing](#)], connection refusal, or
696 other network indication that the SERVICE is unavailable.

697 4.3 WSDL

698 *R2004: If a HOSTED SERVICE exposes Notifications, its portType MUST include Notification and/or*
699 *Solicit-Response Operations describing those Notifications.*

700 R2004 relaxes R2303 in [[BP 1.1, Section 4](#)].

701 *R2019: A HOSTED SERVICE MUST at least include a document-literal Binding for SOAP 1.2 over HTTP*
702 *for each portType in its WSDL.*

703 Because the document-literal SOAP Binding is more general than an rpc-literal SOAP Binding, there is no
704 requirement to use anything other than the document-literal Binding.

705 **R2028: A HOSTED SERVICE is not required to include any WSDL bindings for SOAP 1.1 in its WSDL.**

706 Since this profile brings SOAP 1.2 into scope, it is sufficient to bind to that version of SOAP. There is no
707 requirement to bind to other SOAP versions and thus R2028 updates R2401 in [BP 1.1, Section 4] to
708 SOAP 1.2.

709 Addressing information for a HOSTED SERVICE is included in relationship metadata. For the mandatory
710 SOAP 1.2 binding (R2019), there is no requirement to re-express this information in a WSDL Service and
711 Port, since the endpoint reference used in the relationship metadata refers to this binding by default. The
712 use of WSDL Services and Ports may still be necessary for other bindings not covered by this profile.

713 **R2023: If a HOSTED SERVICE receives a MESSAGE that is inconsistent with its WSDL description, the
714 HOSTED SERVICE SHOULD generate a SOAP Fault with a Code Value of "Sender", unless a
715 "MustUnderstand" or "VersionMismatch" Fault is generated.**

716 **R2024: If a HOSTED SERVICE receives a MESSAGE that is inconsistent with its WSDL description, the
717 HOSTED SERVICE MUST check for "VersionMismatch", "MustUnderstand", and "Sender" fault
718 conditions in that order.**

719 Statements R2023 and R2024 update R2724 and R2725 [BP 1.1, Section 4] to SOAP 1.2.

720 **R2031: A HOSTED SERVICE MUST have at least one Metadata Section with
721 Dialect="http://schemas.xmlsoap.org/wsdl/".**

722 For clarity, separation of levels of abstraction, and/or reuse of standardized components, WSDL may be
723 authored in a style that separates different elements of a Service Definition into separate documents
724 which may be imported or included as needed. Each separate document may be available at the URL in
725 the xs:include/@schemaLocation, xs:import/@schemaLocation, or wsdl:import/@location or may be
726 included in a separate XML Schema or WSDL Metadata Section.

727 GetMetadata [WS-MetadataExchange] is the interoperable means for a CLIENT to retrieve metadata for
728 a HOSTED SERVICE – which includes the WSDL for a HOSTED SERVICE. A HOSTED SERVICE MAY
729 provide other means for a CLIENT to retrieve its WSDL.

730 There is no requirement for a HOSTED SERVICE to store its WSDL and include it in-line in a Get
731 Response SOAP ENVELOPE. The WSDL may be stored at a different location, and the HOSTED
732 SERVICE may include a reference to it in a Get Response SOAP ENVELOPE.

733 CORRECT:

```
734 <soap:Envelope  
735   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"  
736   xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"  
737   xmlns:wsa="http://www.w3.org/2005/08/addressing" >  
738   <soap:Header>  
739     <wsa:Action>  
740       http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response  
741     </wsa:Action>  
742     <wsa:RelatesTo>  
743       urn:uuid:82204a83-52f6-475c-9708-174fa27659ec  
744     </wsa:RelatesTo>  
745     <wsa:To>  
746       http://www.w3.org/2005/08/addressing/anonymous  
747     </wsa:To>  
748   </soap:Header>  
749   <soap:Body>  
750     <wsx:Metadata>  
751       <wsx:MetadataSection  
752         Dialect="http://schemas.xmlsoap.org/wsdl" >  
753         <wsx:MetadataReference>  
754           <wsa:Address>http://172.30.184.244/print</wsa:Address>
```

```

755     <wsa:ReferenceParameters>
756         <x:Acme xmlns:x="urn:acme.com:webservices">
757             WSDL
758         </x:Acme>
759     </wsa:ReferenceParameters>
760 </wsx:MetadataReference>
761 </wsx:MetadataSection>
762
763 <!-- Other Metadata Sections omitted for brevity. -->
764
765 </wsx:Metadata>
766 </soap:Body>
767 </soap:Envelope>

```

768 4.4 WS-Policy

769 To indicate that a SERVICE is compliant with this profile, this profile defines the following WS-Policy [WS-
770 Policy] assertion:

```
771 <dpws:Profile wsp:Optional="true"? ... />
```

772 The following describes additional, normative constraints on the outline above:

773 dpws:Profile

774 Assertion indicating compliance with this profile is required. This assertion has Endpoint Policy
775 Subject [WS-PolicyAttachment]: a policy expression containing this assertion MAY be attached to
776 a wsdl:port, SHOULD be attached to a wsdl:binding, but MUST NOT be attached to a
777 wsdl:portType; the latter is prohibited because the assertion specifies a concrete behavior
778 whereas the wsdl:portType is an abstract construct.

779 dpws:Profile/@wsp:Optional="true"

780 Per WS-Policy [WS-Policy], this is compact notation for two policy alternatives, one with and one
781 without the assertion. The intuition is that the behavior indicated by the assertion is optional, or in
782 this case, that the SERVICE supports but does not require compliance with this profile.

783 CORRECT:

```

784 <wsp:Policy
785     xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
786     xmlns:wsp="http://www.w3.org/ns/ws-policy" >
787     <dpws:Profile />
788 </wsp:Policy>

```

789 **R2037: A SERVICE MUST include the dpws:Profile assertion in its policy.**

790 This assertion has Endpoint Policy Subject: a policy expression containing this assertion MAY be
791 attached to a wsdl:port, SHOULD be attached to a wsdl:binding, but MUST NOT be attached to a
792 wsdl:portType; the latter is prohibited because this assertion specifies concrete behavior whereas the
793 wsdl:portType is an abstract construct.

794 **R2041: If a SERVICE uses wsp:PolicyReference/@URI to attach a policy identified by an absolute IRI,**
795 **the SERVICE MUST have a Metadata Section with Dialect equal to "http://www.w3.org/ns/ws-**
796 **policy" and Identifier equal to that IRI.**

797 **R2025: If a SERVICE uses wsp:PolicyReference/@URI to attach a policy identified by an absolute IRI,**
798 **then in a Get Response SOAP ENVELOPE, the SERVICE MUST include the Metadata Section**
799 **with Dialect equal to "http://www.w3.org/ns/ws-policy" and Identifier equal to that IRI.**

800 **R2035: If a SERVICE uses wsp:PolicyReference/@URI to attach a policy identified by a relative IRI, the**
801 **SERVICE MUST embed that policy as a child of wsdl:definitions, and the policy MUST have a**
802 **@wsu:Id containing that IRI.**

803 **R2036: A SERVICE MUST NOT use @wsp:PolicyURIs to attach policy.**

804 Because all components in WSDL are extensible via elements [BP 1.1, Section 4], attachment using
805 wsp:PolicyReference/@URI is sufficient.

806 Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve attached policy.

807 CORRECT:

```
808 <soap:Envelope
809   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
810   xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
811   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
812   xmlns:wsp="http://www.w3.org/ns/ws-policy"
813   xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
814   xmlns:wsa="http://www.w3.org/2005/08/addressing" >
815 <soap:Header>
816   <wsa:Action>
817     http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
818   </wsa:Action>
819   <wsa:RelatesTo>
820     urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
821   </wsa:RelatesTo>
822   <wsa:To>
823     http://www.w3.org/2005/08/addressing/anonymous
824   </wsa:To>
825 </soap:Header>
826 <soap:Body>
827   <wsx:Metadata>
828     <wsx:MetadataSection
829       Dialect="http://schemas.xmlsoap.org/wSDL/" >
830       <wSDL:definitions
831         targetNamespace="http://acme.example.com/colorbeam"
832         xmlns:image="http://printer.example.org/imaging" >
833         <wsp:Policy wsu:Id="DpPolicy" >
834           <dpws:Profile />
835         </wsp:Policy>
836
837         <!-- Other WSDL components omitted for brevity. -->
838
839         <wSDL:binding name="PrintBinding" type="image:PrintPortType" >
840           <wsp:PolicyReference URI="#DpPolicy"
841             wSDL:required="true" />
842           <!-- Other WSDL components omitted for brevity. -->
843         </wSDL:binding>
844       </wSDL:definitions>
845     </wsx:MetadataSection>
846
847     <!-- Other Metadata Sections omitted for brevity. -->
848
849   </wsx:Metadata>
850 </soap:Body>
851 </soap:Envelope>
```

5 Eventing

853 The scope of this section is the following set of Web services specifications. All of the requirements in
854 these specifications are included by reference except where superseded by normative statements herein:

- 855 • [\[WS-Eventing\]](#)

5.1 Subscription

857 *R3009: A HOSTED SERVICE MUST at least support Push Delivery Mode indicated by*
858 *"http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push".*

859 The Push Delivery Mode [\[WS-Eventing\]](#) is the default Delivery Mode and indicates the Event Source
860 (HOSTED SERVICE) will push Notifications to the Event Sink (CLIENT).

861 *R3017: If a HOSTED SERVICE does not understand the [address] of the Notify To of a Subscribe SOAP*
862 *ENVELOPE, the HOSTED SERVICE MUST generate a wsa:DestinationUnreachable SOAP Fault*
863 *in place of a SubscribeResponse message.*

864 *R3018: If a HOSTED SERVICE does not understand the [address] of the End To of a Subscribe SOAP*
865 *ENVELOPE, the HOSTED SERVICE MUST generate a wsa:DestinationUnreachable SOAP Fault*
866 *in place of a SubscribeResponse message.*

867 R3017 and R3018 do not ensure that a HOSTED SERVICE can contact an event sink, but they do
868 provide a mechanism for the event source to fault on unsupported URI schemes or addresses it knows it
869 cannot contact.

870 *R5003: If a HOSTED SERVICE generates a wsa:DestinationUnreachable SOAP Fault under R3017 or*
871 *R3018, the SOAP Fault Detail MUST be the EndTo or NotifyTo Endpoint Reference Address that*
872 *the HOSTED SERVICE did not understand.*

873 [R5003](#) allows a client to distinguish between a SOAP Fault generated due to an unreachable [destination]
874 information header in the Subscribe message, and a SOAP Fault generated due to an unreachable
875 NotifyTo or EndTo address.

876 *R3019: If a HOSTED SERVICE cannot deliver a Notification SOAP ENVELOPE to an Event Sink, the*
877 *HOSTED SERVICE MAY terminate the corresponding Subscription.*

878 *R5004: If a HOSTED SERVICE terminates a subscription (per R3019), the HOSTED SERVICE SHOULD*
879 *send a Subscription End SOAP ENVELOPE with a Status of*
880 *"http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryFailure".*

5.1.1 Filtering

882 To enable subscribing to one or more Notifications exposed by a HOSTED SERVICE, this profile defines
883 a Filter Dialect designated "http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action".

- 884 • A Filter in this Dialect contains a white space-delimited list of URIs that indicate the [action]
885 property of desired Notifications.
- 886 • The content of a Filter in this Dialect is defined as xs:list/@itemType="xs:anyURI" [[XML Schema](#)
887 [Part 2](#)].
- 888 • A Filter in this Dialect evaluates to true for an Output Message of a Notification or Solicit-
889 Response operation if and only if a URI in the Filter matches the [action] property of the Message
890 using the "http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01/rfc3986" matching rule [[WS-](#)
891 [Discovery](#)].
- 892 • A Filter in this Dialect with no URIs specified will always evaluate to false for all messages.

893 The Action Dialect uses the RFC 3986 prefix matching rule so CLIENTs can subscribe to a related set of
894 Notifications by including the common prefix of the [action] property of those Notifications. Typically, the

895 Notifications within a WSDL portType [WSDL 1.1] will share a common [action] property prefix, and
 896 specifying that prefix with the Action Dialect will be a convenient means to subscribe to all Notifications
 897 defined by a portType.

898 *R3008: A HOSTED SERVICE MUST at least support Filtering by the Dialect "http://docs.oasis-*
 899 *open.org/ws-dd/ns/dpws/2009/01/Action".*

900 CORRECT:

```

901 <soap:Envelope
902   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
903   xmlns:wsa="http://www.w3.org/2005/08/addressing"
904   xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing" >
905   <soap:Header>
906     <wsa:Action>
907       http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
908     </wsa:Action>
909     <wsa:MessageID>
910       urn:uuid:314bea3b-03af-47a1-8284-f495497f1e33
911     </wsa:MessageID>
912     <wsa:ReplyTo>
913       <wsa:Address>
914         http://www.w3.org/2005/08/addressing/anonymous
915       </wsa:Address>
916     </wsa:ReplyTo>
917     <wsa:To>http://172.30.184.244/print</wsa:To>
918   </soap:Header>
919   <soap:Body>
920     <wse:Subscribe>
921       <wse:Delivery>
922         <wse:NotifyTo>
923           <wsa:Address>
924             urn:uuid:3726983d-02de-4d41-8207-d028ae92ce3d
925           </wsa:Address>
926         </wse:NotifyTo>
927       </wse:Delivery>
928       <wse:Expires>PT10M</wse:Expires>
929       <wse:Filter
930 Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action"
931       >
932 http://printer.example.org/imaging/PrintBasicPortType/JobEndState
933 http://printer.example.org/imaging/PrintBasicPortType/PrinterState
934     </wse:Filter>
935   </wse:Subscribe>
936 </soap:Body>
937 </soap:Envelope>
  
```

938 *R3011: A HOSTED SERVICE MUST NOT generate a wse:FilteringNotSupported SOAP Fault in*
 939 *response to a Subscribe SOAP ENVELOPE.*

940 A HOSTED SERVICE is required to support filtering, at least by [action], so the Filtering Not Supported
 941 SOAP Fault is not appropriate.

942 To indicate that a HOSTED SERVICE does not expose any Notifications that would match the contents of
 943 a Filter with the Action Dialect, this profile defines the following SOAP Fault:

[action]	http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/fault
[Code]	Soap:Sender
[Subcode]	dpws:FilterActionNotSupported

[Reason]	E.g., "no notifications match the supplied filter"
[Detail]	(None defined.)

944 *R3020: If none of the Notifications exposed by a HOSTED SERVICE match the [action] values in a*
945 *Subscribe SOAP ENVELOPE Filter whose Dialect is "http://docs.oasis-open.org/ws-*
946 *dd/ns/dpws/2009/01/Action", the HOSTED SERVICE SHOULD generate a*
947 *dpws:FilterActionNotSupported SOAP Fault.*

948 **5.2 Subscription Duration and Renewal**

949 *R3016: A HOSTED SERVICE MUST NOT generate a wse:UnsupportedExpirationType SOAP Fault in*
950 *response to a Subscribe or Renew SOAP ENVELOPE with an Expiration type of xs:duration.*

951 *R3013: A HOSTED SERVICE MAY generate a wse:UnsupportedExpirationType SOAP Fault in response*
952 *to a Subscribe or Renew SOAP ENVELOPE with an Expiration of type xs:dateTime.*

953 Event Sources are required to have an internal clock, but there is no requirement that the clock be
954 synchronized with clients or other HOSTED SERVICES. Event Sources are only required to support
955 Expirations expressed in duration, but they should attempt to match the type of the Subscription
956 Expiration when possible. If the value or type of the Expiration is unacceptable, the Event Source MAY
957 select an appropriate Expiration and return it in the Subscribe Response or Renew Response.

958 *R3015: A HOSTED SERVICE MAY generate a wsa:ActionNotSupported SOAP Fault in response to a*
959 *Get Status SOAP ENVELOPE.*

960 Event Sources are not required to support retrieving subscription status.

961 6 Security

962 This section defines a RECOMMENDED baseline for interoperable security between a DEVICE and a
963 CLIENT. A DEVICE (or CLIENT) is free to support other security mechanisms, and alternate profiles may
964 be developed to accommodate different deployment requirements. The use of alternate profiles may be
965 indicated by WSDL [[WSDL 1.1](#)], policies [[WS-Policy](#)], or by other means.

966 In the absence of an explicit indication stating that a different security mechanism is to be used, the
967 default security mechanism is determined by the transport addresses of the DEVICE: HTTP transport
968 addresses (URLs) indicate the device supports no security, and HTTPS transport addresses indicate the
969 device supports the security profile defined in this section.

970 A DEVICE may support more than one security profile, but security technologies do not always compose
971 in a way that results in interoperability. Implementers of multiple security profiles should take care to
972 preserve interoperability with each profile individually.

973 All requirements and recommendations in this section are conditional on the SERVICE or CLIENT
974 implementing this security profile. If a SERVICE or CLIENT does not implement the profile defined in this
975 section, it is not obligated to follow any of the requirements defined herein.

976 This scope of this section is the following set of Web services specifications. All of the requirements in
977 these specifications are included by reference except where superseded by normative statements herein:

- 978 • [[AES/TLS](#)]
- 979 • [[HTTP Authentication](#)]
- 980 • [[SHA](#)]
- 981 • [[TLS](#)]
- 982 • [[RFC 4122](#)]
- 983 • [[X.509.v3](#)]
- 984 • [[WS-Discovery](#)]

985 6.1 Terminology

986 Compact Signature

987 A WS-Discovery Compact Signature [[WS-Discovery](#)] is evidence of authenticity of the
988 unencrypted contents of a WS-Discovery message. The Compact Signature is included inside
989 the unencrypted message.

990 Secure Channel

991 A Secure Channel is a point-to-point transport-level TLS/SSL connection established between a
992 CLIENT and a SERVICE. Messages transmitted through a Secure Channel receive some
993 security protection, but that protection does not extend beyond the CLIENT and SERVICE that
994 established the channel.

995 Certificate

996 An x.509.v3 Certificate [[x.509.v3](#)] is a cryptographic credential that a SERVICE or a CLIENT use
997 for authentication. When a SERVICE or a CLIENT receives a Certificate from another entity, it
998 may inspect the contents to ensure they are valid credentials.

999 6.2 Model

1000 The security profile defined in this section has two parts: optional message-level signatures for UDP WS-
1001 Discovery traffic, and transport-level encryption. Transport-level encryption is mandatory for metadata
1002 and is optional for control traffic.

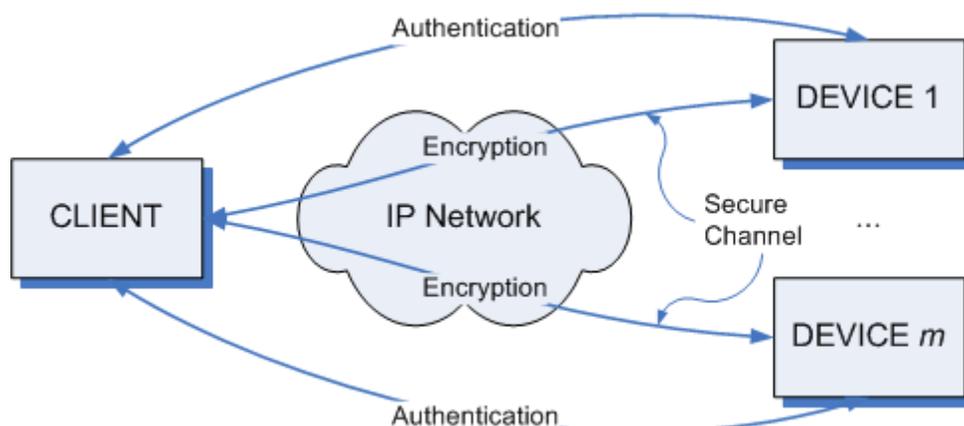
1003 WS-Discovery Compact Signatures allow a CLIENT to verify the integrity of multicast or unicast WS-
 1004 Discovery messages, and to identify WS-Discovery traffic that was signed by a DEVICE with a specific
 1005 cryptographic credential.

1006 TLS/SSL is used to establish a point-to-point Secure Channel between a CLIENT and a DEVICE, and
 1007 provides a mechanism for each participant to authenticate the identity of the other, and to verify the
 1008 integrity of the exchanged messages. It also provides confidentiality for all messages sent in the Secure
 1009 Channel established between the CLIENT and the DEVICE.

1010 A DEVICE uses an x.509.v3 certificate as its credential, and it uses this credential to sign WS-Discovery
 1011 messages and to establish TLS/SSL connections. A DEVICE may require CLIENT authentication in the
 1012 form of x.509.v3 certificates negotiated in the TLS/SSL connection, or username/password credentials
 1013 communicated through HTTP Authentication after the TLS/SSL connection is established.

1014 A DEVICE uses TLS/SSL to secure its HTTP traffic, and HOSTED SERVICES may also use TLS/SSL to
 1015 secure their HTTP traffic. A DEVICE may use a physical HTTPS address, or a logical address and
 1016 HTTPS xAddr. If a DEVICE and its HOSTED SERVICES are all reachable at the same address and
 1017 port, a CLIENT and DEVICE may reuse a TLS/SSL connection for multiple operations.

1018



1019

1020 **Figure 2: Communication mechanisms for authentication information and for encrypted messages**

1021 The organization of CLIENT and DEVICE credentials, mechanism for provisioning them, and criteria for
 1022 distinguishing valid and invalid credentials is out of scope of this profile.

1023 6.3 Endpoint Reference and xAddr

- 1024 *R5009: If a DEVICE uses a physical transport address for the [address] property of its Endpoint*
 1025 *Reference, it MUST be an HTTPS scheme IRI.*
- 1026 *R5012: A DEVICE MUST NOT advertise HTTP scheme addresses the xAddr fields of WS-Discovery*
 1027 *messages.*

1028 A DEVICE is prohibited from advertising non-secure HTTP transport addresses. It may advertise a
 1029 logical Endpoint Reference Address and HTTPS xAddr, or a physical HTTPS transport address for its
 1030 Endpoint Reference Address.

- 1031 *R5010: A SERVICE MAY use an HTTP scheme IRI for the [address] property of its Endpoint Reference.*

1032 A DEVICE may have secure HOSTED SERVICES, non-secure HOSTED SERVICES, neither, or both.
 1033 Secure HOSTED SERVICES must comply with the requirements for secure SERVICES in this section.

1034 6.4 Credentials

- 1035 *R4043: Each DEVICE SHOULD have its own, unique Certificate.*

1036 Restrictions in further subsections require that a DEVICE have an x.509.v3 certificate. R4043
 1037 recommends that this certificate is unique.

1038 *R4045: The format of the certificate MUST follow the common standard x.509.v3.*

1039 The Certificate contains information pertinent to the specific device including its public key. Typically,
1040 certificates are issued by a trusted authority or a delegate (2nd tier) or a delegate of the delegate.

1041 See [Appendix D](#) for an example x.509.v3 certificate.

1042 Provisioning of credentials, definition of valid credentials, and certificate management are out of the
1043 scope of this profile.

1044 *R4008: A SERVICE MAY use additional mechanisms to verify the authenticity of the SENDER of any*
1045 *received MESSAGE by analyzing information provided by the lower networking layers.*

1046 For example, a SERVICE may only allow CLIENTs whose IP address exists in a preconfigured list.

1047 **6.5 Discovery**

1048 *R4032: A DEVICE MUST NOT send a Probe Match SOAP ENVELOPE if the DEVICE is outside the local*
1049 *subnet of the CLIENT, and the Probe SOAP ENVELOPE was sent using the multicast binding as*
1050 *defined in WS-Discovery section 3.1.1.*

1051 *R4065: A CLIENT MUST discard a Probe Match SOAP ENVELOPE if it is received MATCH_TIMEOUT*
1052 *seconds or more later than the last corresponding Probe SOAP ENVELOPE was sent.*

1053 *R4036: A DEVICE MUST NOT send a Resolve Match SOAP ENVELOPE if the DEVICE is outside the*
1054 *local subnet of the CLIENT, and the Resolve SOAP ENVELOPE was sent using the multicast*
1055 *binding as defined in WS-Discovery section 3.1.1.*

1056 *R4066: A CLIENT MUST discard a Resolve Match SOAP ENVELOPE if it is received MATCH_TIMEOUT*
1057 *seconds or more later than the last corresponding Resolve SOAP ENVELOPE was sent.*

1058 **6.5.1 WS-Discovery Compact Signatures**

1059 *R5011: A DEVICE SHOULD sign its UDP discovery traffic using WS-Discovery Compact Signatures [WS-*
1060 *Discovery] to provide CLIENTs with a mechanism to verify the integrity of the messages, and to*
1061 *authenticate the DEVICE as the signor of the messages.*

1062 WS-Discovery Compact Signatures use WS-Security [[WS-Security](#)] to generate a cryptographic signature
1063 that can be used by a CLIENT to verify the validity of the unencrypted message.

1064 In cases where CLIENTs persist enough information about the credentials and presence of security on a
1065 DEVICE to protect against impersonation, the DEVICE may not sign its discovery messages.

1066 *R4017: A CLIENT MAY ignore MESSAGEs received during discovery that have no signature or a*
1067 *nonverifiable signature.*

1068 Messages signed with WS-Discovery Compact Signatures must also meet the requirements in sections
1069 6.7 Authentication and 6.8 Integrity.

1070 **6.6 Secure Channel**

1071 A TLS/SSL Secure Channel at the transport level is used to secure traffic between CLIENT and
1072 SERVICE.

1073 *R4057: All secure communication for Description, Control, and Eventing between the CLIENT and*
1074 *SERVICE MUST use a Secure Channel.*

1075 *R4072: A DEVICE MUST support receiving and responding to a Probe SOAP ENVELOPE over HTTP*
1076 *using a Secure Channel.*

1077 *R4073: A DEVICE MAY ignore a Probe SOAP ENVELOPE sent over HTTP that does not use a Secure*
1078 *Channel.*

1079 As described in [R1015](#), a CLIENT MAY send a Probe over HTTP; this Probe and ProbeMatches are sent
1080 using the Secure Channel.

1081 *R5013: A CLIENT MAY use a Secure Channel to contact multiple SERVICES if they can be reached at*
1082 *the same address and port.*

1083 *R4042: Following the establishment of a TLS/SSL Secure Channel, subsequent MESSAGE exchanges*
1084 *over HTTP SHOULD use the existing TLS/SSL session.*

1085 Secure Channels must also meet the minimum requirements in sections 6.7 Authentication, 6.8 Integrity,
1086 and 6.9 Confidentiality.

1087 **6.6.1 TLS/SSL Ciphersuites**

1088 *R4059: It is the responsibility of the sender to convert the embedded URL to use HTTPS as different*
1089 *transport security mechanisms can be negotiated.*

1090 *R4060: A SERVICE MUST support the following TLS Ciphersuite: TLS_RSA_WITH_RC4_128_SHA.*

1091 *R4061: It is recommended that a SERVICE also support the following TLS Ciphersuite:*
1092 *TLS_RSA_WITH_AES_128_CBC_SHA.*

1093 *R4062: Additional Ciphersuites MAY be supported. They are negotiated during the TLS/SSL handshake.*

1094 Where appropriate, DEVICES are encouraged to support additional Ciphersuites that rely on more robust
1095 security technology, such as the SHA-2 [SHA] family of hashing standards.

1096 *R5014: A SERVICE SHOULD NOT negotiate any of the following TLS/SSL Ciphersuites: (a)*
1097 *TLS_RSA_WITH_NULL_SHA, (b) SSL_RSA_WITH_NULL_SHA, (c) any Ciphersuite with*
1098 *DH_anon in their symbolic name, (d) any Ciphersuites with MD5 in their symbolic name.*

1099 **6.6.2 SERVICE Authentication in a Secure Channel**

1100 X.509.v3 certificates are the only mechanism for a CLIENT to authenticate a DEVICE or a HOSTED
1101 SERVICE (if TLS/SSL is supported on that HOSTED SERVICE).

1102 *R4039: A CLIENT MUST initiate authentication with the DEVICE by setting up a TLS/SSL session.*

1103 *R5017: If a SERVICE uses TLS/SSL, it MUST authenticate itself to a CLIENT by supplying an X.509v3*
1104 *certificate during the TLS/SSL handshake.*

1105 **6.6.3 CLIENT Authentication in a Secure Channel**

1106 *R4014: A DEVICE MAY require authentication of a CLIENT.*

1107 A DEVICE may authenticate a CLIENT by negotiating and x.509.v3 certificate, or by requesting a
1108 username and password communicated over HTTP Authentication inside the Secure Channel.

1109 X.509.v3 certificates are the preferred mechanism for authenticating a CLIENT.

1110 *R4018: A DEVICE SHOULD cache authentication information for a CLIENT as valid as long as the*
1111 *DEVICE is connected to the CLIENT.*

1112 **6.6.3.1 CLIENT Authentication with x.509.v3 certificates**

1113 *R4071: If the CLIENT and the SERVICE exchanged certificates during the TLS/SSL handshake, and the*
1114 *SERVICE as well as the CLIENT were able to verify the certificates, the CLIENT and SERVICE*
1115 *are mutually authenticated, and no further steps SHALL be required.*

1116 **6.6.3.2 CLIENT Authentication with HTTP Authentication**

1117 In cases where x.509.v3 client certificates are unavailable or where validation is infeasible, the DEVICE
1118 may use HTTP Authentication [HTTP/1.1] to request client credentials.

1119 HTTP authentication requires credentials in the form of username and password. It is assumed that how
1120 the CLIENT and SERVICE share knowledge of the username and password is out-of-band and beyond
1121 the scope of this profile.

1122 Because the authentication is performed over the Secure Channel established during TLS/SSL
1123 handshake and after the CLIENT has authenticated the SERVICE, HTTP Basic authentication may be
1124 used safely.

1125 *R4046: A SERVICE MAY require HTTP Authentication step after the TLS/SSL handshake, if the*
1126 *SERVICE was not able to verify the certificate, or if the CLIENT did not provide a certificate*
1127 *during the TLS/SSL handshake.*

1128 *R4048: If the HTTP authentication is successful, and the CLIENT presents a certificate to the SERVICE,*
1129 *the SERVICE SHOULD cache the certificate in its local certificate store of trusted certificates for*
1130 *future authentication of the CLIENT.*

1131 R4048 avoids the need for HTTP authentication for subsequent connections.

1132 *R4050: If a SERVICE requires HTTP authentication, the SERVICE SHALL challenge the CLIENT using*
1133 *the HTTP 401 response code.*

1134 *R4051: A CLIENT MUST authenticate using one of the options listed in the HTTP-Authenticate header.*

1135 *R4052: HTTP Authentication MUST use the following parameters for username and password of the*
1136 *HTTP Request: username, PIN / password.*

1137 The username is supplied to the SERVICE during HTTP authentication and MAY be used for establishing
1138 multiple access control classes, such as administrators, users, and guests. The naming and use of
1139 username is implementation-dependent and out of the scope of this profile.

1140 *R4053: If no username is provided, "admin" SHALL be used as the default username.*

1141 The purpose of the PIN / password is to authenticate the CLIENT to the DEVICE during the HTTP
1142 authentication.

1143 *R4054: The RECOMMENDED size of a PIN / password is at least 8 characters using at least a 32*
1144 *character alphabet.*

1145 *R4055: The PIN / password that is unique to the SERVICE SHALL be conveyed to the CLIENT out-of-*
1146 *band. The methods of conveying the PIN out-of-band are out of the scope of this profile.*

1147 *R4056: To reduce the attack surface, the SERVICE and CLIENT MAY limit the number of failed*
1148 *authentication attempts as well as the time interval successive attempts are made for one*
1149 *TLS/SSL session.*

1150 **6.7 Authentication**

1151 Authentication is the process by which the identity of the sender is determined by the recipient.
1152 Authentication MUST adhere to the following requirements:

1153 *R4004: A SENDER MUST authenticate itself to a RECEIVER using credentials acceptable to the*
1154 *RECEIVER.*

1155 In this profile, authentication is done using certificates or a combination of certificates and HTTP
1156 authentication. If multicast messages are secured, the following additional requirements apply:

1157 *R4005: On multicast MESSAGEs, a CLIENT MUST use an authentication credential that is suitable for all*
1158 *DEVICEs that could legitimately process the multicast MESSAGE.*

1159 *R5023: If a SERVICE uses TLS/SSL, it MUST provide Authentication (as defined in this section) for any*
1160 *TLS/SSL connections.*

1161 Credentials MAY be cached on the DEVICE and/or CLIENT to simplify subsequent authentications.

1162 6.8 Integrity

1163 Integrity is the process that protects MESSAGES against tampering while in transit. Integrity MUST
1164 adhere to the following requirements:

1165 *R5015: If a SERVICE uses TLS/SSL or WS-Discovery Compact Signatures, it MUST provide Integrity (as*
1166 *defined in this section) for any TLS/SSL connections or signatures, respectively.*

1167 *R4000: A SERVICE MUST not send a SOAP ENVELOPE without protecting the integrity of any Message*
1168 *Information Header blocks matching the following XPath expressions: (a)*
1169 */soap:Envelope/soap:Header/wsa:Action, (b) /soap:Envelope/soap:Header/wsa:MessageID, (c)*
1170 */soap:Envelope/soap:Header/wsa:To, (d) /soap:Envelope/soap:Header/wsa:ReplyTo, (e)*
1171 */soap:Envelope/soap:Header/wsa:RelatesTo, and (f)*
1172 */soap:Envelope/soap:Header/*[@isReferenceParameter='true'].*

1173 *R4063: A SERVICE MAY reject a SOAP ENVELOPE that has unprotected Message Information Header*
1174 *blocks.*

1175 *R4001: A SERVICE MUST not send a SOAP ENVELOPE (including SOAP Faults) without protecting the*
1176 *integrity of the SOAP ENVELOPE Body in conjunction with any Message Information Block(s)*
1177 *from R4000.*

1178 *R4064: A SERVICE MAY reject a SOAP ENVELOPE that does not protect the integrity of the SOAP*
1179 *ENVELOPE Body.*

1180 In this profile, the integrity of UDP discovery SOAP ENVELOPES is protected using message-level
1181 signatures, while the integrity of other MESSAGES is protected using a Secure Channel.

1182 6.9 Confidentiality

1183 Confidentiality is the process by which sensitive information is protected against unauthorized disclosure
1184 while in transit. Confidentiality MUST adhere to the following requirements:

1185 *R5016: If a SERVICE uses TLS/SSL, it MUST provide Confidentiality (as defined in this section) for any*
1186 *TLS/SSL connections.*

1187 *R4002: A SERVICE MUST NOT send a SOAP ENVELOPE without encrypting the SOAP ENVELOPE*
1188 *Body.*

1189 *R4067: A SERVICE MAY reject a SOAP ENVELOPE that does not encrypt the SOAP ENVELOPE Body.*

1190 In this profile, UDP WS-Discovery MESSAGES are not treated as confidential. Confidential MESSAGES
1191 are encrypted using a Secure Channel.

1192 7 Conformance

1193 An endpoint is expected to implement at least one of the roles defined herein ([DEVICE](#), [CLIENT](#), or
1194 [HOSTED SERVICE](#)) and MAY implement more than one of the roles. An endpoint is not compliant with
1195 this specification if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined
1196 herein for the roles it implements.

1197 Normative text within this specification takes precedence over normative outlines, which in turn take
1198 precedence over the XML Schema [[XML Schema Part 1](#), [Part 2](#)] descriptions, which in turn take
1199 precedence over examples.

1200 Appendix A. Acknowledgements

1201 The following individuals have participated in the creation of this specification and are gratefully
1202 acknowledged:

1203 **Participants:**

1204 Geoff Bullen, Microsoft Corporation
1205 Steve Carter, Novell
1206 Dan Conti, Microsoft Corporation
1207 Doug Davis, IBM
1208 Scott deDeugd, IBM
1209 Oliver Dohndorf, Technische Universitat Dortmund
1210 Dan Driscoll, Microsoft Corporation
1211 Colleen Evans, Microsoft Corporation
1212 Max Feingold, Microsoft Corporation
1213 Travis Grigsby, IBM
1214 Francois Jammes, Schneider Electric
1215 Ram Jeyaraman, Microsoft Corporation
1216 Mike Kaiser, IBM
1217 Supun Kamburugamuva, WSO2
1218 Devon Kemp, Canon Inc.
1219 Akira Kishida, Canon Inc.
1220 Jan Krueger, Technische Universitaet Dortmund
1221 Mark Little, Red Hat
1222 Dr. Ingo Lueck, Technische Universitaet Dortmund
1223 Jonathan Marsh, WSO2
1224 Carl Mattocks
1225 Antoine Mensch
1226 Jaime Meritt, Progress Software
1227 Vipul Modi, Microsoft Corporation
1228 Anthony Nadalin, IBM
1229 Tadahiro Nakamura, Canon Inc.
1230 Masahiro Nishio, Canon Inc.
1231 Toby Nixon, Microsoft Corporation
1232 Shin Ohtake, Fuji Xerox Co., Ltd.
1233 Venkat Reddy, CA
1234 Alain Regnier, Ricoh Company, Ltd.
1235 Hitoshi Sekine, Ricoh Company, Ltd.
1236 Yasuji Takeuchi, Konica Minolta Business Technologies
1237 Hiroshi Tamura, Ricoh Company, Ltd.
1238 Minoru Torii, Canon Inc.
1239 Asir S Vedamuthu, Microsoft Corporation
1240 David Whitehead, Lexmark International Inc.
1241 Don Wright, Lexmark International Inc.
1242 Prasad Yendluri, Software AG, Inc.
1243 Elmar Zeeb, University of Rostock
1244 Gottfried Zimmermann

1245

1246 **Co-developers of the initial contributions:**

1247 This document is based on initial contributions to the OASIS WS-DD Technical Committee by the follow
1248 co-developers:

1249 Shannon Chan, Microsoft Corporation
1250 Dan Conti, Microsoft Corporation
1251 Chris Kaler, Microsoft Corporation

1252 Thomas Kuehnel, Microsoft Corporation
1253 Alain Regnier, Ricoh Company Limited
1254 Bryan Roe, Intel Corporation
1255 Dale Sather, Microsoft Corporation
1256 Jeffrey Schlimmer, Microsoft Corporation (Editor)
1257 Hitoshi Sekine, Ricoh Company Limited
1258 Jorgen Thelin, Microsoft Corporation (Editor)
1259 Doug Walter, Microsoft Corporation
1260 Jack Weast, Intel Corporation
1261 Dave Whitehead, Lexmark International Inc.
1262 Don Wright, Lexmark International Inc.
1263 Yevgeniy Yarmosh, Intel Corporation

1264

Appendix B. Constants

1265 The following constants are used throughout this profile. The values listed below supersede other values
1266 defined in other specifications listed below.

Constant	Value	Specification
APP_MAX_DELAY	2,500 milliseconds	[WS-Discovery]
DISCOVERY_PORT	3702	[WS-Discovery]
MATCH_TIMEOUT	10 seconds	[WS-Discovery]
MAX_ENVELOPE_SIZE	32,767 octets	This profile
MAX_UDP_ENVELOPE_SIZE	4,096 octets	This profile
MAX_FIELD_SIZE	256 Unicode characters	This profile
MAX_URI_SIZE	2,048 octets	This profile
MULTICAST_UDP_REPEAT	1	[SOAP-over-UDP]
UDP_MAX_DELAY	250 milliseconds	[SOAP-over-UDP]
UDP_MIN_DELAY	50 milliseconds	[SOAP-over-UDP]
UDP_UPPER_DELAY	450 milliseconds	[SOAP-over-UDP]
UNICAST_UDP_REPEAT	1	[SOAP-over-UDP]

1267

Appendix C. Declaring Discovery Types in WSDL

1268 Solutions built on DPWS often define portTypes implemented by Hosted Services, and a discovery-layer
1269 portType implemented by the Host Service so the presence of these functional services can be
1270 determined at the discovery layer. The binding between a service-layer type and its discovery-layer type
1271 can be defined purely in descriptive text, but this appendix provides an optional mechanism to declare a
1272 discovery-layer type inside WSDL that can be consumed and understood by tools.

1273 This appendix defines an @dpws:DiscoveryType attribute to annotate the WSDL 1.1 portType [WSDL
1274 1.1] for the service-layer type. The normative outline for @dpws:DiscoveryType is:

```
1275 <wsdl:definitions ...>  
1276   [<wsdl:portType [dpws:DiscoveryType="xs:QName"]? >  
1277     ...  
1278     </wsdl:portType>]*  
1279 </wsdl:definitions>
```

1280 The following describes additional, normative constraints to the outline listed above:

1281 /wsdl:definitions/wsdl:portType/@dpws:DiscoveryType

1282 The name of the portType to be advertised by the Host Service to indicate that this device
1283 supports the annotated Hosted Service portType.

1284 If omitted, no implied value

1285 This mechanism is only suitable in cases where a functional service type is bound to a single discovery-
1286 layer type, and authors of more complex type topologies may express the relationship between service
1287 and discovery types through normative text or through other means.

1288 Example usage follows. PrintDeviceType is the discovery-layer type for PrintPortType.

```
1289 <wsdl:definitions  
1290   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
1291   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"  
1292   targetNamespace="http://printer.example.com/imaging"  
1293   xmlns:tns="http://printer.example.com/imaging">  
1294  
1295   <wsdl:portType name="PrintPortType"  
1296     dpws:DiscoveryType="tns:PrintDeviceType">  
1297  
1298     <!-- Contents omitted for brevity -->  
1299  
1300   </wsdl:portType>  
1301  
1302   <!-- Define PrintDeviceType to be empty -->  
1303   <wsdl:portType name="PrintDeviceType" />  
1304  
1305 </wsdl:definitions>
```

1306

Appendix D. Example x.509.v3 Certificate

1307 An example of a self-signed X.509 certificate is shown below. In this case, the Subject field contains the
1308 UUID in string representation format (i.e., not represented numerically).

Type	Element	Usage	Example
Basic Elements	Version	TLS	3
	Certificate Serial Number		1234567
	Certificate Algorithm Identifier		RSA
	Issuer		a7731471-4b54-4a64-942c-7d481dcb9614
	Validity Period		11/09/2001 - 01/07/2015
	Subject		a7731471-4b54-4a64-942c-7d481dcb9614
	Subject Public Key Information		rsaEncryption 1024 10888232e76740bd873462ea2c64ca1d a6f9112656a34b949d32cede0e476547 84ba0f7e62e143429d3217ee45ce5304 308e65a6eee6474cb4d9a3c0295c8267 761661ccba7546a09d5f03a8ea3b1160 dac9fb6e6ba94e54b6c8ee892e492f4c e3a96bbd9d7b4c4bb98b7c052ff361ba cee01718122c4f0d826efc123bb1b03d
Extensions	Extended Key Usage	Server Authentication	1.3.6.1.5.5.7.3.1
		Client Authentication	1.3.6.1.5.5.7.3.2
Signature	Certificate Authority's Digital Signature		5938f9908916cca32321916a184a6e75 2becb14fb99c4f33a03b03c3c752117c 91b8fb163d3541fca78bca235908ba69 1f7e36004a2d499a8e23951bd8af961d 36be05307ec34467a7c66fbb7fb5e49c 25e8dbdae4084ca9ba244b5bc1a377e5 262b9ef543ce47ad8a6b1d28c9138d0a dc8f5e3b469e42a5842221f9cf0a50d1

1309

1310

Appendix E. Revision History

1311 [optional; should not be included in OASIS Standards]

1312

Revision	Date	Editor	Changes Made
wd-01	09/16/2008	Dan Driscoll	Converted input specification to OASIS template.
wd-02	10/08/2008	Dan Driscoll	Resolved the following issues: <ul style="list-style-type: none"> • 001: Clarify R4032 and R4036 w.r.t. other multicast bindings • 002: Define matching for empty Action filter • 003: Fault Action should use lowercase 'f' • 004: Faulting to non-anonymous endpoints • 005: SOAP Binding should apply to clients • 013: Restrict encoding of SOAP messages to UTF-8 • 016: Edit R0042 • 028: Review constants • 045: EndpointReference subelement • 061: Assign an OASIS namespace for the specifications
wd-02	10/14/2008	Dan Driscoll	<ul style="list-style-type: none"> • Changed document format from doc to docx • Fixed "authoritative reference"
wd-02	10/14/2008	Dan Driscoll	<ul style="list-style-type: none"> • Changed version number to 1.1 • Removed "related work" section
wd-02	10/14/2008	Dan Driscoll	<ul style="list-style-type: none"> • Changed copyrights from 2007 to 2008
wd-03	12/12/2008	Dan Driscoll	<ul style="list-style-type: none"> • Changed draft from cd-01 to wd-03 • Updated dates to 2008/12/12 • Updated namespace to 2009/01 • Issue 098: Update namespace • Editorial: Changed 'wsdp' prefix to 'dpws'
wd-03	12/12/2008	Dan Driscoll Antoine Mensch	<ul style="list-style-type: none"> • 011: Fix SERVICE terminology • 015: Remove R0007 • 024: Fix Directed Discovery

			<ul style="list-style-type: none"> • 029: Fix SERVICE/DEVICE for WS-Policy • 038: Contents of Host EPR • 039: Recursive hosting • 055: WS-Addressing 1.0 • 070: HTTP content negotiation for PresentationUrl • 071: Update to WS-Policy 1.5 • 073: Clarify "stable" identifier • 074: Clarify R0036/R0037 • 075: Clarify "Target Service" • 077: Remove R3010 as redundant • 080: Secure all WS-A headers • 084: Faulting behavior on Subscribe • 085: Get/GetMetadata • 092: Split R3019 • 093: Remove R3012 • 094: Clean up expiration type/value switching • 095: Clarify expiration value switching • 109: Update references
wd-03	1/2/2009	Dan Driscoll	<ul style="list-style-type: none"> • 032: Describe security composability • 051: Generalize security • 112: Remove WS-Security reference • 113: Cleanup Network Model • 114: Remove security negotiation • 115: Replace R4070 with switches on HTTPS ID/xAddrs • 138: Create introduction and concrete description of security profile • 139: Remove protocol negotiation • 140: Clean up HTTP Authentication
wd-03	1/21/2009	Antoine Mensch	<ul style="list-style-type: none"> • Issue 012 • Issue 040 • Issue 046 • Issue 117 • Issue 127 • Issue 128 • Issue 135 • Issue 143
cd-02	1/21/2009	Dan Driscoll	<ul style="list-style-type: none"> • Changed draft from wd-03 to cd-02

Candidate			<ul style="list-style-type: none"> • Updated date, copyrights • Updated WS-Discovery and SOAP-over-UDP references to CD-02 • 072: Fix HOSTEDSERVICE • 083: Fix R0031 and wsa:ReplyTo • 130: Make FilterActionNotSupported recommended, not mandatory • 132: Define relative PresentationUrl • 134: Make Types/Scopes mandatory in directed ProbeMatches • 137: Add Appendix C • More security edits (see Section 7)
cd-02 Candidate	1/26/2009	Dan Driscoll	<ul style="list-style-type: none"> • Fixed WS-DD committee site links • Added TC participants to Appendix A; remove company names to meet OASIS rules • Removed "Last Approved Version"
cd-02	1/27/2009	Dan Driscoll	<ul style="list-style-type: none"> • Updated to reflect CD-02 status
pr-01	1/30/2009	Dan Driscoll	<ul style="list-style-type: none"> • Updated to reflect PR-01 status
wd-04	2/10/2009	Dan Driscoll	<ul style="list-style-type: none"> • Changed draft from PR-01 to WD-04 • Updated references to WS-Discovery and SOAP-over-UDP
wd-04	2/11/2009	Dan Driscoll	<ul style="list-style-type: none"> • 150: Add pointer to RDDDL and XSD • 151: Reorder terminology section • Reformat references section • Reformat appendix headers • Add missed internal hyperlinks
wd-04	2/20/2009	Dan Driscoll	<ul style="list-style-type: none"> • 147: Add URL for Directed Probe • 154: Fix R0031 • 155: Update XML schema references
wd-05	2/25/2009	Dan Driscoll	<ul style="list-style-type: none"> • 148: Reorganize security section
wd-06	4/9/2009	Dan Driscoll	<ul style="list-style-type: none"> • Updated draft from WD-05 to WD-06 • Update list of TC participants • Pr007.1: review non-normative RFC2119 keywords • Pr007.2: cross-reference roles to terms/definitions • Pr007.4: Update conformance section
cd-03	4/14/2009	Dan Driscoll	<ul style="list-style-type: none"> • Updated to reflect CD-03 status
cd-04	4/28/2009	Dan Driscoll	<ul style="list-style-type: none"> • Updated to reflect CD-04 status

