



Devices Profile for Web Services Version 1.1

Committee Draft 02

27 January 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-02/wsdd-dpws-1.1-spec-cd-02.html>
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-02/wsdd-dpws-1.1-spec-cd-02.docx> (Authoritative Format)
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-02/wsdd-dpws-1.1-spec-cd-02.pdf>

Previous Version:

<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-01/wsdd-dpws-1.1-spec-cd-01.html>
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-01/wsdd-dpws-1.1-spec-cd-01.docx>
<http://docs.oasis-open.org/ws-dd/dpws/1.1/cd-01/wsdd-dpws-1.1-spec-cd-01.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	6
1.1	Requirements	6
1.2	Terminology	6
1.3	Notational Conventions.....	7
1.4	XML Namespaces	8
1.5	Normative References	8
1.6	Non-Normative References	10
2	Messaging	11
2.1	URI.....	11
2.2	UDP	11
2.3	HTTP.....	11
2.4	SOAP Envelope.....	12
2.5	WS-Addressing.....	12
2.6	Attachments	13
3	Discovery.....	14
4	Description.....	16
4.1	Characteristics	16
4.2	Hosting.....	19
4.3	WSDL.....	22
4.4	WS-Policy	24
5	Eventing.....	26
5.1	Subscription	26
5.1.1	Filtering.....	26
5.2	Subscription Duration and Renewal	28
6	Security.....	29
6.1	Terminology	29
6.2	Model	29
6.3	Integrity	30
6.4	Confidentiality	30
6.5	Authentication	31
6.6	Trust.....	31
6.7	DEVICE Behavior	31
6.8	Security for Discovery.....	31
6.9	Authentication	32
6.9.1	Transport Layer Security (TLS/SSL)	32
6.9.2	Certificates.....	32
6.9.3	DEVICE Authentication with TLS/SSL	33
6.9.4	CLIENT Authentication with TLS/SSL	33
6.9.5	CLIENT Authentication with HTTP Authentication	34
6.10	Secure Channel	34
6.11	TLS/SSL Ciphersuites	34
7	Conformance	36
A.	Acknowledgements	37

B. Constants	39
C. Declaring Discovery Types in WSDL	40
D. Revision History.....	41

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

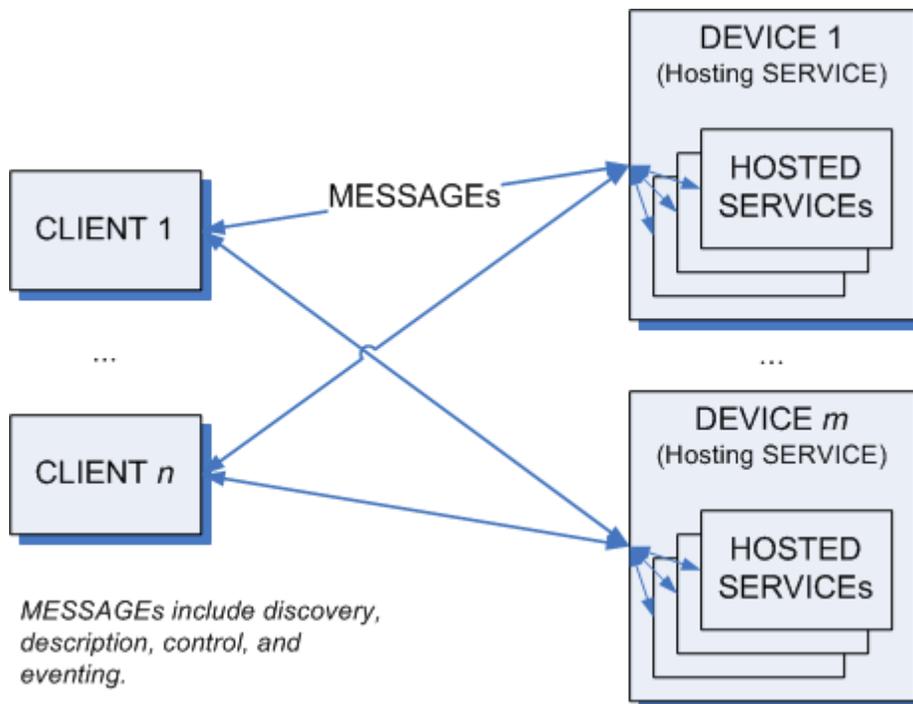
In each of these areas of scope, this profile defines minimal implementation requirements for compliant Web service implementations.

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



20 MESSAGE

21 Protocol elements that are exchanged, usually over a network, to affect a Web service. Always
22 includes a SOAP ENVELOPE. Typically also includes transport framing information such as
23 HTTP headers, TCP headers, and IP headers.
24

25 SOAP ENVELOPE

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

29 MIME SOAP ENVELOPE

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

31 TEXT SOAP ENVELOPE

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

33 CLIENT

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

35 SERVICE

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

38 DEVICE

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

41 HOSTED SERVICE

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

46 SENDER

47 A CLIENT or SERVICE that sends a MESSAGE.

48 RECEIVER

49 A CLIENT or SERVICE that receives a MESSAGE.

50 1.3 Notational Conventions

51 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
52 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described
53 in [RFC 2119].

- 54 • This specification uses the following syntax to define normative outlines for messages:
- 55 • The syntax appears as an XML instance, but values in italics indicate data types instead of literal
56 values.
- 57 • Characters are appended to elements and attributes to indicate cardinality:
 - 58 ○ "?" (0 or 1)
 - 59 ○ "*" (0 or more)
 - 60 ○ "+" (1 or more)
- 61 • The character "|" is used to indicate a choice between alternatives.
- 62 • The characters "(" and ")" are used to indicate that contained items are to be treated as a group
63 with respect to cardinality or choice.
- 64 • The characters "[" and "]" are used to call out references and property names.
- 65 • Ellipses (i.e., "...") indicate points of extensibility. Additional children and/or attributes MAY be
66 added at the indicated extension points but MUST NOT contradict the semantics of the parent
67 and/or owner, respectively. By default, if a receiver does not recognize an extension, the receiver
68 SHOULD ignore the extension; exceptions to this processing rule, if any, are clearly indicated
69 below.

- XML namespace prefixes (see Table 1) are used to indicate the namespace of the element being defined.

This specification uses the **[action]** and Fault properties [\[WS-Addressing\]](#) to define faults.

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

Rnnn: Normative statement text goes here.

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

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

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

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

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]

1.5 Normative References

- [RFC 2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [AES/TLS]** P.Chown, *Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)*, <http://www.ietf.org/rfc/rfc3268.txt>, IETF RFC 3268, June 2004.
- [BP 1.1, Section 4]** K. Ballinger, et al, *Basic Profile Version 1.1, Section 4: Service Description*, <http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html#description>, August 2004.
- [HTTP/1.1]** R.Fielding, et al, *Hypertext Transfer Protocol -- HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>, IETF RFC 2616, June 1999.
- [HTTP Authentication]** J. Franks, et al, *HTTP Authentication: Basic and Digest Access Authentication*, <http://www.ietf.org/rfc/rfc2617.txt>, IETF RFC 2617, June 1999.
- [MIME]** N. Freed, et al, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, <http://www.ietf.org/rfc/rfc2045.txt>, IETF RFC 2045, November 1996.

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

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

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

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

108 **[SOAP 1.2, Part 2]**

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

112 **[SOAP-over-UDP]** OASIS Committee Draft 02, *SOAP-over-UDP*, <http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cd-02/wsdd-soapoverudp-1.1-spec-cd-02.docx>, 27 January
113 2009.

114

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

117 **[WS-Addressing]** W3C Recommendation, *Web Services Addressing 1.0 - Core*,
118 <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509>, 9 May, 2006.

119 **[WS-Discovery]** OASIS Committee Draft 02, *Web Services Dynamic Discovery (WS-Discovery)*,
120 <http://docs.oasis-open.org/ws-dd/discovery/1.1/cd-02/wsdd-discovery-1.1-spec-cd-02.docx>, 27 January 2009.
121

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

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

125 K. Ballinger, et al, *WSDL 1.1 Binding Extension for SOAP 1.2*,
126 <http://www.w3.org/Submission/2006/SUBM-wsdl11soap12-20060405/>, 5 April
127 2006.

128 **[WS-Eventing]** D. Box, et al, *Web Services Eventing (WS-Eventing)*,
129 <http://www.w3.org/Submission/2006/SUBM-WS-Eventing-20060315/>, 15 March
130 2006.

131 **[WS-MetadataExchange]**

132 K. Ballinger, et al, *Web Services Metadata Exchange 1.1 (WS-*
133 *MetadataExchange)*, [http://www.w3.org/Submission/2008/SUBM-WS-](http://www.w3.org/Submission/2008/SUBM-WS-MetadataExchange-20080813/)
134 [MetadataExchange-20080813/](http://www.w3.org/Submission/2008/SUBM-WS-MetadataExchange-20080813/), 13 August 2008.

135 **[WS-Policy]** W3C Recommendation, *Web Services Policy 1.5 - Framework*,
136 <http://www.w3.org/TR/2007/REC-ws-policy-20070904/>, 4 September 2007.

137 **[WS-PolicyAttachment]**

138 W3C Recommendation, *Web Services Policy 1.5 - Attachment*,
139 <http://www.w3.org/TR/2007/REC-ws-policy-attach-20070904/>, 4 September
140 2007.

141 **[WS-Transfer]** J. Alexander, et al, *Web Service Transfer (WS-Transfer)*,
142 <http://www.w3.org/Submission/2006/SUBM-WS-Transfer-20060927/>, 27
143 September 2006.

144 **[X.509.v3]** *ITU-T X.509.v3 Information technology - Open Systems Interconnection - The*
145 *Directory: Public-key and attribute certificate frameworks (ISO/IEC/ITU 9594-8)*

146 **[XML Schema, Part 1]**

147 H. Thompson, et al, *XML Schema Part 1: Structures*,
148 <http://www.w3.org/TR/2001/REC-xmlschema-1/20010502/>, May 2001.

149 **[XML Schema, Part 2]**

150 P. Biron, et al, *XML Schema Part 2: Datatypes*, [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/)
151 [xmlschema-2-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/), May 2001.

152

153 **1.6 Non-Normative References**

154 **[IPv6 Autoconfig]** S. Thomson, et al, *IPv6 Stateless Address Autoconfiguration*,
155 <http://www.ietf.org/rfc/2462.txt>, IETF RFC 2462, December 1998.

156 **[DHCP]** R. Droms, et al, *Dynamic Host Configuration Protocol*,
157 <http://www.ietf.org/rfc/2131.txt>, IETF RFC 2131, March 1997.

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

160 **[WS-Security]** OASIS Standard Specification, *Web Services Security: SOAP Message Security*
161 *1.1 (WS-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)
162 [SOAPMessageSecurity.pdf](http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf), 1 February 2006.

163 2 Messaging

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

- 166 • [SOAP 1.2, Part 1]
- 167 • [SOAP 1.2, Part 2]
- 168 • [SOAP-over-UDP]
- 169 • [HTTP/1.1]
- 170 • [WS-Addressing]
- 171 • [RFC 4122]
- 172 • [MTOM]

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

176 2.1 URI

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

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

179 The constant MAX_URI_SIZE is defined in Appendix B -- Constants.

180 2.2 UDP

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

183 To improve reliability, a SERVICE should minimize the size of SOAP ENVELOPEs sent over UDP.
184 However, some SOAP ENVELOPEs may be larger than an MTU; for example, a signed Hello SOAP
185 ENVELOPE. If a SOAP ENVELOPE is larger than an MTU, the underlying IP network stacks may
186 fragment and reassemble the UDP packet.

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

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

191 Unlike TCP or HTTP messages, UDP datagrams must be received in one chunk, which may lead to
192 excessive resource requirements when receiving large datagrams on small embedded systems. The
193 constant MAX_UDP_ENVELOPE_SIZE is defined in Appendix B -- Constants.

194 2.3 HTTP

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

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

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

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

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

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

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

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

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

210 2.4 SOAP Envelope

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

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

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

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

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

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

219 2.5 WS-Addressing

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

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

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

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

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

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

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

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

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

236 R0031: A SERVICE MUST NOT generate a `wsa:InvalidMessageInformationHeader` SOAP Fault if the
237 [address] of the [reply endpoint] of an HTTP Request Message SOAP ENVELOPE is
238 "<http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous>".

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

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

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

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

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

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

255 2.6 Attachments

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

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

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

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

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

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

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

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

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

278 3 Discovery

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

- 281 • [WS-Discovery]

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

325 DEVICES may omit their Types and Scopes in their UDP WS-Discovery messages to reduce message
326 size and prevent fragmentation. However, they are obligated to return all Types and Scopes in their
327 HTTP ProbeMatches messages as increased risk of packet loss due to fragmentation is not a
328 consideration.

329 4 Description

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

- 332 • [XML Schema Part 1, Part 2]
- 333 • [WSDL 1.1]
- 334 • [BP 1.1, Section 4]
- 335 • [WSDL Binding for SOAP 1.2]
- 336 • [WS-MetadataExchange]
- 337 • [WS-Policy]
- 338 • [WS-PolicyAttachment]
- 339 • [WS-Transfer]

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

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

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

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

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

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

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

360 A DEVICE is not required to support all of the operations defined in [WS-Transfer].

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

363 4.1 Characteristics

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

```
366 <dpws:ThisModel ...>  
367   <dpws:Manufacturer xml:lang="..."? >xs:string</dpws:Manufacturer>+  
368   <dpws:ManufacturerUrl>xs:anyURI</dpws:ManufacturerUrl?>  
369   <dpws:ModelName xml:lang="..."? >xs:string</dpws:ModelName>+  
370   <dpws:ModelNumber>xs:string</dpws:ModelNumber?>  
371   <dpws:ModelUrl>xs:anyURI</dpws:ModelUrl?>  
372   <dpws:PresentationUrl>xs:anyURI</dpws:PresentationUrl?>
```

373 ...
374 </dpws:ThisModel>
375 The following describes additional, normative constraints on the outline above:
376 dpws:ThisModel/ dpws:Manufacturer
377 Name of the manufacturer of the DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode
378 characters, SHOULD be localized, and SHOULD be repeated for each supported locale.
379 dpws:ThisModel/ dpws:ManufacturerUrl
380 URL to a Web site for the manufacturer of the DEVICE. It MUST have fewer than
381 MAX_URI_SIZE octets.
382 dpws:ThisModel/ dpws:ModelName
383 User-friendly name for this model of device chosen by the manufacturer. It MUST have fewer
384 than MAX_FIELD_SIZE Unicode characters, SHOULD be localized, and SHOULD be repeated
385 for each supported locale.
386 dpws:ThisModel/ dpws:ModelNumber
387 Model number for this model of DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode
388 characters.
389 dpws:ThisModel/ dpws:ModelUrl
390 URL to a Web site for this model of DEVICE. It MUST have fewer than MAX_URI_SIZE octets.
391 dpws:ThisModel/ dpws:PresentationUrl
392 URL to a presentation resource for this DEVICE. It MAY be relative to the HTTP transport
393 address over which metadata was retrieved, and MUST have fewer than MAX_URI_SIZE octets.
394 If PresentationUrl is specified, the DEVICE MAY provide the resource in multiple formats, but
395 MUST at least provide an HTML page. CLIENTs and DEVICEs MAY use HTTP content
396 negotiation [HTTP/1.1] to determine the format and content of the presentation resource.
397 DEVICEs that use a relative URL MAY use HTTP Redirection 3xx codes [HTTP/1.1] to direct
398 CLIENTs to a dedicated web server running on another port.

399 CORRECT:

```

400 <dpws:ThisModel
401    xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
402    <dpws:Manufacturer>ACME Manufacturing</dpws:Manufacturer>
403    <dpws:ModelName xml:lang="en-GB" >ColourBeam 9</dpws:ModelName>
404    <dpws:ModelName xml:lang="en-US" >ColorBeam 9</dpws:ModelName>
405 </dpws:ThisModel>

```

406 A Dialect [WS-MetadataExchange] equal to "http://docs.oasis-open.org/ws-
407 dd/ns/dpws/2009/01/ThisModel" indicates an instance of the ThisModel metadata format.

408 No Identifier [WS-MetadataExchange] is defined for instances of the ThisModel metadata format.

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

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

413 Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve the resource representation data
414 for a DEVICE – which includes the ThisModel metadata for a DEVICE. A DEVICE may also provide other
415 means for a CLIENT to retrieve its ThisModel metadata.

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

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

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

```
422 <dpws:ThisDevice ...>  
423   <dpws:FriendlyName xml:lang="..."? >xs:string</dpws:FriendlyName>+  
424   <dpws:FirmwareVersion>xs:string</dpws:FirmwareVersion>?  
425   <dpws:SerialNumber>xs:string</dpws:SerialNumber>?  
426   ...  
427 </dpws:ThisDevice>
```

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

429 dpws:ThisDevice/dpws:FriendlyName

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

432 dpws:ThisDevice/dpws:FirmwareVersion

433 Firmware version for this DEVICE. It MUST have fewer than MAX_FIELD_SIZE Unicode
434 characters.

435 dpws:ThisDevice/dpws:SerialNumber

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

438 CORRECT:

```
439 <dpws:ThisDevice  
440   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >  
441   <dpws:FriendlyName xml:lang="en-GB" >  
442     ACME ColourBeam Printer  
443   </dpws:FriendlyName>  
444   <dpws:FriendlyName xml:lang="en-US" >  
445     ACME ColorBeam Printer  
446   </dpws:FriendlyName>  
447 </dpws:ThisDevice>
```

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

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

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

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

455 CORRECT:

```
456 <soap:Envelope  
457   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"  
458   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"  
459   xmlns:wxs="http://schemas.xmlsoap.org/ws/2004/09/mex"  
460   xmlns:wsa="http://www.w3.org/2005/08/addressing" >  
461   <soap:Header>  
462     <wsa:Action>  
463       http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse  
464     </wsa:Action>  
465     <wsa:RelatesTo>  
466       urn:uuid:82204a83-52f6-475c-9708-174fa27659ec  
467     </wsa:RelatesTo>  
468     <wsa:To>  
469       http://www.w3.org/2005/08/addressing/anonymous  
470     </wsa:To>
```

```

471 </soap:Header>
472 <soap:Body>
473   <wsx:Metadata>
474     <wsx:MetadataSection
475       Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisModel"
476       >
477       <dpws:ThisModel>
478         <dpws:Manufacturer>ACME Manufacturing</dpws:Manufacturer>
479         <dpws:ModelName xml:lang="en-GB" >
480           ColourBeam 9
481         </dpws:ModelName>
482         <dpws:ModelName xml:lang="en-US" >
483           ColorBeam 9
484         </dpws:ModelName>
485       </dpws:ThisModel>
486     </wsx:MetadataSection>
487     <wsx:MetadataSection
488       Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/ThisDevice"
489       >
490       <dpws:ThisDevice>
491         <dpws:FriendlyName xml:lang="en-GB" >
492           ACME ColourBeam Printer
493         </dpws:FriendlyName>
494         <dpws:FriendlyName xml:lang="en-US" >
495           ACME ColorBeam Printer
496         </dpws:FriendlyName>
497       </dpws:ThisDevice>
498     </wsx:MetadataSection>
499
500     <!-- Other Metadata Sections omitted for brevity. -->
501
502   </wsx:Metadata>
503 </soap:Body>
504 </soap:Envelope>

```

505 Get [WS-Transfer] is the interoperable means for a CLIENT to retrieve the resource representation data
506 for a DEVICE – which includes the ThisDevice metadata for a DEVICE. A DEVICE may also provide
507 other means for a CLIENT to retrieve its ThisDevice metadata.

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

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

512 4.2 Hosting

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

```

515 <dpws:Relationship Type="xs:anyURI" ... >
516   (<dpws:Host>
517     <wsa:EndpointReference>endpoint-reference</wsa:EndpointReference>
518     <dpws:Types>list of xs:QName</dpws:Types>?
519     ...
520   </dpws:Host>)?
521   (<dpws:Hosted>
522     <wsa:EndpointReference>endpoint-reference</wsa:EndpointReference>+
523     <dpws:Types>list of xs:QName</dpws:Types>
524     <dpws:ServiceId>xs:anyURI</dpws:ServiceId>

```

```
525     ...
526     </dpws:Hosted>)*
527     ...
528 </dpws:Relationship>
```

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

530 dpws:Relationship

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

532 dpws:Relationship/@Type

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

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

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

540 dpws:Relationship/dpws:Host

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

543 dpws:Relationship/dpws:Host/wsa:EndpointReference

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

548 dpws:Relationship/dpws:Host/dpws:Types

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

551 dpws:Relationship/dpws:Hosted

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

557 dpws:Relationship/dpws:Hosted/wsa:EndpointReference

558 Endpoint References for a HOSTED SERVICE.

559 dpws:Relationship/dpws:Hosted/dpws:Types

560 Unordered set of Types implemented by a HOSTED SERVICE. All implemented Types SHOULD
561 be included.

562 dpws:Relationship/dpws:Hosted/dpws:ServiceId

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

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

567 CORRECT:

```
568 <dpws:Relationship
569   Type="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host"
570   xmlns:img="http://printer.example.org/imaging"
571   xmlns:wsa="http://www.w3.org/2005/08/addressing"
572   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01" >
573 <dpws:Hosted>
```

```

574 <wsa:EndpointReference>
575   <wsa:Address>http://172.30.184.244/print</wsa:Address>
576 </wsa:EndpointReference>
577 <dpws:Types>
578   img:PrintBasicPortType img:PrintAdvancedPortType
579 </dpws:Types>
580 <dpws:ServiceId>
581   http://printer.example.org/imaging/PrintService
582 </dpws:ServiceId>
583 </dpws:Hosted>
584 </dpws:Relationship>

```

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

587 No Identifier [WS-MetadataExchange] is defined for instances of the Relationship metadata format.

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

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

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

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

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

599 A DEVICE or HOSTED SERVICE may provide other means for a CLIENT to retrieve its relationship
600 metadata.

601 CORRECT:

```

602 <soap:Envelope
603   xmlns:gen="http://example.org/general"
604   xmlns:img="http://printer.example.org/imaging"
605   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
606   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
607   xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
608   xmlns:wsa="http://www.w3.org/2005/08/addressing" >
609   <soap:Header>
610     <wsa:Action>
611       http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
612     </wsa:Action>
613     <wsa:RelatesTo>
614       urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
615     </wsa:RelatesTo>
616     <wsa:To>
617       http://www.w3.org/2005/08/addressing/anonymous
618     </wsa:To>
619   </soap:Header>
620   <soap:Body>
621     <wsx:Metadata>
622       <wsx:MetadataSection
623         Dialect
624         ="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Relationship"
625       >
626     </dpws:Relationship

```

```

627     Type="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/host" >
628     <dpws:Hosted>
629         <wsa:EndpointReference>
630             <wsa:Address>http://172.30.184.244/print</wsa:Address>
631         </wsa:EndpointReference>
632         <wsa:EndpointReference>
633             <wsa:Address>http://[fdaa:23]/print1</wsa:Address>
634         </wsa:EndpointReference>
635         <dpws:Types>
636             img:PrintBasicPortType img:PrintAdvancedPortType
637         </dpws:Types>
638         <dpws:ServiceId>
639             http://printer.example.org/imaging/PrintService
640         </dpws:ServiceId>
641     </dpws:Hosted>
642     <dpws:Hosted>
643         <wsa:EndpointReference>
644             <wsa:Address>http://172.30.184.244/scan</wsa:Address>
645         </wsa:EndpointReference>
646         <wsa:EndpointReference>
647             <wsa:Address>http://[fdaa:24]/scan</wsa:Address>
648         </wsa:EndpointReference>
649         <dpws:Types>img:ScanBasicPortType</dpws:Types>
650         <dpws:ServiceId>
651             http://printer.example.org/imaging/ScanService
652         </dpws:ServiceId>
653     </dpws:Hosted>
654 </dpws:Relationship>
655 </wsx:MetadataSection>
656
657 <!-- Other Metadata Sections omitted for brevity. -->
658
659 </wsx:Metadata>
660 </soap:Body>
661 </soap:Envelope>

```

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

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

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

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

672 4.3 WSDL

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

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

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

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

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

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

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

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

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

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

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

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

702 GetMetadata [WS-MetadataExchange] is the interoperable means for a CLIENT to retrieve metadata for
703 a HOSTED SERVICE – which includes the WSDL for a HOSTED SERVICE. A HOSTED SERVICE may
704 provide other means for a CLIENT to retrieve its WSDL.

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

708 CORRECT:

```
709 <soap:Envelope
710   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
711   xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
712   xmlns:wsa="http://www.w3.org/2005/08/addressing" >
713   <soap:Header>
714     <wsa:Action>
715       http://schemas.xmlsoap.org/ws/2004/09/mex/GetMetadata/Response
716     </wsa:Action>
717     <wsa:RelatesTo>
718       urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
719     </wsa:RelatesTo>
720     <wsa:To>
721       http://www.w3.org/2005/08/addressing/anonymous
722     </wsa:To>
723   </soap:Header>
724   <soap:Body>
725     <wsx:Metadata>
726       <wsx:MetadataSection
727         Dialect="http://schemas.xmlsoap.org/wsdl" >
728         <wsx:MetadataReference>
729           <wsa:Address>http://172.30.184.244/print</wsa:Address>
```

```

730     <wsa:ReferenceParameters>
731         <x:Acme xmlns:x="urn:acme.com:webservices">
732             WSDL
733         </x:Acme>
734     </wsa:ReferenceParameters>
735     </wsx:MetadataReference>
736 </wsx:MetadataSection>
737
738     <!-- Other Metadata Sections omitted for brevity. -->
739
740 </wsx:Metadata>
741 </soap:Body>
742 </soap:Envelope>

```

743 4.4 WS-Policy

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

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

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

748 dpws:Profile

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

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

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

758 CORRECT:

```

759 <wsp:Policy
760     xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
761     xmlns:wsp="http://www.w3.org/ns/ws-policy" >
762     <dpws:Profile />
763 </wsp:Policy>

```

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

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

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

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

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

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

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

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

782 CORRECT:

```
783 <soap:Envelope
784   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
785   xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
786   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"
787   xmlns:wsp="http://www.w3.org/ns/ws-policy"
788   xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
789   xmlns:wsa="http://www.w3.org/2005/08/addressing" >
790 <soap:Header>
791   <wsa:Action>
792     http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
793   </wsa:Action>
794   <wsa:RelatesTo>
795     urn:uuid:82204a83-52f6-475c-9708-174fa27659ec
796   </wsa:RelatesTo>
797   <wsa:To>
798     http://www.w3.org/2005/08/addressing/anonymous
799   </wsa:To>
800 </soap:Header>
801 <soap:Body>
802   <wsx:Metadata>
803     <wsx:MetadataSection
804       Dialect="http://schemas.xmlsoap.org/wSDL/" >
805       <wSDL:definitions
806         targetNamespace="http://acme.example.com/colorbeam"
807         xmlns:image="http://printer.example.org/imaging" >
808         <wsp:Policy wsu:Id="DpPolicy" >
809           <dpws:Profile />
810         </wsp:Policy>
811
812         <!-- Other WSDL components omitted for brevity. -->
813
814         <wSDL:binding name="PrintBinding" type="image:PrintPortType" >
815           <wsp:PolicyReference URI="#DpPolicy"
816             wSDL:required="true" />
817           <!-- Other WSDL components omitted for brevity. -->
818         </wSDL:binding>
819       </wSDL:definitions>
820     </wsx:MetadataSection>
821
822     <!-- Other Metadata Sections omitted for brevity. -->
823
824   </wsx:Metadata>
825 </soap:Body>
826 </soap:Envelope>
```

827 5 Eventing

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

- 830 • [\[WS-Eventing\]](#)

831 5.1 Subscription

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

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

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

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

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

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

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

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

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

856 5.1.1 Filtering

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

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

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

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

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

875 CORRECT:

```

876 <soap:Envelope
877   xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
878   xmlns:wsa="http://www.w3.org/2005/08/addressing"
879   xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing" >
880 <soap:Header>
881   <wsa:Action>
882     http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
883   </wsa:Action>
884   <wsa:MessageID>
885     urn:uuid:314bea3b-03af-47a1-8284-f495497f1e33
886   </wsa:MessageID>
887   <wsa:ReplyTo>
888     <wsa:Address>
889       http://www.w3.org/2005/08/addressing/anonymous
890     </wsa:Address>
891   </wsa:ReplyTo>
892   <wsa:To>http://172.30.184.244/print</wsa:To>
893 </soap:Header>
894 <soap:Body>
895   <wse:Subscribe>
896     <wse:Delivery>
897       <wse:NotifyTo>
898         <wsa:Address>
899           urn:uuid:3726983d-02de-4d41-8207-d028ae92ce3d
900         </wsa:Address>
901       </wse:NotifyTo>
902     </wse:Delivery>
903     <wse:Expires>PT10M</wse:Expires>
904     <wse:Filter
905 Dialect="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action"
906   >
907     http://printer.example.org/imaging/PrintBasicPortType/JobEndState
908     http://printer.example.org/imaging/PrintBasicPortType/PrinterState
909   </wse:Filter>
910 </wse:Subscribe>
911 </soap:Body>
912 </soap:Envelope>
  
```

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

915 A HOSTED SERVICE must support filtering, at least by [action], so the Filtering Not Supported SOAP
 916 Fault is not appropriate.

917 To indicate that a HOSTED SERVICE does not expose any Notifications that would match the contents of
 918 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.)

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

923 **5.2 Subscription Duration and Renewal**

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

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

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

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

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

936 6 Security

937 This section defines a RECOMMENDED baseline for interoperable security between a DEVICE and a
938 CLIENT. A DEVICE (or CLIENT) is free to support other security mechanisms in place of this mechanism
939 as specified by WSDL [[WSDL 1.1](#)], policies [[WS-Policy](#)], or by other means.

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

944 A DEVICE may support at most one security profile.

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

- 947 • [[AES/TLS](#)]
- 948 • [[HTTP Authentication](#)]
- 949 • [[SHA](#)]
- 950 • [[TLS](#)]
- 951 • [[RFC 4122](#)]
- 952 • [[X.509.v3](#)]
- 953 • [[WS-Discovery](#)]

954 6.1 Terminology

955 Compact Signature

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

959 Secure Channel

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

964 6.2 Model

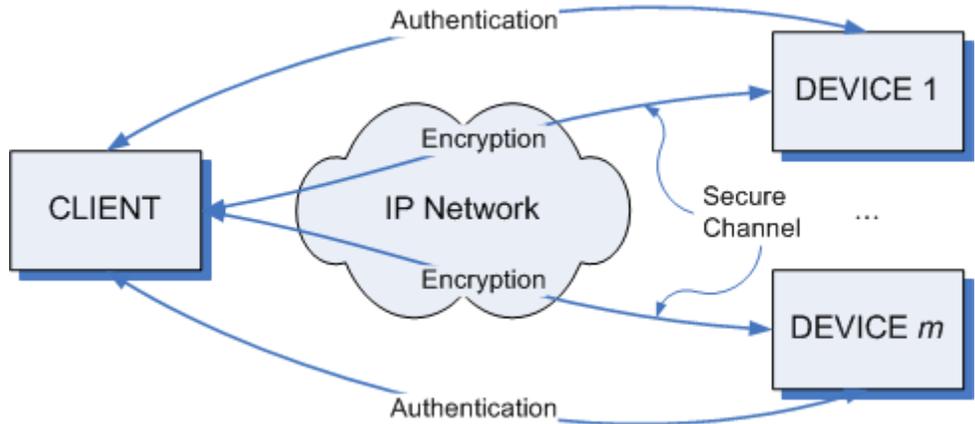
965 The security profile defined in this section has two parts: optional message-level signatures for UDP WS-
966 Discovery traffic, and mandatory transport-level encryption for metadata and control traffic.

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

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

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

978 A DEVICE uses TLS/SSL to secure its HTTP traffic, and HOSTED SERVICES may also use TLS/SSL to
 979 secure their HTTP traffic. A DEVICE may use a physical HTTPS address, or a logical address and
 980 HTTPS xAddrs. If a DEVICE and its HOSTED SERVICES are all reachable at the same address and
 981 port, a CLIENT and DEVICE may reuse a TLS/SSL connection for multiple operations.
 982



983
 984 The organization of CLIENT and DEVICE credentials, mechanism for provisioning them, and criteria for
 985 distinguishing valid and invalid credentials is out of scope of this profile.

986 6.3 Integrity

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

989	<i>R5015: If a SERVICE uses TLS/SSL or WS-Discovery Compact Signatures, it MUST provide Integrity (as defined in this section) for any TLS/SSL connections or signatures, respectively.</i>
991	<i>R4000: A SERVICE MUST not send a SOAP ENVELOPE without protecting the integrity of any Message Information Header blocks matching the following XPath expressions: (a) /soap:Envelope/soap:Header/wsa:Action, (b) /soap:Envelope/soap:Header/wsa:MessageID, (c) /soap:Envelope/soap:Header/wsa:To, (d) /soap:Envelope/soap:Header/wsa:ReplyTo, (e) /soap:Envelope/soap:Header/wsa:RelatesTo, and (f) /soap:Envelope/soap:Header/*[@isReferenceParameter='true'].</i>
997	<i>R4063: A SERVICE MAY reject a SOAP ENVELOPE that has unprotected Message Information Header blocks.</i>
999	<i>R4001: A SERVICE MUST not send a SOAP ENVELOPE (including SOAP Faults) without protecting the integrity of the SOAP ENVELOPE Body in conjunction with any Message Information Block(s) from R4000.</i>
1002	<i>R4064: A SERVICE MAY reject a SOAP ENVELOPE that does not protect the integrity of the SOAP ENVELOPE Body.</i>

1004 In this profile, the integrity of UDP discovery SOAP ENVELOPES is protected using message-level
 1005 signatures, while the integrity of other MESSAGES is protected using a Secure Channel.

1006 6.4 Confidentiality

1007 Confidentiality is the process by which sensitive information is protected against unauthorized disclosure
 1008 while in transit. Confidentiality MUST adhere to the following requirements:

1009	<i>R5016: If a SERVICE uses TLS/SSL, it MUST provide Confidentiality (as defined in this section) for any TLS/SSL connections.</i>
1011	<i>R4002: A SERVICE MUST NOT send a SOAP ENVELOPE without encrypting the SOAP ENVELOPE Body.</i>

1013 *R4067: A SERVICE MAY reject a SOAP ENVELOPE that does not encrypt the SOAP ENVELOPE Body.*

1014 In this profile, UDP WS-Discovery MESSAGES are not treated as confidential. Confidential MESSAGES
1015 are encrypted using a Secure Channel.

1016 **6.5 Authentication**

1017 Authentication is the process by which the identity of the sender is determined by the recipient.
1018 Authentication MUST adhere to the following requirements:

1019 *R4004: A SENDER MUST authenticate itself to a RECEIVER using credentials acceptable to the*
1020 *RECEIVER.*

1021 In this profile, authentication is done using certificates or a combination of certificates and HTTP
1022 authentication. If multicast messages are secured, the following additional requirements apply:

1023 *R4005: On multicast MESSAGES, a CLIENT MUST use an authentication credential that is suitable for all*
1024 *DEVICES that could legitimately process the multicast MESSAGE.*

1025 *R5023: If a SERVICE uses TLS/SSL, it MUST provide Authentication (as defined in this section) for any*
1026 *TLS/SSL connections.*

1027 **6.6 Trust**

1028 The distribution of the credentials needed for establishing the trust relationship is out of the scope of this
1029 profile.

1030 *R4008: A SERVICE MAY use additional mechanisms to verify the authenticity of the SENDER of any*
1031 *received MESSAGE by analyzing information provided by the lower networking layers.*

1032 For example, a SERVICE may authenticate only CLIENTs whose IP address exists in a preconfigured list.

1033 **6.7 DEVICE Behavior**

1034 *R4014: A DEVICE MAY require authentication of a CLIENT.*

1035 *R4017: A CLIENT MAY ignore MESSAGES received during discovery that have no signature or a*
1036 *nonverifiable signature.*

1037 *R4018: A DEVICE SHOULD cache authentication information for a CLIENT as valid as long as the*
1038 *DEVICE is connected to the CLIENT.*

1039 *R5009: If a DEVICE uses a physical transport address for the [address] property of its Endpoint*
1040 *Reference, it MUST be an HTTPS scheme IRI.*

1041 *R5010: A SERVICE MAY use an HTTP scheme IRI for the [address] property of its Endpoint Reference.*

1042 **6.8 Security for Discovery**

1043 In the discovery phase, the client learns of the existence of the device on the network. Subsequently, the
1044 identity of the device is verified, and the device is connected to the client.

1045 *R5011: A DEVICE SHOULD sign its UDP discovery traffic using WS-Discovery Compact Signatures [WS-*
1046 *Discovery] to provide CLIENTs with a mechanism to verify the integrity of the messages, and to*
1047 *authenticate the DEVICE as the signor of the messages.*

1048 WS-Discovery Compact Signatures use WS-Security [WS-Security] to generate a cryptographic signature
1049 that can be used by a CLIENT to verify the validity of the unencrypted message.

1050 In cases where CLIENTs persist enough information about the credentials and presence of security on a
1051 DEVICE to protect against impersonation, the DEVICE may not sign its discovery messages.

1052 *R5012: A DEVICE MUST NOT advertise HTTP scheme addresses the xAddrs fields of WS-Discovery*
1053 *messages.*

1054 **Probe**
 1055 A CLIENT initiates the discovery process by probing the network for a DEVICE it is interested in.

1056 *R4032: A DEVICE MUST NOT send a Probe Match SOAP ENVELOPE if the DEVICE is outside the local*
 1057 *subnet of the CLIENT, and the Probe SOAP ENVELOPE was sent using the multicast binding as*
 1058 *defined in WS-Discovery section 2.4.*

1059 *R4065: A CLIENT MUST discard a Probe Match SOAP ENVELOPE if it is received MATCH_TIMEOUT*
 1060 *seconds or more later than the last corresponding Probe SOAP ENVELOPE was sent.*

1061 **Resolve**

1062 *R4036: A DEVICE MUST NOT send a Resolve Match SOAP ENVELOPE if the DEVICE is outside the*
 1063 *local subnet of the CLIENT, and the Resolve SOAP ENVELOPE was sent using the multicast*
 1064 *binding as defined in WS-Discovery section 2.4*

1065 *R4066: A CLIENT MUST discard a Resolve Match SOAP ENVELOPE if it is received MATCH_TIMEOUT*
 1066 *seconds or more later than the last corresponding Resolve SOAP ENVELOPE was sent.*

1067 6.9 Authentication

1068 The authentication step that follows discovery verifies the credentials of the DEVICE and CLIENT in a
 1069 secure manner. Credentials may be cached on the DEVICE and/or CLIENT to simplify subsequent
 1070 authentications.

1071 6.9.1 Transport Layer Security (TLS/SSL)

1072 TLS/SSL provides mutual authentication of CLIENT and DEVICE as well as the establishment of a
 1073 Secure Channel over which MESSAGES are exchanged in a secure manner.

1074 *R4039: A CLIENT MUST initiate authentication with the DEVICE by setting up a TLS/SSL session.*

1075 *R4042: Following the establishment of a TLS/SSL Secure Channel, subsequent MESSAGE exchanges*
 1076 *over HTTP SHOULD use the existing TLS/SSL session.*

1077 6.9.2 Certificates

1078 *R4043: Each DEVICE SHOULD have its own, unique Certificate.*

1079 The Certificate contains information pertinent to the specific device including its public key. Typically,
 1080 certificates are issued by a trusted authority or a delegate (2nd tier) or a delegate of the delegate.

1081 *R4045: The format of the certificate MUST follow the common standard X.509v3.*

1082 An example of a self-signed X.509 certificate is shown below. in this case, the Subject field contains the
 1083 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	UUID	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

1084

1085 Certificate management is out of the scope of this profile.

1086 **6.9.3 DEVICE Authentication with TLS/SSL**

1087 X.509 certificates are the only mechanism for a CLIENT to authenticate a DEVICE or a HOSTED
1088 SERVICE (if TLS/SSL is supported on that HOSTED SERVICE).

1089 *R5017: If a SERVICE uses TLS/SSL, it MUST authenticate itself to a CLIENT by supplying an X.509v3*
1090 *certificate during the TLS/SSL handshake.*

1091 **6.9.4 CLIENT Authentication with TLS/SSL**

1092 *R4071: If the CLIENT and the SERVICE exchanged certificates during the TLS/SSL handshake, and the*
1093 *SERVICE as well as the CLIENT were able to verify the certificates, the CLIENT and SERVICE*
1094 *are mutually authenticated, and no further steps SHALL be required.*

1095 *R4046: A SERVICE MAY require HTTP Authentication step after the TLS/SSL handshake, if the*
1096 *SERVICE was not able to verify the certificate, or if the CLIENT did not provide a certificate*
1097 *during the TLS/SSL handshake.*

1098 X.509 certificates are the preferred mechanism for authenticating a client, but in cases where x.509 client
1099 certificates are unavailable or where validation is infeasible, the DEVICE may use HTTP Authentication to
1100 request client credentials.

1101 *R4048: If the HTTP authentication is successful, and the CLIENT presents a certificate to the SERVICE,*
1102 *the SERVICE SHOULD cache the certificate in its local certificate store of trusted certificates for*
1103 *future authentication of the CLIENT.*

1104 R4048 avoids the need for HTTP authentication for subsequent connections.

1105 6.9.5 CLIENT Authentication with HTTP Authentication

1106 HTTP authentication requires credentials in the form of username and password. It is assumed that how
1107 the CLIENT and SERVICE share knowledge of the username and password is out-of-band and beyond
1108 the scope of this profile.

1109 Because the authentication is performed over the Secure Channel established during TLS/SSL
1110 handshake and after the CLIENT has authenticated the SERVICE, HTTP Basic authentication may be
1111 used safely.

1112 *R4050: If a SERVICE requires HTTP authentication, the SERVICE SHALL challenge the CLIENT using*
1113 *the HTTP 401 response code.*

1114 *R4051: A CLIENT MUST authenticate using one of the options listed in the HTTP-Authenticate header.*

1115 *R4052: HTTP Authentication MUST use the following parameters for username and password of the*
1116 *HTTP Request: UserName, PIN / Password.*

1117 The UserName is supplied to the SERVICE during HTTP authentication and MAY be used for
1118 establishing multiple access control classes, such as administrators, users, and guests. The naming and
1119 use of UserName is implementation-dependent and out of the scope of this profile.

1120 *R4053: If no UserName is provided, "admin" SHALL be used as the default UserName.*

1121 The purpose of the PIN / Password is to authenticate the CLIENT to the DEVICE during the HTTP
1122 authentication.

1123 *R4054: The RECOMMENDED size of a PIN / Password is at least 8 characters using at least a 32*
1124 *character alphabet.*

1125 *R4055: The PIN / Password that is unique to the SERVICE SHALL be conveyed to the CLIENT out-of-*
1126 *band. The methods of conveying the PIN out-of-band are out of the scope of this profile.*

1127 *R4056: To reduce the attack surface, the SERVICE and CLIENT MAY limit the number of failed*
1128 *authentication attempts as well as the time interval successive attempts are made for one*
1129 *TLS/SSL session.*

1130 6.10 Secure Channel

1131 A Secure Channel at the transport level is used to secure traffic between CLIENT and SERVICE.

1132 *R4057: All secure communication for Description, Control, and Eventing between the CLIENT and*
1133 *SERVICE MUST use a Secure Channel.*

1134 *R4072: A SERVICE MUST support receiving and responding to a Probe SOAP ENVELOPE over HTTP*
1135 *using a Secure Channel.*

1136 *R4073: A SERVICE MAY ignore a Probe SOAP ENVELOPE sent over HTTP that does not use a Secure*
1137 *Channel.*

1138 *R5013: A CLIENT MAY use a Secure Channel to contact multiple SERVICES if they can be reached at*
1139 *the same address and port. As prescribed by R1015, a CLIENT may send a Probe over HTTP;*
1140 *this Probe and ProbeMatches are sent using the Secure Channel.*

1141 6.11 TLS/SSL Ciphersuites

1142 *R4059: It is the responsibility of the sender to convert the embedded URL to use HTTPS as different*
1143 *transport security mechanisms can be negotiated.*

1144 *R4060: A SERVICE MUST support the following TLS Ciphersuite: TLS_RSA_WITH_RC4_128_SHA.*

1145 *R4061: It is recommended that a SERVICE also support the following TLS Ciphersuite:*
1146 *TLS_RSA_WITH_AES_128_CBC_SHA.*

1147 *R4062: Additional Ciphersuites MAY be supported. They are negotiated during the TLS/SSL handshake.*

1148 Where appropriate, DEVICES are encouraged to support additional Ciphersuites that rely on more robust
1149 security technology, such as the SHA-2 [SHA] family of hashing standards.

1150 *R5014: A SERVICE SHOULD NOT negotiate any of the following TLS/SSL Ciphersuites: (a)*
1151 *TLS_RSA_WITH_NULL_SHA, (b) SSL_RSA_WITH_NULL_SHA, (c) any Ciphersuite with*
1152 *DH_anon in their symbolic name, (d) any Ciphersuites with MD5 in their symbolic name.*

1153 7 Conformance

1154 An endpoint MAY implement more than one of the roles defined herein. An endpoint is not compliant with
1155 this specification if it fails to satisfy one or more of the MUST or REQUIRED level requirements defined
1156 herein for the roles it implements.

1157 Normative text within this specification takes precedence over normative outlines, which in turn take
1158 precedence over the XML Schema [[XML Schema Part 1](#), [Part 2](#)] descriptions, which in turn take
1159 precedence over examples.

1160 A. Acknowledgements

1161 The following individuals have participated in the creation of this specification and are gratefully
1162 acknowledged:

1163 **Participants:**

1164 Geoff Bullen, Microsoft Corporation
1165 Steve Carter, Novell
1166 Dan Conti, Microsoft Corporation
1167 Doug Davis, IBM
1168 Scott deDeugd, IBM
1169 Dan Driscoll, Microsoft Corporation
1170 Colleen Evans, Microsoft Corporation
1171 Max Feingold, Microsoft Corporation
1172 Travis Grigsby, IBM
1173 Francois Jammes, Schneider Electric
1174 Ram Jeyaraman, Microsoft Corporation
1175 Mike Kaiser, IBM
1176 Supun Kamburugamuva, WSO2
1177 Devon Kemp, Canon Inc.
1178 Akira Kishida, Canon Inc.
1179 Mark Little, Red Hat
1180 Dr. Ingo Lueck, Technische Universitaet Dortmund
1181 Jonathan Marsh, WSO2
1182 Carl Mattocks
1183 Antoine Mensch
1184 Jaime Meritt, Progress Software
1185 Vipul Modi, Microsoft Corporation
1186 Anthony Nadalin, IBM
1187 Tadahiro Nakamura, Canon Inc.
1188 Masahiro Nishio, Canon Inc.
1189 Toby Nixon, Microsoft Corporation
1190 Shin Ohtake, Fuji Xerox Co., Ltd.
1191 Venkat Reddy, CA
1192 Alain Regnier, Ricoh Company, Ltd.
1193 Hitoshi Sekine, Ricoh Company, Ltd.
1194 Hiroshi Tamura, Ricoh Company, Ltd.
1195 Minoru Torii, Canon Inc.
1196 Asir S Vedamuthu, Microsoft Corporation
1197 David Whitehead, Lexmark International Inc.
1198 Don Wright, Lexmark International Inc.
1199 Prasad Yendluri, Software AG, Inc.
1200 Elmar Zeeb, University of Rostock
1201 Gottfried Zimmermann

1202

1203 **Co-developers of the initial contributions:**

1204 This document is based on initial contributions to the OASIS WS-DD Technical Committee by the follow
1205 co-developers:

1206 Shannon Chan, Microsoft Corporation
1207 Dan Conti, Microsoft Corporation
1208 Chris Kaler, Microsoft Corporation
1209 Thomas Kuehnel, Microsoft Corporation
1210 Alain Regnier, Ricoh Company Limited
1211 Bryan Roe, Intel Corporation

1212 Dale Sather, Microsoft Corporation
1213 Jeffrey Schlimmer, Microsoft Corporation (Editor)
1214 Hitoshi Sekine, Ricoh Company Limited
1215 Jorgen Thelin, Microsoft Corporation (Editor)
1216 Doug Walter, Microsoft Corporation
1217 Jack Weast, Intel Corporation
1218 Dave Whitehead, Lexmark International Inc.
1219 Don Wright, Lexmark International Inc.
1220 Yevgeniy Yarmosh, Intel Corporation

1221

B. Constants

1222 The following constants are used throughout this profile. The values listed below supersede other values
1223 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]

1224

C. Declaring Discovery Types in WSDL

1225 Solutions built on DPWS often define portTypes implemented by Hosted Services, and a discovery-layer
1226 portType implemented by the Host Service so the presence of these functional services can be
1227 determined at the discovery layer. The binding between a service-layer type and its discovery-layer type
1228 can be defined purely in descriptive text, but this appendix provides an optional mechanism to declare a
1229 discovery-layer type inside WSDL that can be consumed and understood by tools.

1230 This appendix defines an @dpws:DiscoveryType attribute to annotate the WSDL 1.1 portType [WSDL
1231 1.1] for the service-layer type. The normative outline for @dpws:DiscoveryType is:

```
1232 <wsdl:definitions ...>  
1233   [<wsdl:portType [dpws:DiscoveryType="xs:QName"? >  
1234     ...  
1235     </wsdl:portType>]*  
1236 </wsdl:definitions>
```

1237 The following describes additional, normative constraints to the outline listed above:

1238 /wsdl:definitions/wsdl:portType/@dpws:DiscoveryType

1239 The name of the portType to be advertised by the Host Service to indicate that this device
1240 supports the annotated Hosted Service portType.

1241 If omitted, no implied value

1242 This mechanism is only suitable in cases where a functional service type is bound to a single discovery-
1243 layer type, and authors of more complex type topologies may express the relationship between service
1244 and discovery types through normative text or through other means.

1245 Example usage follows. PrintDeviceType is the discovery-layer type for PrintPortType.

```
1246 <wsdl:definitions  
1247   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
1248   xmlns:dpws="http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01"  
1249   targetNamespace="http://printer.example.com/imaging"  
1250   xmlns:tns="http://printer.example.com/imaging">  
1251  
1252   <wsdl:portType name="PrintPortType"  
1253     dpws:DiscoveryType="tns:PrintDeviceType">  
1254  
1255     <!-- Contents omitted for brevity -->  
1256  
1257   </wsdl:portType>  
1258  
1259   <!-- Define PrintDeviceType to be empty -->  
1260   <wsdl:portType name="PrintDeviceType" />  
1261  
1262 </wsdl:definitions>
```

1263

D. Revision History

1264 [optional; should not be included in OASIS Standards]

1265

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/xAddr • 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

1266