



SOAP-over-UDP Version 1.1

Committee Draft 01

27 January 2009

Specification URIs:

This Version:

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

Previous Version:

N/A

Latest Version:

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

Ram Jeyaraman, Microsoft Corporation

Declared XML Namespace(s):

None.

Abstract:

This specification defines a binding for SOAP envelopes to use datagrams.

Status:

This document was last revised or approved by the 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 trademarks 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	4
1.1	Requirements	4
1.2	Notational Conventions.....	4
1.3	Terminology	5
1.4	XML Namespaces	5
1.5	Relationship to Web Service Specifications	5
1.6	Normative References	5
2	UDP Packet.....	7
2.1	Source Address and Port.....	7
2.2	Data Octets	7
3	Message Patterns.....	8
3.1	One-way.....	8
3.1.1	One-way Example	8
3.2	Request-response	8
3.2.1	Anonymous [reply endpoint].....	8
3.2.2	Request Example	9
3.2.3	Response Example	9
3.3	Multicast.....	10
3.4	Retransmission	10
4	Message Encoding	11
5	URI Scheme	12
5.1	Syntax	12
5.2	Semantics	12
6	Security Considerations	13
7	Conformance	14
A.	Appendix (non-normative) — Example retransmission algorithm.....	15
B.	Appendix (non-normative) — Example duplicate detection mechanisms.....	16
C.	Acknowledgements	17
D.	Revision History.....	19

1 Introduction

Many application protocol patterns match the semantics of the User Datagram Protocol (UDP) [RFC 768]

Some do not require the delivery guarantees of TCP while others make use of multicast transmission. In order to allow Web services to support these patterns, we need a way to map SOAP envelopes to user datagrams. This support is essential for services using WS-Discovery, where the use of multicast and need for low connection overhead makes UDP a natural choice. It is anticipated that other protocols will have similar requirements. This specification defines a binding of SOAP to user datagrams, including message patterns, addressing requirements, and security considerations.

1.1 Requirements

This specification intends to meet the following requirements:

- Support a one-way message-exchange pattern (MEP) where a SOAP envelope is carried in a user datagram.
- Support a request-response message-exchange pattern (MEP) where SOAP envelopes are carried in user datagrams.
- Support multicast transmission of SOAP envelopes carried in user datagrams.

Support both SOAP 1.1 [SOAP 1.1] and SOAP 1.2 [SOAP 1.2 Part 1]

Envelopes.

1.2 Notational Conventions

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

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

The syntax appears as an XML instance, but values in italics indicate data types instead of literal values.

Characters are appended to elements and attributes to indicate cardinality:

- "?" (0 or 1)
- "*" (0 or more)
- "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "[" and "]" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- Ellipses (i.e., "...") indicate points of extensibility. Additional children and/or attributes MAY be added at the indicated extension points but MUST NOT contradict the semantics of the parent and/or owner, respectively. If a receiver does not recognize an extension, the receiver SHOULD ignore the extension.
- XML namespace prefixes (see Table 13) are used to indicate the namespace of the element being defined.

Elsewhere in this specification, the characters "[" and "]" are used to call out references and property names. This specification uses the **[action]** and Fault properties **[WS-Addressing]**

to define faults.

41 1.3 Terminology

42 Receiver

43 The endpoint terminating a SOAP/UDP datagram

44 Sender

45 The endpoint originating a SOAP/UDP datagram

46 SOAP/UDP datagram

47 A user datagram containing a SOAP envelope in the data octets

48 User datagram

49 A User Datagram Protocol (UDP) packet

50 1.4 XML Namespaces

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

53 **Table 1: Prefix and XML Namespaces used in this specification.**

Prefix	XML Namespace	Specification(s)
s	(Either SOAP 1.1 or 1.2)	(Either SOAP 1.1 or 1.2)
s11	http://schemas.xmlsoap.org/soap/envelope/	[SOAP 1.1]
s12	http://www.w3.org/2003/05/soap-envelope	[SOAP 1.2]
a	http://schemas.xmlsoap.org/ws/2004/08/addressing	[WS-Addressing]
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	[WS-Security]

54 1.5 Relationship to Web Service Specifications

55 This specification provides a binding appropriate for:

- 56 • SOAP 1.1 [SOAP 1.1]

57 SOAP 1.2 [SOAP 1.2 Part 1]

58 Messages conforming to either SOAP specification can use this binding. This specification relies on WS-
59 Addressing [WS-Addressing]

60

61 1.6 Normative References

62 [RFC 768]

63 J. Postel, "User Datagram Protocol," RFC 768, August 1980.

64 [RFC 2119]

65 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, March
66 1997.

67 [RFC 2365]

68 D. Meyer, "Administratively Scoped IP Multicast," RFC 2365, July 1998.

69 [RFC 2396]

- 70 T. Berners-Lee, et al, "Uniform Resource Identifiers (URI): Generic Syntax," [RFC 2396](#), August
71 1998.
- 72 **[SOAP 1.1]**
- 73 D. Box, et al, "[Simple Object Access Protocol \(SOAP\) 1.1](#)," May 2000.
- 74 **[SOAP 1.2 Part 1]**
- 75 M. Gudgin, et al, "[SOAP Version 1.2 Part 1: Messaging Framework](#)," June 2003.
- 76 **[WS-Addressing]**
- 77 D. Box, et al, "[Web Services Addressing \(WS-Addressing\)](#)," August 2004.
- 78 **[WS-Security]**
- 79 A. Nadalin, et al, "[Web Services Security: SOAP Message Security V1.0](#)," March 2004.
- 80 **[XML 1.0]**
- 81 T. Bray, et al, "[Extensible Markup Language \(XML\) 1.0 \(Second Edition\)](#)," October 2000.

82 2 UDP Packet

83 Except as noted explicitly below, this specification does not constrain RFC 768 [**RFC 768**]

84

85 2.1 Source Address and Port

86 For security reasons, the source address **MUST** be supplied at the UDP packet level and **MUST** be the
87 IPv4 or IPv6 address of the sender; the receiver **SHOULD** reject SOAP/UDP datagrams that have
88 inappropriate values for the source address.

89 A source port **MAY** be specified. If a source port is not specified then the source port is assumed to be
90 the default port for the SOAP-over-UDP protocol.

91 2.2 Data Octets

92 The data octets **MUST** contain a SOAP envelope [**SOAP 1.1**][**SOAP 1.2 Part 1**]

93 The SOAP envelope **MUST** fit within a single datagram, that is it **MUST** be small enough that the overall
94 datagram is less than 65,536 (2^{16}) octets.

95 The SOAP envelope **MUST** use the mechanisms defined in WS-Addressing [**WS-Addressing**]

96

97 3 Message Patterns

98 This specification supports the following message patterns:

- 99 • Unicast one-way
- 100 • Multicast one-way
- 101 • Unicast request, unicast response
- 102 • Multicast request, unicast response
- 103 as detailed in the rest of this section.

104 3.1 One-way

105 This specification uses the constructs in WS-Addressing for one-way messages. The one-way message
106 is sent in a user datagram.

107 3.1.1 One-way Example

```
108 (001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"  
109     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" >  
110 (002) <S:Header>  
111 (003)   <wsa:To>http://fabrikam.com/Server</wsa:To>  
112 (004)   <wsa:Action>http://fabrikam.com/Probe</wsa:Action>  
113 (005)   <wsa:MessageId>  
114         urn:uuid:1da72f1a-5546-493c-934c-a9e3577e206a  
115         </wsa:MessageId>  
116 (006) </S:Header>  
117 (007) <S:Body>  
118 (008)   ...  
119 (009) </S:Body>  
120 (010) </S:Envelope>
```

121 This example shows a one-way SOAP message. Lines 001-002 are standard SOAP elements. Lines 003-
122 005 specify various WS-Addressing headers. Note that despite the fact that the **[destination]** for the
123 message is specified using a URI that uses the http scheme, the message is still transmitted over UDP.
124 Lines 006-010 show standard SOAP elements.

125 3.2 Request-response

126 This specification uses the constructs in WS-Addressing for request and response messages. The
127 request message is sent in one user datagram, response messages are sent in separate user datagrams.

128 3.2.1 Anonymous [reply endpoint]

129 WS-Addressing defines a URI, "http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous",
130 that can appear in the **[address]** property of an endpoint reference. If the **[reply endpoint]** property of a
131 SOAP message transmitted over UDP has an **[address]** property with this value, the UDP source
132 address (and source port) is considered to be the address to which reply messages should be sent.

133 The implied value of the **[reply endpoint]** property for SOAP messages transmitted over UDP is an
134 endpoint reference with an **[address]** property whose value is
135 "http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous".

136 3.2.2 Request Example

```
137 (001) <S:Envelope xmlns:S=http://www.w3.org/2003/05/soap-envelope
138       xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" >
139 (002) <S:Header>
140 (003)   <wsa:To>http://fabrikam.com/Server</wsa:To>
141 (004)   <wsa:Action>http://fabrikam.com/Probe</wsa:Action>
142 (005)   <wsa:MessageId>
143         urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
144     </wsa:MessageId>
145 (006)   <wsa:ReplyTo>
146         <wsa:Address>
147           http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
148         </wsa:Address>
149       </wsa:ReplyTo>
150 (007) </S:Header>
151 (008) <S:Body>
152 (009)   ...
153 (010) </S:Body>
154 (011) </S:Envelope>
```

155 This example shows a request SOAP message. Lines 001-002 are standard SOAP elements. Lines 003-
156 005 specify various WS-Addressing headers. Note that despite the fact that the **[destination]** for the
157 message is specified using a URI that uses the http scheme, the message is still transmitted over UDP.
158 Line 6 shows a **[reply endpoint]** header specifying the anonymous URI (see section 4.2.1). Lines 007-
159 011 show standard SOAP elements.

160 3.2.3 Response Example

```
161 (001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
162       xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" >
163 (002) <S:Header>
164 (003)   <wsa:To>
165         http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
166     </wsa:To>
167 (004)   <wsa:Action>http://fabrikam.com/ProbeMatch</wsa:Action>
168 (005)   <wsa:MessageId>
169         urn:uuid:5a6ed11a-7a80-409a-82bf-43c4c5092911
170     </wsa:MessageId>
171 (006)   <wsa:RelatesTo>
172         urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
173     </wsa:RelatesTo>
174 (007) </S:Header>
175 (008) <S:Body>
176 (009)   ...
177 (010) </S:Body>
178 (011) </S:Envelope>
```

179 This example shows a response SOAP message. Lines 001-002 are standard SOAP elements. Lines
180 003-005 specify various WS-Addressing headers. Note that the **[destination]** for the message is
181 specified as the anonymous URI. Line 006 shows a **[relationship]** header indicating that this message is
182 a reply to the example message in Section 4.2.2. Lines 007-011 show standard SOAP elements.

183 **3.3 Multicast**

184 The message patterns defined above can be used with unicast or multicast transmission of UDP
185 datagrams with the following restriction: The response in a request-response message pattern **MUST**
186 **NOT** be multicast.

187 Note that in the case of a multicast request, unicast response MEP, the sender of the request might
188 receive multiple responses.

189 Multicast SOAP/UDP datagrams **SHOULD** be scoped to ensure they are not forwarded beyond the
190 boundaries of the administrative system. This **MAY** be done with either TTL or administrative scopes
191 **[RFC 2365]**

192 depending on what is implemented in the network. If TTL is used it is **RECOMMENDED** that the TTL
193 value be set to 1.

194 The destination IP address of a multicast message **MUST** be a multicast group.

195 **3.4 Retransmission**

196 To avoid repeated packet collisions, implementation retransmission **SHOULD** observe good practices
197 such as using exponential back-off algorithms and spreading. An implementation **MAY** use the algorithm
198 defined in Appendix A. If a message is to be retransmitted it **MUST** have a **[message id]** property. For
199 each transmission of such a message, the value of the **[message id]** property **MUST** be the same.

200 **4 Message Encoding**

201 The algorithm defined in Appendix F of XML 1.0 [[XML 1.0](#)] should be used to determine whether a
202 message is encoded as XML. If use of said algorithm does not result in an XML serialization, the
203 encoding is undefined.

204 5 URI Scheme

205 This section defines a URI scheme for UDP endpoints. The scheme allows hostname and port to be
206 specified. Resolving such a URI provides the information needed to send messages to a UDP endpoint
207 per the protocol defined in this document.

208 5.1 Syntax

209 The syntax of the URI scheme is as follows:

```
210 soap.udp: // <host> [ : <port> ] [ / <rel_path> ] [ ? <query> ]
```

211 The syntax and interpretation of the host, port, rel_path and query portions is as defined in RFC 2396
212 [RFC 2396]

213

214 5.2 Semantics

215 The semantics of resolving a soap.udp URI are as follows:

- 216 1. Use the port portion as the port number, if specified, otherwise use the default port as the port
217 number.
- 218 2. Resolve the host portion to an IP address.
- 219 3. Using the message protocol defined in this document, send a message to the IP address determined
220 in step 2 using the port number determined in step 1.

221 6 Security Considerations

222 It is recommended that all messages be secured using the mechanisms described in **[WS-Security]**
223 to prevent tampering or falsification.

224 All critical headers, such as those described in **[WS-Addressing]**
225 , and the message body, need to be included in signatures to bind all parts of the message together.

226 Recipients should verify that the sender has the right to speak for the specified source or response
227 location (if one is provided).

228 Messages should be accepted and processed only from trusted sources (either directly trusted or
229 indirectly trusted via third parties).

230 The UDP packet size introduces a challenge for secure messages due to its limited size. For this reason
231 it is recommended that security tokens not be passed but referenced using the Key Identifier mechanisms
232 described in **[WS-Security]**

233

234 **7 Conformance**

235 An implementation is not conformant with this specification if it fails to satisfy one or more of the MUST or
236 REQUIRED level requirements defined herein.

237 **A. Appendix (non-normative) — Example**
238 **retransmission algorithm**

239 Constants referenced within the algorithm are defined in Table 1 (for unicast messages) and Table 2 (for
240 unreliable multicast messages).

241 **Retry and back-off algorithm.**

- 242 1. Transmit; *_UDP_REPEAT--;
243 2. If *_UDP_REPEAT <= 0 goto Step 9;
244 3. Generate a random number T in [UDP_MIN_DELAY .. UDP_MAX_DELAY];
245 4. Wait T milliseconds;
246 5. Transmit; *_UDP_REPEAT--;
247 6. If *_UDP_REPEAT <= 0 goto Step 9;
248 7. $T = T * 2$; If $T > \text{UDP_UPPER_DELAY}$ then $T = \text{UDP_UPPER_DELAY}$;
249 8. goto 4
250 9. Done.

251 **Table 1: Protocol Retry and back-off constants for unicast messages**

Constant / Message	Value
UNICAST_UDP_REPEAT	2
UDP_MIN_DELAY	50
UDP_MAX_DELAY	250
UDP_UPPER_DELAY	500

252 **Table 2: Protocol Retry and back-off constants for unreliable multicast messages**

Constant / Message	Value
MULTICAST_UDP_REPEAT	4
UDP_MIN_DELAY	50
UDP_MAX_DELAY	250
UDP_UPPER_DELAY	500

253

254 **B. Appendix (non-normative) — Example duplicate**
255 **detection mechanisms**

256 A receiver keeps a list of the last n messages received along with their **[message id]** properties **[WS-**
257 **Addressing]**

258 1. . When a new (non-duplicate) message arrives, the oldest message is removed from the list.

259 A receiver tracks all messages received in the last x milliseconds along with their **[message id]** property
260 **[WS-Addressing]**

261 2. . Messages received more than x milliseconds ago are removed from the list.

262 For both approaches any message arriving with a **[message id]** property identical to one of those the
263 receiver has in its list is a duplicate. Messages with unique values for the **[message id]** property are not
264 duplicates.

265 The timestamp specified in the Security header block **[WS-Security]**

266 MAY be used to limit the duration for which **[message id]** properties need to be remembered.

267

C. Acknowledgements

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

270 **Participants:**

271 Geoff Bullen, Microsoft Corporation
272 Steve Carter, Novell
273 Dan Conti, Microsoft Corporation
274 Doug Davis, IBM
275 Scott deDeugd, IBM
276 Dan Driscoll, Microsoft Corporation
277 Colleen Evans, Microsoft Corporation
278 Max Feingold, Microsoft Corporation
279 Travis Grigsby, IBM
280 Francois Jammes, Schneider Electric
281 Ram Jeyaraman, Microsoft Corporation
282 Mike Kaiser, IBM
283 Supun Kamburugamuva, WSO2
284 Devon Kemp, Canon Inc.
285 Akira Kishida, Canon Inc.
286 Mark Little, Red Hat
287 Dr. Ingo Lueck, Technische Universitaet Dortmund
288 Jonathan Marsh, WSO2
289 Carl Mattocks
290 Antoine Mensch
291 Jaime Meritt, Progress Software
292 Vipul Modi, Microsoft Corporation
293 Anthony Nadalin, IBM
294 Tadahiro Nakamura, Canon Inc.
295 Masahiro Nishio, Canon Inc.
296 Toby Nixon, Microsoft Corporation
297 Shin Ohtake, Fuji Xerox Co., Ltd.
298 Venkat Reddy, CA
299 Alain Regnier, Ricoh Company, Ltd.
300 Hitoshi Sekine, Ricoh Company, Ltd.
301 Hiroshi Tamura, Ricoh Company, Ltd.
302 Minoru Torii, Canon Inc.
303 Asir S Vedomuthu, Microsoft Corporation
304 David Whitehead, Lexmark International Inc.
305 Don Wright, Lexmark International Inc.
306 Prasad Yendluri, Software AG, Inc.
307 Elmar Zeeb, University of Rostock
308 Gottfried Zimmermann

309

310 **Co-authors of the initial contributions:**

311 This document is based on initial contributions to the OASIS WS-DD Technical Committee by the
312 following co-authors.

313 Harold Combs, Lexmark International Inc.
314 Martin Gudgin (editor), Microsoft Corporation
315 John Justice, Microsoft Corporation
316 Gopal Kakivaya, Microsoft Corporation
317 David Lindsey, Lexmark International Inc.
318 David Orchard, BEA Systems, Inc.

319 Alain Regnier, Ricoh Company Limited
320 Jeffrey Schlimmer, Microsoft Corporation
321 Stacy Simpson, Lexmark International Inc.
322 Hiroshi Tamura, Ricoh Company Limited
323 Don Wright, Lexmark International Inc.
324 Kenny Wolf, Microsoft Corporation
325

326 **Acknowledgements of the initial contributions:**

327 This specification has been developed as a result of joint work with many individuals and teams,
328 including:

329
330 Erik Christensen, Microsoft Corporation
331 David Langworthy, Microsoft Corporation
332 Yaniv Pessach, Microsoft Corporation
333 Stefan Pharies, Microsoft Corporation
334 Sam Rhodus, Lexmark International Inc.
335 Jerry Thrasher, Lexmark International Inc.
336 Mike Vernal, Microsoft Corporation
337 Elliot Waingold, Microsoft Corporation
338 Dave Whitehead, Lexmark International Inc.

339 **D. Revision History**

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

341

Revision	Date	Editor	Changes Made
wd-01	09/16/2008	Ram Jeyaraman	Created the initial working draft by converting the input specification to OASIS template.
wd-02	09/29/2008	Ram Jeyaraman	Updated document identifier, added co-chair and editor names, use of urn:uuid (issue 50) in examples.

342