SOAP-over-UDP Version 1.1

Committee Draft 02

27 January 2009

Specification URIs:

This Version:

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

Previous 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
http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/cd-01/wsdd-soapoverudp-1.1-spec-cd-01.pdf

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

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

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 1) 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.

1.3 Terminology

Receiver

The endpoint terminating a SOAP/UDP datagram

Sender

The endpoint originating a SOAP/UDP datagram

SOAP/UDP datagram

A user datagram containing a SOAP envelope in the data octets

User datagram

A User Datagram Protocol (UDP) packet

1.4 XML Namespaces

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

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 Part 1**]
A	http://www.w3.org/2005/08/addressing	[**WS-Addressing**]

1.5 Relationship to Web Service Specifications

This specification provides a binding appropriate for:

- SOAP 1.1 [**SOAP 1.1**]

SOAP 1.2 [**SOAP 1.2 Part 1**]

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

.

1.6 Normative References

[**RFC 768**]

J. Postel, "User Datagram Protocol," RFC 768, http://www.ietf.org/rfc/rfc768.txt, August 1980.

[**RFC 2119**]

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

[**RFC 2365**]

D. Meyer, "Administratively Scoped IP Multicast," RFC 2365, http://www.ietf.org/rfc/rfc2365.txt, July 1998.

[**RFC 3986**]

T. Berners-Lee, et al, "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 3986, http://www.ietf.org/rfc/rfc3986.txt, January 2005.

[RFC 791]

"Internet Protocol (IPv4)", IETF RFC 791, http://www.ietf.org/rfc/rfc791.txt, September 1981.

[RFC 2460]

S. Deering, et al, "Internet Protocol, Version 6 (IPv6) Specification", IETF RFC 2460, http://www.ietf.org/rfc/rfc2460.txt, December 1998.

[**SOAP 1.1**]

W3C Note, "Simple Object Access Protocol (SOAP) 1.1", http://www.w3.org/TR/2000/NOTE-SOAP-20000508, 08 May 2000.

[**SOAP 1.2 Part 1**]

W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", http://www.w3.org/TR/2007/REC-soap12-part1-20070427, April 2007.

[**WS-Addressing**]

W3C Recommendation, "Web Services Addressing 1.0 - Core", http://www.w3.org/TR/2006/REC-ws-addr-core-20060509, 9 May 2006.

[**WS-Security**]

OASIS Standard, "Web Services Security Core specification 1.1", http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, February 2006.

[**XML 1.0**]

W3C Recommendation, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", http://www.w3.org/TR/2006/REC-xml-20060816, 16 August 2006.

2 UDP Packet

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

2.1 Source Address and Port

For security reasons, the source address MUST be supplied at the IP packet level and MUST be the IPv4 [**RFC 791**] address (including but not limited to unicast, multicast, and broadcast addresses) or IPv6 [**RFC 2460**] address (including but not limited to unicast and multicast addresses) of the sender; the receiver SHOULD reject IP packets containing a SOAP/UDP datagram that have inappropriate values for the source address.

2.2 Data Octets

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

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

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

3 Message Patterns

This specification supports the following message patterns:

- Unicast one-way
- Multicast one-way
- Unicast request, unicast response
- Multicast request, unicast response

as detailed in the rest of this section.

This specification uses the constructs **[action]**, **[destination]**, **[message id]**, **[reply endpoint]**, **[address]** in WS-Addressing [**WS-Addressing**]

. SOAP messages transmitted over UDP MUST have a **[message id]** property**.**

3.1 One-way

The one-way message is sent in a user datagram.

3.1.1 One-way Example

```
(001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
        xmlns:wsa="http://www.w3.org/2005/08/addressing" >
(002)    <S:Header>
(003)        <wsa:To>http://fabrikam.com/Server</wsa:To>
(004)        <wsa:Action>http://fabrikam.com/Probe</wsa:Action>
(005)        <wsa:MessageID>
                      urn:uuid:1da72f1a-5546-493c-934c-a9e3577e206a
             </wsa:MessageID>
(006)    </S:Header>
(007)    <S:Body>
(008)        ...
(009)    </S:Body>
(010) </S:Envelope>
```

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

3.2 Request-response

The request message is sent in one user datagram and the corresponding response message is sent in another user datagram.

3.2.1 Anonymous [reply endpoint]

WS-Addressing defines a URI, "http://www.w3.org/2005/08/addressing/anonymous", that can appear in the **[address]** property of an endpoint reference. If the **[reply endpoint]** property of a SOAP message transmitted over UDP has an **[address]** property with this value, the UDP source address (and source port) is considered to be the address to which reply messages should be sent.

The implied value of the **[reply endpoint]** property for SOAP messages transmitted over UDP is an endpoint reference with an **[address]** property whose value is "http://www.w3.org/2005/08/addressing/anonymous".

3.2.2 Request Example 1

```
(001) <S:Envelope xmlns:S=http://www.w3.org/2003/05/soap-envelope
        xmlns:wsa="http://www.w3.org/2005/08/addressing" >
(002)    <S:Header>
(003)       <wsa:To>http://fabrikam.com/Server</wsa:To>
(004)       <wsa:Action>http://fabrikam.com/Probe</wsa:Action>
(005)       <wsa:MessageID>
                    urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
            </wsa:MessageID>
(006)       <wsa:ReplyTo>
               <wsa:Address>
                    http://www.w3.org/2005/08/addressing/anonymous
               </wsa:Address>
            </wsa:ReplyTo>
(007)    </S:Header>
(008)    <S:Body>
(009)       ...
(010)    </S:Body>
(011) </S:Envelope>
```

This example shows a request SOAP message. Lines 001-002 are standard SOAP elements. Lines 003-005 specify various WS-Addressing headers. Note that despite the fact that the **[destination]** for the message is specified using a URI that uses the http scheme, the message is still transmitted over UDP. Line 6 shows a **[reply endpoint]** header specifying the anonymous URI (see Section 3.2.1). Lines 007-011 show standard SOAP elements.

3.2.3 Response Example 1

```
(001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
        xmlns:wsa="http://www.w3.org/2005/08/addressing" >
(002)    <S:Header>
(003)       <wsa:To>
                    http://www.w3.org/2005/08/addressing/anonymous
            </wsa:To>
(004)       <wsa:Action>http://fabrikam.com/ProbeMatch</wsa:Action>
(005)       <wsa:MessageID>
                    urn:uuid:5a6ed11a-7a80-409a-82bf-43c4c5092911
            </wsa:MessageID>
(006)       <wsa:RelatesTo>
                    urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
            </wsa:RelatesTo>
(007)    </S:Header>
(008)    <S:Body>
(009)       ...
(010)    </S:Body>
(011) </S:Envelope>
```

This example shows a response SOAP message. Lines 001-002 are standard SOAP elements. Lines 003-005 specify various WS-Addressing headers. Note that the **[destination]** for the message is specified as the anonymous URI. Line 006 shows a **[relationship]** header indicating that this message is a reply to the example message in Section 3.2.2. Lines 007-011 show standard SOAP elements.

3.2.4 Request Example 2

```
(001) <S:Envelope xmlns:S=http://www.w3.org/2003/05/soap-envelope
        xmlns:wsa="http://www.w3.org/2005/08/addressing" >
(002)    <S:Header>
(003)       <wsa:To>soap.udp://fabrikam1.com:54321/Server</wsa:To>
(004)       <wsa:Action>http://fabrikam1.com/Probe</wsa:Action>
(005)       <wsa:MessageID>
                      urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
            </wsa:MessageID>
(006)       <wsa:ReplyTo>
               <wsa:Address>
                      soap.udp://fabrikam2.com:54322/Client
               </wsa:Address>
            </wsa:ReplyTo>
(007)    </S:Header>
(008)    <S:Body>
(009)       ...
(010)    </S:Body>
(011) </S:Envelope>
```

This example shows a request SOAP message. Lines 001-002 are standard SOAP elements. Lines 003-005 specify various WS-Addressing headers. Note that the **[destination]** for the message is specified using a URI that uses the soap.udp scheme. Line 6 shows a **[reply endpoint]** header containing an addressable URI that uses the soap.udp scheme. Lines 007-011 show standard SOAP elements.

3.2.5 Response Example 2

```
(001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
        xmlns:wsa="http://www.w3.org/2005/08/addressing" >
(002)    <S:Header>
(003)       <wsa:To>
                      soap.udp://fabrikam2.com:54322/Client
            </wsa:To>
(004)       <wsa:Action>http://fabrikam.com/ProbeMatch</wsa:Action>
(005)       <wsa:MessageID>
                      urn:uuid:5a6ed11a-7a80-409a-82bf-43c4c5092911
            </wsa:MessageID>
(006)       <wsa:RelatesTo>
                      urn:uuid:9ceada16-2403-4404-a8cc-60799acd9d1c
            </wsa:RelatesTo>
(007)    </S:Header>
(008)    <S:Body>
(009)       ...
(010)    </S:Body>
(011) </S:Envelope>
```

This example shows a response SOAP message. Lines 001-002 are standard SOAP elements. Lines 003-005 specify various WS-Addressing headers. Note that the **[destination]** for the message contains an addressable URI that uses the soap.udp scheme. Line 006 shows a **[relationship]** header indicating that this message is a reply to the example message in Section 3.2.4. Lines 007-011 show standard SOAP elements.

3.3 Multicast

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

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

Multicast SOAP/UDP datagrams SHOULD be scoped to ensure they are not forwarded beyond the boundaries of the administrative system. For IPv4, this can be done with either time-to-live (TTL) field or administrative scopes [**RFC 2365**]

depending on what is implemented in the network. For IPv6, this can be done by setting the hop-limit field. If either IPv4 TTL or IPv6 hop-limit is used, it is RECOMMENDED that its value be set to 1.

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

3.4 Retransmission

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

4 Message Encoding

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

5 URI Scheme

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

5.1 Syntax

The syntax of the URI scheme is as follows:

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

The syntax and interpretation of the host, port, rel_path and query portions is as defined in RFC 3986 [**RFC 3986**]. **Error! Reference source not found.**

5.2 Semantics

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

1. Use the port portion as the port number.
2. Resolve the host portion to an IP address.
3. Using the message protocol defined in this document, send a message to the IP address determined in step 2 using the port number determined in step 1.

6 Security Considerations

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

All critical headers, such as those described in [**WS-Addressing**]

, and the message body, need to be included in signatures to bind all parts of the message together.

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

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

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

SOAP messages containing a **[reply endpoint]** property transmitted over UDP MAY be rejected by a recipient due to security concerns such as distributed denial-of-service attacks.

7 Conformance

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

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

Constants referenced within the algorithm are defined in Table 1 (for unicast messages) and Table 2 (for unreliable multicast messages). The value of those constants (as defined in Table 1 and Table 2) is non-normative. Implementations and other specifications may override the value of those constants.

Retry and back-off algorithm.

```
1.  Transmit;
2.  if *_UDP_REPEAT <= 0 go to Step 11;
3.  else *_UDP_REPEAT--;
4.  Generate a random number T in [UDP_MIN_DELAY .. UDP_MAX_DELAY];
5.  Wait T milliseconds;
6.  Retransmit;
7.  if *_UDP_REPEAT <= 0 goto Step 11;
8.  else *_UDP_REPEAT--;
9.  T = T * 2;If T > UDP_UPPER_DELAY then T = UDP_UPPER_DELAY;
10. go to 5;
11. Done.
```

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

Constant / Message	Value
UNICAST_UDP_REPEAT	1
UDP_MIN_DELAY	50
UDP_MAX_DELAY	250
UDP_UPPER_DELAY	500

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

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

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

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

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

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

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

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

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

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

C. Acknowledgements

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

Participants:

Geoff Bullen, Microsoft Corporation
Steve Carter, Novell
Dan Conti, Microsoft Corporation
Doug Davis, IBM
Scott deDeugd, IBM
Dan Driscoll, Microsoft Corporation
Colleen Evans, Microsoft Corporation
Max Feingold, Microsoft Corporation
Travis Grigsby, IBM
Francois Jammes, Schneider Electric
Ram Jeyaraman, Microsoft Corporation
Mike Kaiser, IBM
Supun Kamburugamuva, WSO2
Devon Kemp, Canon Inc.
Akira Kishida, Canon Inc.
Mark Little, Red Hat
Dr. Ingo Lueck, Technische Universitaet Dortmund
Jonathan Marsh, WSO2
Carl Mattocks
Antoine Mensch
Jaime Meritt, Progress Software
Vipul Modi, Microsoft Corporation
Anthony Nadalin, IBM
Tadahiro Nakamura, Canon Inc.
Masahiro Nishio, Canon Inc.
Toby Nixon, Microsoft Corporation
Shin Ohtake, Fuji Xerox Co., Ltd.
Venkat Reddy, CA
Alain Regnier, Ricoh Company, Ltd.
Hitoshi Sekine, Ricoh Company, Ltd.
Hiroshi Tamura, Ricoh Company, Ltd.
Minoru Torii, Canon Inc.
Asir S Vedamuthu, Microsoft Corporation
David Whitehead, Lexmark International Inc.
Don Wright, Lexmark International Inc.
Prasad Yendluri, Software AG, Inc.
Elmar Zeeb, University of Rostock
Gottfried Zimmermann

Co-authors of the initial contributions:

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

Harold Combs, Lexmark International Inc.
Martin Gudgin (editor), Microsoft Corporation
John Justice, Microsoft Corporation
Gopal Kakivaya, Microsoft Corporation
David Lindsey, Lexmark International Inc.
David Orchard, BEA Systems, Inc.

Alain Regnier, Ricoh Company Limited
Jeffrey Schlimmer, Microsoft Corporation
Stacy Simpson, Lexmark International Inc.
Hiroshi Tamura, Ricoh Company Limited
Don Wright, Lexmark International Inc.
Kenny Wolf, Microsoft Corporation

Acknowledgements of the initial contributions:

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

Erik Christensen, Microsoft Corporation
David Langworthy, Microsoft Corporation
Yaniv Pessach, Microsoft Corporation
Stefan Pharies, Microsoft Corporation
Sam Rhodus, Lexmark International Inc.
Jerry Thrasher, Lexmark International Inc.
Mike Vernal, Microsoft Corporation
Elliot Waingold, Microsoft Corporation
Dave Whitehead, Lexmark International Inc.

D. Revision History

[optional; should not be included in OASIS Standards]

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.
wd-03	1/15/2008	Ram Jeyaraman	Created working draft 03 by applying the proposed resolutions of the following issues to CD-01 version: 116 - Update references and related changes 136 - SOAP-over-UDP - UNICAST_UDP_REPEAT and MULTICAST_UDP_REPEAT constant values Updated copyright year to 2009. Appendix C. Acknowledgements.Included a list of TC participants.