



# 1 Web Services Make Connection 2 (WS-MakeConnection)

## 3 Committee Draft 05, February 1, 2007

### 4 Document identifier:

5 wsmc-1.0-spec-cd-05

### 6 Location:

7 <http://docs.oasis-open.org/ws-rx/wsmc/200702/wsmc-1.0-spec-cd-05.pdf>

### 8 Editors:

9 Doug Davis, IBM <[dug@us.ibm.com](mailto:dug@us.ibm.com)>  
10 Anish Karmarkar, Oracle <[Anish.Karmarkar@oracle.com](mailto:Anish.Karmarkar@oracle.com)>  
11 Gilbert Pilz, BEA <[gpilz@bea.com](mailto:gpilz@bea.com)>  
12 Steve Winkler, SAP <[steve.winkler@sap.com](mailto:steve.winkler@sap.com)>  
13 Ümit Yalçınalp, SAP <[umit.yalcinalp@sap.com](mailto:umit.yalcinalp@sap.com)>

### 14 Contributors:

15 See the Acknowledgments (Appendix D).

### 16 Abstract:

17 This specification (WS-MakeConnection) describes a protocol that allows messages to be transferred  
18 between nodes implementing this protocol by using a transport-specific back-channel. The protocol is  
19 described in this specification in a transport-independent manner allowing it to be implemented using  
20 different network technologies. To support interoperable Web services, a SOAP binding is defined within  
21 this specification.

22 The protocol defined in this specification depends upon other Web services specifications for the  
23 identification of service endpoint addresses and policies. How these are identified and retrieved are  
24 detailed within those specifications and are out of scope for this document.

25 By using the XML [XML], SOAP [SOAP 1.1], [SOAP 1.2] and WSDL [WSDL 1.1] extensibility model,  
26 SOAP-based and WSDL-based specifications are designed to be composed with each other to define a  
27 rich Web services environment. As such, WS-MakeConnection by itself does not define all the features  
28 required for a complete messaging solution. WS-MakeConnection is a building block that is used in  
29 conjunction with other specifications and application-specific protocols to accommodate a wide variety of  
30 requirements and scenarios related to the operation of distributed Web services.

### 31 Status:

32 This document was last revised or approved by the WS-RX on the above date. The level of approval is  
33 also listed above. Check the current location noted above for possible later revisions of this document.  
34 This document is updated periodically on no particular schedule. Technical Committee members should  
35 send comments on this specification to the Technical Committee's email list. Others should send  
36 comments to the Technical Committee by using the "Send A Comment" button on the Technical  
37 Committee's web page at <http://www.oasis-open.org/committees/ws-rx>. For information on whether any  
38 patents have been disclosed that may be essential to implementing this specification, and any offers of  
39 patent licensing terms, please refer to the Intellectual Property Rights section of the Technical  
40 Committee web page (<http://www.oasis-open.org/committees/ws-rx/ipr.php>). The non-normative errata  
41 page for this specification is located at <http://www.oasis-open.org/committees/ws-rx>.

## Table of Contents

1	Introduction.....	3
1.1	Notational Conventions.....	3
1.2	Namespace.....	4
1.3	Conformance.....	4
2	MakeConnection Model.....	5
2.1	Glossary.....	5
2.2	Protocol Preconditions.....	6
2.3	Example Message Exchange.....	6
3	MakeConnection.....	8
3.1	MakeConnection Anonymous URI.....	8
3.2	MakeConnection Message.....	8
3.3	MessagePending.....	10
3.4	MakeConnection Policy Assertion.....	10
4	Faults.....	12
4.1	Unsupported Selection .....	13
4.2	Missing Selection .....	13
5	Security Considerations.....	15
6	References.....	16
6.1	Normative.....	16
6.2	Non-Normative.....	17
	Appendix A. Schema.....	19
	Appendix B. WSDL.....	21
	Appendix C. Message Examples.....	23
	Appendix C.1 Example use of MakeConnection.....	23
	Appendix D. Acknowledgments.....	27
	Appendix E. Revision History.....	28
	Appendix F. Notices.....	29

# 1 Introduction

The primary goal of this specification is to create a mechanism for the transfer of messages between two endpoints when the sending endpoint is unable to initiate a new connection to the receiving endpoint. It defines a mechanism to uniquely identify non-addressable endpoints, and a mechanism by which messages destined for those endpoints can be delivered. It also defines a SOAP binding that is required for interoperability. Additional bindings can be defined.

This mechanism is extensible allowing additional functionality, such as security, to be tightly integrated. This specification integrates with and complements the WS-ReliableMessaging[WS-RM], WS-Security [WS-Security], WS-Policy [WS-Policy], and other Web services specifications. Combined, these allow for a broad range of reliable, secure messaging options.

## 1.1 Notational Conventions

The keywords "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 [KEYWORDS].

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 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.
- An ellipsis (i.e. "...") indicates a point of extensibility that allows other child or attribute content specified in this document. Additional children elements and/or attributes MAY be added at the indicated extension points but they MUST NOT contradict the semantics of the parent and/or owner, respectively. If an extension is not recognized it SHOULD be ignored.
- XML namespace prefixes (See Section 1.2) are used to indicate the namespace of the element being defined.

Elements and Attributes defined by this specification are referred to in the text of this document using XPath 1.0 [XPATH 1.0] expressions. Extensibility points are referred to using an extended version of this syntax:

- An element extensibility point is referred to using {any} in place of the element name. This indicates that any element name can be used, from any namespace other than the `wsmc:` namespace.
- An attribute extensibility point is referred to using @{any} in place of the attribute name. This indicates that any attribute name can be used, from any namespace other than the `wsmc:` namespace.

## 1.2 Namespace

The XML namespace [XML-ns] URI that MUST be used by implementations of this specification is:

<http://docs.oasis-open.org/ws-rx/wsmc/200702>

Dereferencing the above URI will produce the Resource Directory Description Language [RDDL 2.0] document that describes this namespace.

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

Table 1

Prefix	Namespace
S	(Either SOAP 1.1 or 1.2)
S11	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
S12	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
wsmc	<a href="http://docs.oasis-open.org/ws-rx/wsmc/200702">http://docs.oasis-open.org/ws-rx/wsmc/200702</a>
wsrc	<a href="http://docs.oasis-open.org/ws-rx/wsrc/200702">http://docs.oasis-open.org/ws-rx/wsrc/200702</a>
wsa	<a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a>
wsam	<a href="http://www.w3.org/2007/02/addressing/metadata">http://www.w3.org/2007/02/addressing/metadata</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>

The normative schema for WS-MakeConnection can be found linked from the namespace document that is located at the namespace URI specified above.

All sections explicitly noted as examples are informational and are not to be considered normative.

## 1.3 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 SOAP Node MUST NOT use the XML namespace identifier for this specification (listed in Section 1.2) within SOAP Envelopes unless it is conformant with this specification.

Normative text within this specification takes precedence over normative outlines, which in turn take precedence over the XML Schema [XML Schema Part 1, Part 2] descriptions.

## 2 MakeConnection Model

The WS-Addressing [WS-Addressing] specification defines the anonymous URI to identify non-addressable endpoints and to indicate a protocol-specific back-channel is to be used for any messages destined for that endpoint. For example, when used in the WS-Addressing ReplyTo EPR, the use of this anonymous URI is meant to indicate that any response message is to be transmitted on the transport-specific back-channel. In the HTTP case this would mean that any response message is sent back on the HTTP response flow.

In cases where the connection is still available the WS-Addressing URI is sufficient. However, in cases where the original connection is no longer available, additional mechanisms are needed. Take the situation where the original connection that carried a request message is broken and therefore is no longer available to carry a response back to the original sender. Traditionally, non-anonymous (addressable) EPRs would be used in these cases to allow for the sender of the response message to initiate new connections as needed. However, if the sender of the request message is unable (or unwilling) to accept new connections then the only option available is for it to establish a new connection for the purposes of allowing the response message to be sent. This specification defines a mechanism by which a new connection can be established.

The MakeConnection model consists of a two key aspects:

- An optional anonymous-like URI template is defined that has similar semantics to WS-Addressing's anonymous, but also allows for each non-addressable endpoint to be uniquely identified
- A new message is defined that establishes a connection that can then be used to transmit messages to these non-addressable endpoints

Figure 1 below illustrates the overall flow involved in the use of MakeConnection:

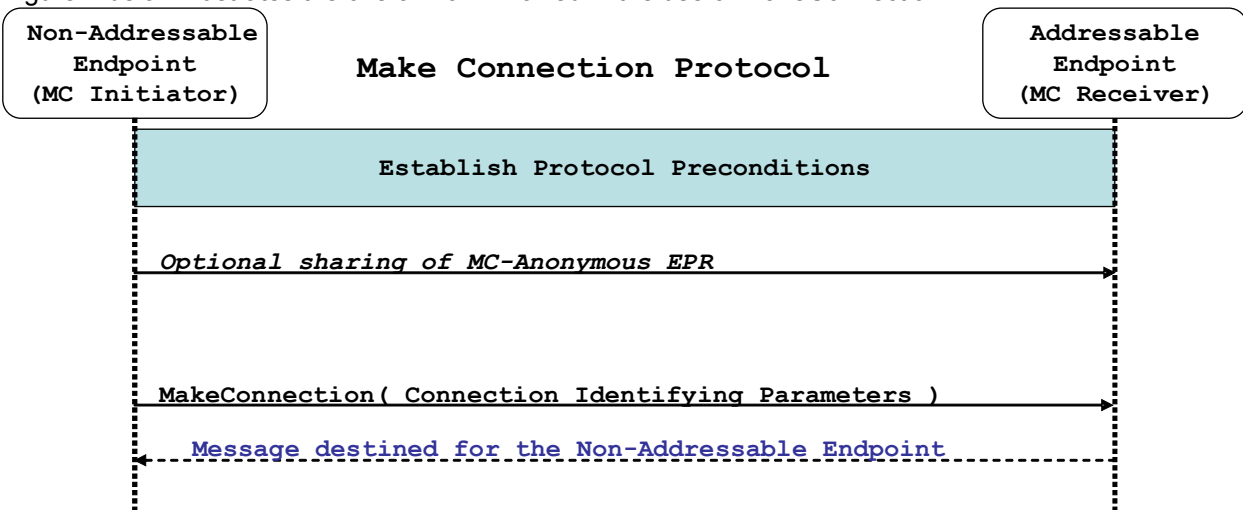


Figure 1 – Make Connection Model

The MakeConnection message is used to establish a new connection between the two endpoints. Within the message is identifying information that is used to uniquely identify a message that is eligible for transmission.

### 2.1 Glossary

The following definitions are used throughout this specification:

155 **Back-channel:** When the underlying transport provides a mechanism to return a transport-protocol  
156 specific response, capable of carrying a SOAP message, without initiating a new connection, this  
157 specification refers to this mechanism as a back-channel.

158 **Endpoint:** As defined in the WS-Addressing specification; a Web service Endpoint is a (referenceable)  
159 entity, processor, or resource to which Web service messages can be addressed. Endpoint references  
160 (EPRs) convey the information needed to address a Web service Endpoint.

161 **MC Initiator** The endpoint that transmits the MakeConnection message – the destination endpoint for the  
162 messages being sent on the transport-specific back-channel.

163 **MC Receiver:** The endpoint that receives the MakeConnection message – the source endpoint for the  
164 messages being sent on the transport-specific back-channel.

165 **Receive:** The act of reading a message from a network connection.

166 **Transmit:** The act of writing a message to a network connection.

167 **2.2 Protocol Preconditions**

168 The correct operation of the protocol requires that a number of preconditions **MUST** be established prior  
169 to the processing of the initial sequenced message:

- 170     • The MC Receiver **MUST** be capable of accepting new incoming connections.
- 171     • The MC Initiator **MUST** be capable of creating new outgoing connections to the MC Receiver, and  
172 those connections **MUST** have a back-channel.
- 173     • If a secure exchange of messages is **REQUIRED**, then the MC Initiator and MC Receiver **MUST**  
174 have a security context.

175 **2.3 Example Message Exchange**

176 Figure 2 illustrates a message exchange in which the response message is delivered using  
177 MakeConnection.



178 Figure 2: Example WS-MakeConnection Message Exchange

- 179 1. The protocol preconditions are established. These include policy exchange, endpoint resolution,  
180 and establishing trust.
- 181 2. The client (MC Initiator) sends a GetQuote request message to the service (MC Receiver). The  
182 WS-Addressing `wsa:ReplyTo` EPR uses the MakeConnection Anonymous URI Template –  
183 indicating that if the GetQuoteResponse message is not sent back on this connection's back-  
184 channel, then the client will use MakeConnection to retrieve it.
- 185 3. The service receives the request message and decides to close the connection by sending back an  
186 empty response (in the HTTP case an HTTP 202 Accept is sent).
- 187 4. The client sends a MakeConnection message to the service. Within the MakeConnection element is  
188 the `wsmc:Address` element containing the same MakeConnection Anonymous URI used in step 2.
- 189 5. The service has not completed executing the GetQuote operation and decides to close the  
190 connection by sending back an empty response (in the HTTP case an HTTP 202 Accept) indicating  
191 that no messages destined for this MC Initiator are available at this time.
- 192 6. The client sends a second MakeConnection message to the service. Within the MakeConnection  
193 element is the `wsmc:Address` element containing the same MakeConnection Anonymous URI  
194 used in step 2.
- 195 7. The service uses this new connection to transmit the GetQuoteResponse message.

196 The service can assume that because the MakeConnection Anonymous URI Template was used in the  
197 `wsa:ReplyTo` EPR the client will act as an MC Initiator for the purposes of retrieving messages destined  
198 to that EPR (i.e. responses to the GetQuote). This allows the service the option of immediately releasing  
199 resources used by the original connection – knowing that the client will, at some later point in time,  
200 establish a new connection on which the GetQuoteResponse can be transmitted. Likewise, when the first  
201 MakeConnection is received by the service, it again has the option of leaving the connection open until the  
202 GetQuoteResponse is ready to be transmitted, or it can close the connection immediately knowing that the  
203 MC Initiator will retransmit the MakeConnection message at some later point in time. Since the nature and  
204 dynamic characteristics of the underlying transport and potential intermediaries are unknown in the  
205 general case, the timing of re-transmissions cannot be specified. Additionally, over-aggressive re-  
206 transmissions have been demonstrated to cause transport or intermediary flooding which are  
207 counterproductive. Consequently, implementers are encouraged to utilize adaptive mechanisms that  
208 dynamically adjust re-transmission time and the back-off intervals that are appropriate to the nature of the  
209 transports and intermediaries envisioned. For the case of TCP/IP transports, a mechanism similar to that  
210 described as RTTM in RFC 1323 [RTTM] SHOULD be considered.

211 Now that the basic model has been outlined, the details of this protocol are now provided in Section 3.

## 3 MakeConnection

The following sub-sections define the various MakeConnection features, and prescribe their usage by a conformant implementations.

### 3.1 MakeConnection Anonymous URI

When an Endpoint is not directly addressable (e.g. behind a firewall or not able to allow incoming connections), an anonymous URI in the EPR address property can indicate such an Endpoint. The WS-Addressing anonymous URI is one such anonymous URI. This specification defines a URI template (the WS-MC anonymous URI) which may be used to uniquely identify anonymous Endpoints.

```
http://docs.oasis-open.org/ws-rx/wsmc/200702/anonymous?id={unique-String}
```

The appearance of an instance of this URI template in the `wsa:Address` value of an EPR indicates a protocol-specific back-channel will be established through a mechanism such as `MakeConnection`, defined below. When using this URI template, “{unique-String}” MUST be replaced by a globally unique string (e.g a UUID value as defined by RFC4122[UUID]). This specification does not require the use of one particular string generation scheme. This string uniquely distinguishes the Endpoint. A sending Endpoint SHOULD Transmit messages at Endpoints identified with the URI template using a protocol-specific back-channel, including but not limited to those established with a `MakeConnection` message. Note, this URI template is semantically similar to the WS-Addressing anonymous URI if a protocol-specific back-channel is available.

### 3.2 MakeConnection Message

The `MakeConnection` element is sent in the body of a one-way message that establishes a contextualized back-channel for the transmission of messages according to matching criteria (defined below). In the non-faulting case, if no matching message is available then no SOAP envelope will be returned on the back-channel. A common usage will be a client sending `MakeConnection` to a server for the purpose of receiving asynchronous response messages.

The following exemplar defines the `MakeConnection` syntax:

```
<wsmc:MakeConnection ...>
  <wsmc:Address ...> xs:anyURI </wsmc:Address> ?
  <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier> ?
  ...
</wsmc:MakeConnection>
```

The following describes the content model of the `MakeConnection` element.

/wsmc:MakeConnection

This element allows the sender to create a transport-specific back-channel that can be used to return a message that matches the selection criteria. Endpoints MUST NOT send this element as a header block. At least one selection criteria sub-element MUST be specified – if not a `MissingSelection` fault MUST be generated.

/wsmc:MakeConnection/wsmc:Address

This element specifies the URI (`wsa:Address`) of the initiating Endpoint. Endpoints MUST NOT return messages on the transport-specific back-channel unless they have been addressed to this URI. This `Address` property and a message's WS-Addressing destination property are considered identical when they are exactly the same character-for-character. Note that URIs which are not identical in this sense



253 may in fact be functionally equivalent. Examples include URI references which differ only in case, or  
254 which are in external entities which have different effective base URIs.

255 `/wsmc:MakeConnection/wsmc:Address/@{any}`

256 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
257 element.

258 `/wsmc:MakeConnection/wsrn:Identifier`

259 This element specifies the WS-RM Sequence Identifier that establishes the context for the transport-  
260 specific back-channel. The Sequence Identifier should be compared with the Sequence Identifiers  
261 associated with the messages held by the sending Endpoint, and if there is a matching message it will be  
262 returned.

263 `/wsmc:MakeConnection/wsrn:Identifier/@{any}`

264 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
265 element.

266 `/wsmc:MakeConnection/{any}`

267 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
268 to be passed. This allows fine-tuning of the messages to be returned, additional selection criteria included  
269 here are logically ANDed with the `Address` and/or `wsrn:Identifier`. If an extension is not supported  
270 by the Endpoint then it should generate an `UnsupportedSelection` fault.

271 `/wsmc:MakeConnection/@{any}`

272 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
273 element.

274 If more than one selection criteria element is present, then the MC Receiver processing the  
275 `MakeConnection` message MUST insure that any SOAP Envelope flowing on the back-channel satisfies  
276 all of those selection criteria.

277 The management of messages that are awaiting the establishment of a back-channel to their receiving  
278 Endpoint is an implementation detail that is outside the scope of this specification. Note, however, that  
279 these messages form a class of asynchronous messages that is not dissimilar from "ordinary"  
280 asynchronous messages that are waiting for the establishment of a connection to their destination  
281 Endpoints.

282 This specification places no constraint on the types of messages that can be returned on the transport-  
283 specific back-channel. As in an asynchronous environment, it is up to the recipient of the  
284 `MakeConnection` message to decide which messages are appropriate for transmission to any particular  
285 Endpoint. However, the Endpoint processing the `MakeConnection` message MUST insure that the  
286 messages match the selection criteria as specified by the child elements of the `MakeConnection`  
287 element.

288 Since the message exchange pattern use by `MakeConnection` is untraditional, the following points need  
289 to be reiterated for clarification:

- 290 ● The `MakeConnection` message is logically part of a one-way operation; there is no reply  
291 message to the `MakeConnection` itself, and any response flowing on the transport back-channel  
292 is a pending message.
- 293 ● Since there is no reply message to `MakeConnection`, the WS-Addressing specific rules in  
294 section 3.4 "Formulating a Reply Message" are not used. Therefore, the value of any

295        `wsa:ReplyTo` element in the `MakeConnection` message has no effective impact since the WS-  
296        Addressing [`reply endpoint`] property that is set by the presence of `wsa:ReplyTo` is not  
297        used.

- 298        ● In the absence of any pending message, there will be no message transmitted on the transport  
299        back-channel. E.g. in the HTTP case just an `HTTP 202 Accepted` will be returned without any  
300        SOAP envelope in the HTTP response message.
- 301        ● When there is a message pending, it is sent on the transport back-channel, using the connection  
302        that has been initiated by the `MakeConnection` request.

### 303 3.3 MessagePending

304        When `MakeConnection` is used, and a message is returned on the transport-specific back-channel, the  
305        `MessagePending` header SHOULD be included on the returned message as an indicator whether there  
306        are additional messages waiting to be retrieved using the same selection criteria that was specified in the  
307        `MakeConnection` element.

308        The following exemplar defines the `MessagePending` syntax:

```
309        <wsmc:MessagePending pending="xs:boolean" ...>  
310        ...  
311        </wsmc:MessagePending>
```

312        The following describes the content model of the `MessagePending` header block.

313        `/wsmc:MessagePending`

314        This element indicates whether additional messages are waiting to be retrieved.

315        `/wsmc:MessagePending@pending`

316        This attribute, when set to "true", indicates that there is at least one message waiting to be retrieved.

317        When this attribute is set to "false" it indicates there are currently no messages waiting to be retrieved.

318        `/wsmc:MessagePending/{any}`

319        This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
320        to be passed.

321        `/wsmc:MessagePending/@{any}`

322        This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
323        element.

324        The absence of the `MessagePending` header has no implication as to whether there are additional  
325        messages waiting to be retrieved.

### 326 3.4 MakeConnection Policy Assertion

327        The `MakeConnection` policy assertion indicates that the `MakeConnection` protocol (operation and the use  
328        of the `MakeConnection` URI template in `EndpointReferences`) is supported. This assertion has `Endpoint`  
329        Policy Subject [`WS-PolicyAttachment`].

330        The normative outline for the `MakeConnection` assertion is:

```
331        <wsmc:MCSupported ...> ... </wsmc:MCSupported>
```

332        The following describes the content model of the `MCSupported` element.

333 /wsmc:MCSupported

334 A policy assertion that specifies that the MakeConnection protocol is supported.

335 /wsmc:MCSupported/{any}

336 This is an extensibility mechanism to allow different (extensible) types of information, based on a schema,  
337 to be passed.

338 /wsmc:MCSupported/@{any}

339 This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the  
340 element.

341 Because this policy assertion expresses a capability of a receiver (rather than a requirement on the  
342 sender), care should be taken to ensure that it is decorated with the appropriate WS-Policy artifacts to  
343 indicate that use, support and understanding, of this assertion is optional to the sender.

## 4 Faults

Entities that generate WS-MakeConnection faults MUST include as the [action] property the default fault action IRI defined below. The value from the W3C Recommendation is below for informational purposes:

```
http://docs.oasis-open.org/ws-rx/wsmc/200702/fault
```

The faults defined in this section are generated if the condition stated in the preamble is met. Fault handling rules are defined in section 6 of WS-Addressing SOAP Binding.

The definitions of faults use the following properties:

[Code] The fault code.

[Subcode] The fault subcode.

[Reason] The English language reason element.

[Detail] The detail element(s). If absent, no detail element is defined for the fault. If more than one detail element is defined for a fault, implementations MUST include the elements in the order that they are specified.

Entities that generate WS-MakeConnection faults MUST set the [Code] property to either "Sender" or "Receiver". These properties are serialized into text XML as follows:

SOAP Version	Sender	Receiver
SOAP 1.1	S11:Client	S11:Server
SOAP 1.2	S:Sender	S:Receiver

The properties above bind to a SOAP 1.2 fault as follows:

```
<S:Envelope>
  <S:Header>
    <wsa:Action>
      http://docs.oasis-open.org/ws-rx/wsmc/200702/fault
    </wsa:Action>
    <!-- Headers elided for brevity. -->
  </S:Header>
  <S:Body>
    <S:Fault>
      <S:Code>
        <S:Value> [Code] </S:Value>
        <S:Subcode>
          <S:Value> [Subcode] </S:Value>
        </S:Subcode>
      </S:Code>
      <S:Reason>
        <S:Text xml:lang="en"> [Reason] </S:Text>
      </S:Reason>
      <S:Detail>
        [Detail]
        ...
      </S:Detail>
    </S:Fault>
  </S:Body>
</S:Envelope>
```

The properties bind to a SOAP 1.1 fault as follows when the fault is generated as a result of processing a MakeConnection message:

```

387 <S11:Envelope>
388   <S11:Body>
389     <S11:Fault>
390       <faultcode> [Subcode] </faultcode>
391       <faultstring> [Reason] </faultstring>
392     </S11:Fault>
393   </S11:Body>
394 </S11:Envelope>

```

## 395 4.1 Unsupported Selection

396 The QName of the unsupported element(s) are included in the detail.

397 Properties:

398 [Code] Receiver

399 [Subcode] wsmc:UnsupportedSelection

400 [Reason] The extension element used in the message selection is not supported by the MakeConnection  
401 receiver

402 [Detail]

```

403 <wsmc:UnsupportedElement> xs:QName </wsmc:UnsupportedElement>+

```

Generated by	Condition	Action Upon Generation	Action Upon Receipt
MakeConnection receiver	In response to a <code>MakeConnection</code> message containing a selection criteria in the extensibility section of the message that is not supported	Unspecified.	Unspecified.

## 404 4.2 Missing Selection

405 The MakeConnection element did not contain any selection criteria.

406 Properties:

407 [Code] Receiver

408 [Subcode] wsmc:MissingSelection

409 [Reason] The MakeConnection element did not contain any selection criteria.

410 [Detail]

Generated by	Condition	Action Upon Generation	Action Upon Receipt
MakeConnection receiver	In response to a MakeConnection message that does not contain any selection criteria	Unspecified.	Unspecified.

## 5 Security Considerations

It is strongly RECOMMENDED that the communication between Web services be secured using the mechanisms described in WS-Security. In order to properly secure messages, the body and all relevant headers need to be included in the signature. Specifically, any standard messaging headers, such as those from WS-Addressing, need to be signed with the body in order to "bind" the two together.

Different security mechanisms may be desired depending on the frequency of messages. For example, for infrequent messages, public key technologies may be adequate for integrity and confidentiality. However, for high-frequency events, it may be more performant to establish a security context for the events using the mechanisms described in WS-Trust [[Trust](#)] and WS-SecureConversation [[SecureConversation](#)]. It should be noted that if a shared secret is used it is RECOMMENDED that derived keys be used to strengthen the secret as described in WS-SecureConversation.

Requests for messages which are not available to anonymous parties are strongly RECOMMENDED to require usage of WS-Security so that the requestor can be authenticated and authorized to access the indicated messages. Similarly, integrity and confidentiality SHOULD be used whenever messages have restricted access.

Recipients of messages are RECOMMENDED to validate the signature to authenticate and verify the integrity of the data. Specifically, recipients SHOULD verify that the sender has the right to "speak" for the message.

The following list summarizes common classes of attacks that apply to this protocol and identifies the mechanism to prevent/mitigate the attacks:

- Message alteration - Alteration is prevented by including signatures of the message information using WS-Security.
- Message disclosure - Confidentiality is preserved by encrypting sensitive data using WS-Security.
- Key integrity - Key integrity is maintained by using the strongest algorithms possible (by comparing secured policies - see WS-Policy and WS-SecurityPolicy [[SecurityPolicy](#)]).
- Authentication - Authentication is established using the mechanisms described in WS-Security and WS-Trust. Each message is authenticated using the mechanisms described in WS-Security.
- Accountability - Accountability is a function of the type of and strength of the key and algorithms being used. In many cases, a strong symmetric key provides sufficient accountability. However, in some environments, strong PKI signatures are required.
- Availability - All reliable messaging services are subject to a variety of availability attacks. Replay detection is a common attack and it is RECOMMENDED that this be addressed by the mechanisms described in WS-Security. Other attacks, such as network-level denial of service attacks are harder to avoid and are outside the scope of this specification. That said, care should be taken to ensure that minimal state is saved prior to any authenticating sequences.
- Replay - Messages may be replayed for a variety of reasons. To detect and eliminate this attack, mechanisms should be used to identify replayed messages such as the timestamp/nonce outlined in WS-Security. Alternatively, and optionally, other technologies, such as sequencing, can also be used to prevent replay of application messages.

Service endpoints SHOULD scope its searching of messages to those that were processed under the same security context as the requesting MakeConnection message.

## 6 References

### 6.1 Normative

#### [KEYWORDS]

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

<http://www.ietf.org/rfc/rfc2119.txt>

#### [WS-RM]

OASIS WS-RX Technical Committee Draft, "Web Services Reliable Messaging (WS-ReliableMessaging)," August 2005.

<http://docs.oasis-open.org/ws-rx/wsrmp/200702/wsrmp-1.1-spec-cd-05.pdf>

#### [WS-RM Policy]

OASIS WS-RX Technical Committee Draft, "Web Services ReliableMessaging Policy Assertion( WS-RM Policy)" February 2007

<http://docs.oasis-open.org/ws-rx/wsrmp/200702/wsrmp-1.1-spec-cd-05.pdf>

#### [SOAP 1.1]

W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

#### [SOAP 1.2]

W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging Framework" June 2003.

<http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>

#### [URI]

T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 3986, MIT/LCS, U.C. Irvine, Xerox Corporation, January 2005.

<http://ietf.org/rfc/rfc3986>

#### [UUID]

P. Leach, M. Mealling, R. Salz, "A Universally Unique Identifier (UUID) URN Namespace," RFC 4122, Microsoft, Refactored Networks - LLC, DataPower Technology Inc, July 2005

<http://www.ietf.org/rfc/rfc4122.txt>

#### [XML]

W3C Recommendation, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", September 2006.

<http://www.w3.org/TR/REC-xml/>

#### [XML-ns]

W3C Recommendation, "Namespaces in XML," 14 January 1999.



485 <http://www.w3.org/TR/1999/REC-xml-names-19990114/>  
486 **[XML-Schema Part1]**  
487 W3C Recommendation, "[XML Schema Part 1: Structures](#)," October 2004.  
488 <http://www.w3.org/TR/xmlschema-1/>  
489 **[XML-Schema Part2]**  
490 W3C Recommendation, "[XML Schema Part 2: Datatypes](#)," October 2004.  
491 <http://www.w3.org/TR/xmlschema-2/>  
492 **[XPath 1.0]**  
493 W3C Recommendation, "[XML Path Language \(XPath\) Version 1.0](#)," 16 November 1999.  
494 <http://www.w3.org/TR/xpath>  
495 **[WSDL 1.1]**  
496 W3C Note, "[Web Services Description Language \(WSDL 1.1\)](#)," 15 March 2001.  
497 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>  
498 **[WS-Addressing]**  
499 W3C Recommendation, "[Web Services Addressing 1.0 - Core](#)", May 2006.  
500 <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>  
501 W3C Recommendation, "[Web Services Addressing 1.0 – SOAP Binding](#)", May 2006.  
502 <http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/>

## 503 **6.2 Non-Normative**

504 **[BSP 1.0]**  
505 WS-I Working Group Draft. "[Basic Security Profile Version 1.0](#)," August 2006  
506 <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>  
507 **[RDDL 2.0]**  
508 Jonathan Borden, Tim Bray, eds. "[Resource Directory Description Language \(RDDL\) 2.0](#)," January 2004  
509 <http://www.openhealth.org/RDDL/20040118/rddl-20040118.html>  
510 **[RFC 2617]**  
511 J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Loutonen, L. Stewart, "[HTTP](#)  
512 [Authentication: Basic and Digest Access Authentication](#)," June 1999.  
513 <http://www.ietf.org/rfc/rfc2617.txt>  
514 **[RFC 4346]**  
515 T. Dierks, E. Rescorla, "[The Transport Layer Security \(TLS\) Protocol Version 1.1](#)," April 2006.  
516 <http://www.ietf.org/rfc/rfc4346.txt>  
517 **[WS-Policy]**  
518 W3C Member Submission, "[Web Services Policy Framework \(WS-Policy\)](#)," April 2006.

519 <http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/>  
520 **[WS-PolicyAttachment]**  
521 W3C Member Submission, "[Web Services Policy Attachment \(WS-PolicyAttachment\)](#)," April 2006.  
522 [http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-](http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/)  
523 [20060425/](http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/)  
524 **[WS-Security]**  
525 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "[OASIS Web Services Security: SOAP Message Security 1.0 \(WS-Security 2004\)](#)", OASIS Standard 200401, March 2004.  
526 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>  
527 Anthony Nadalin, Chris Kaler, Phillip Hallam-Baker, Ronald Monzillo, eds. "[OASIS Web Services Security: SOAP Message Security 1.1 \(WS-Security 2004\)](#)", OASIS Standard 200602, February 2006.  
528 <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>  
529 **[RTTM]**  
530 V. Jacobson, R. Braden, D. Borman, "[TCP Extensions for High Performance](#)", RFC 1323, May  
531 1992.  
532 <http://www.rfc-editor.org/rfc/rfc1323.txt>  
533 **[SecurityPolicy]**  
534 G. Della-Libra, et. al. "[Web Services Security Policy Language \(WS-SecurityPolicy\)](#)", July 2005  
535 <http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf>  
536 **[SecureConversation]**  
537 S. Anderson, et al, "[Web Services Secure Conversation Language \(WS-SecureConversation\)](#)," February  
538 2005.  
539 <http://schemas.xmlsoap.org/ws/2004/04/sc/>  
540 **[Trust]**  
541 S. Anderson, et al, "[Web Services Trust Language \(WS-Trust\)](#)," February 2005.  
542 <http://schemas.xmlsoap.org/ws/2005/02/trust>

## 545 Appendix A. Schema

546 The normative schema that is defined for WS-MakeConnection using [XML-Schema Part1] and [XML-  
547 Schema Part2] is located at:

548 <http://docs.oasis-open.org/ws-rx/wsmc/200702/wsmc-1.0-schema-200702.xsd>

549 The following copy is provided for reference.

```
550 <?xml version="1.0" encoding="UTF-8"?>
551 <!--
552 OASIS takes no position regarding the validity or scope of any intellectual
553 property or other rights that might be claimed to pertain to the
554 implementation or use of the technology described in this document or the
555 extent to which any license under such rights might or might not be available;
556 neither does it represent that it has made any effort to identify any such
557 rights. Information on OASIS's procedures with respect to rights in OASIS
558 specifications can be found at the OASIS website. Copies of claims of rights
559 made available for publication and any assurances of licenses to be made
560 available, or the result of an attempt made to obtain a general license or
561 permission for the use of such proprietary rights by implementors or users of
562 this specification, can be obtained from the OASIS Executive Director.
563 OASIS invites any interested party to bring to its attention any copyrights,
564 patents or patent applications, or other proprietary rights which may cover
565 technology that may be required to implement this specification. Please
566 address the information to the OASIS Executive Director.
567 Copyright (c) OASIS Open 2002-2007. All Rights Reserved.
568 This document and translations of it may be copied and furnished to others,
569 and derivative works that comment on or otherwise explain it or assist in its
570 implementation may be prepared, copied, published and distributed, in whole or
571 in part, without restriction of any kind, provided that the above copyright
572 notice and this paragraph are included on all such copies and derivative
573 works. However, this document itself does not be modified in any way, such as
574 by removing the copyright notice or references to OASIS, except as needed for
575 the purpose of developing OASIS specifications, in which case the procedures
576 for copyrights defined in the OASIS Intellectual Property Rights document must
577 be followed, or as required to translate it into languages other than English.
578 The limited permissions granted above are perpetual and will not be revoked by
579 OASIS or its successors or assigns.
580 This document and the information contained herein is provided on an "AS IS"
581 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
582 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
583 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
584 FOR A PARTICULAR PURPOSE.
585 -->
586 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
587 xmlns:wsa="http://www.w3.org/2005/08/addressing"
588 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
589 targetNamespace="http://docs.oasis-open.org/ws-rx/wsmc/200702"
590 elementFormDefault="qualified" attributeFormDefault="unqualified">
591   <xs:import namespace="http://www.w3.org/2005/08/addressing"
592   schemaLocation="http://www.w3.org/2006/03/addressing/ws-addr.xsd"/>
593   <!-- Protocol Elements -->
594   <xs:complexType name="MessagePendingType">
595     <xs:sequence>
596       <xs:any namespace="##other" processContents="lax" minOccurs="0"
597 maxOccurs="unbounded"/>
598     </xs:sequence>
599     <xs:attribute name="pending" type="xs:boolean"/>
600     <xs:anyAttribute namespace="##other" processContents="lax"/>
```

```

601 </xs:complexType>
602 <xs:element name="MessagePending" type="wsmc:MessagePendingType"/>
603 <xs:element name="Address">
604   <xs:complexType>
605     <xs:simpleContent>
606       <xs:extension base="xs:anyURI">
607         <xs:anyAttribute namespace="##other" processContents="lax"/>
608       </xs:extension>
609     </xs:simpleContent>
610   </xs:complexType>
611 </xs:element>
612 <xs:complexType name="MakeConnectionType">
613   <xs:sequence>
614     <xs:element ref="wsmc:Address" minOccurs="0" maxOccurs="1"/>
615     <xs:any namespace="##other" processContents="lax" minOccurs="0"
616 maxOccurs="unbounded"/>
617   </xs:sequence>
618   <xs:anyAttribute namespace="##other" processContents="lax"/>
619 </xs:complexType>
620 <xs:element name="MakeConnection" type="wsmc:MakeConnectionType"/>
621 <xs:element name="UnsupportedElement">
622   <xs:simpleType>
623     <xs:restriction base="xs:QName"/>
624   </xs:simpleType>
625 </xs:element>
626 </xs:schema>

```

## 627 Appendix B. WSDL

628 This WSDL describes the WS-MC protocol from the point of view of the endpoint that receives the  
629 MakeConnection message.

630 Also note that this WSDL is intended to describe the internal structure of the WS-MC protocol, and will not  
631 generally appear in a description of a WS-MC-capable Web service. See section 3.4 Policy for a higher-  
632 level mechanism to indicate that WS-MC is supported.

633 The normative WSDL 1.1 definition for WS-MakeConnection is located at:

634 <http://docs.oasis-open.org/ws-rx/wsmc/200702/wsd/wsmc-1.0-wsdl-200702.wsdl>

635 The following non-normative copy is provided for reference.

```
636 <?xml version="1.0" encoding="utf-8"?>
637 <!--
638 OASIS takes no position regarding the validity or scope of any intellectual
639 property or other rights that might be claimed to pertain to the
640 implementation or use of the technology described in this document or the
641 extent to which any license under such rights might or might not be available;
642 neither does it represent that it has made any effort to identify any such
643 rights. Information on OASIS's procedures with respect to rights in OASIS
644 specifications can be found at the OASIS website. Copies of claims of rights
645 made available for publication and any assurances of licenses to be made
646 available, or the result of an attempt made to obtain a general license or
647 permission for the use of such proprietary rights by implementors or users of
648 this specification, can be obtained from the OASIS Executive Director.
649 OASIS invites any interested party to bring to its attention any copyrights,
650 patents or patent applications, or other proprietary rights which may cover
651 technology that may be required to implement this specification. Please
652 address the information to the OASIS Executive Director.
653 Copyright (c) OASIS Open 2002-2007. All Rights Reserved.
654 This document and translations of it may be copied and furnished to others,
655 and derivative works that comment on or otherwise explain it or assist in its
656 implementation may be prepared, copied, published and distributed, in whole or
657 in part, without restriction of any kind, provided that the above copyright
658 notice and this paragraph are included on all such copies and derivative
659 works. However, this document itself does not be modified in any way, such as
660 by removing the copyright notice or references to OASIS, except as needed for
661 the purpose of developing OASIS specifications, in which case the procedures
662 for copyrights defined in the OASIS Intellectual Property Rights document must
663 be followed, or as required to translate it into languages other than English.
664 The limited permissions granted above are perpetual and will not be revoked by
665 OASIS or its successors or assigns.
666 This document and the information contained herein is provided on an "AS IS"
667 basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
668 NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
669 INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS
670 FOR A PARTICULAR PURPOSE.
671 -->
672 <wsc:definitions xmlns:wsc="http://schemas.xmlsoap.org/wsc/"
673 xmlns:xs="http://www.w3.org/2001/XMLSchema"
674 xmlns:wsa="http://www.w3.org/2005/08/addressing"
675 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
676 xmlns:tns="http://docs.oasis-open.org/ws-rx/wsmc/200702/wsd"
677 targetNamespace="http://docs.oasis-open.org/ws-rx/wsmc/200702/wsd">
678
679   <wsc:types>
```

```

680     <xs:schema>
681         <xs:import namespace="http://docs.oasis-open.org/ws-rx/wsmc/200702"
682 schemaLocation="http://docs.oasis-open.org/ws-rx/wsmc/200702/wsmc-1.1-schema-
683 200702.xsd"/>
684     </xs:schema>
685 </wsdl:types>

686 <wsdl:message name="MakeConnection">
687     <wsdl:part name="makeConnection" element="wsmc:MakeConnection"/>
688 </wsdl:message>

689 <wsdl:portType name="MCAbstractPortType">
690     <wsdl:operation name="MakeConnection">
691         <wsdl:input message="tns:MakeConnection" wsam:Action="http://docs.oasis-
692 open.org/ws-rx/wsmc/200702/MakeConnection"/>
693         <!-- As described in the WS-MakeConnection specification, the
694              MakeConnection operation establishes a connection. If a matching
695              message is available then the back-channel of the connection will
696              be used to carry the message. In SOAP terms the returned message
697              is not a response, so there is no WSDL output message. -->
698     </wsdl:operation>
699 </wsdl:portType>

700 </wsdl:definitions>

```

## 701 Appendix C. Message Examples

### 702 Appendix C.1 Example use of MakeConnection

703 To illustrate how a `MakeConnection` message exchange can be used to deliver messages to an  
704 Endpoint that is not addressable, consider the case of a pub/sub scenario in which the Endpoint to which  
705 notifications are to be delivered (the "event consumer") is not addressable by the notification sending  
706 Endpoint (the "event producer"). In this scenario the event consumer must initiate the connections in order  
707 for the notifications to be delivered. One possible set of message exchanges (using HTTP) that  
708 demonstrate how this can be achieved using `MakeConnection` is shown below.

709 **Step 1** – During a "subscribe" operation, the event consumer's EPR specifies the MC anonymous URI  
710 and the WS-RM Policy Assertion to indicate whether or not RM is required:

```
711 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
712 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
713 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsrm/200702"
714 xmlns:wsa="http://www.w3.org/2005/08/addressing">
715   <S:Header>
716     <wsa:To> http://example.org/subscriptionService </wsa:To>
717     <wsa:MessageID> http://client456.org/id-a6d8-a7c2eb546813</wsa:MessageID>
718     <wsa:ReplyTo>
719       <wsa:To> http://client456.org/response </wsa:To>
720     </wsa:ReplyTo>
721   </S:Header>
722   <S:Body>
723     <sub:Subscribe xmlns:sub="http://example.org/subscriptionService">
724       <!-- subscription service specific data -->
725       <targetEPR>
726         <wsa:Address>http://docs.oasis-open.org/ws-
727 rx/wsrm/200702/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:Address>
728         <wsa:Metadata>
729           <wsp:Policy wsu:Id="MyPolicy">
730             <wsrm:RMAssertion/>
731           </wsp:Policy>
732         </wsa:Metadata>
733       </targetEPR>
734     </sub:Subscribe>
735   </S:Body>
736 </S:Envelope>
```

737 In this example the `subscribe` and `targetEPR` elements are simply examples of what a subscription  
738 request message might contain. Note: the `wsa:Address` element contains the MC anonymous URI  
739 indicating that the notification producer needs to queue the messages until they are requested using the  
740 `MakeConnection` message exchange. The EPR also contains the WS-RM Policy Assertion indicating  
741 the RM must be used when notifications related to this subscription are sent.

742 **Step 2** – Once the subscription is established, the event consumer checks for a pending message:

```
743 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
744 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
745 xmlns:wsa="http://www.w3.org/2005/08/addressing">
746   <S:Header>
747     <wsa:Action>http://docs.oasis-open.org/ws-
748 rx/wsmc/200702/MakeConnection</wsa:Action>
```

```

749     <wsa:To> http://example.org/subscriptionService </wsa:To>
750   </S:Header>
751   <S:Body>
752     <wsmc:MakeConnection>
753       <wsmc:Address>http://docs.oasis-open.org/ws-
754 rx/wsmc/200702/anonymous?id=550e8400-e29b-11d4-a716-
755 446655440000</wsmc:Address>
756     </wsmc:MakeConnection>
757   </S:Body>
758 </S:Envelope>

```

759 **Step 3** – If there are messages waiting to be delivered then a message will be returned back to the event  
760 consumer. However, because WS-RM is being used to deliver the messages, the first message returned  
761 is a `CreateSequence`:

```

762 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
763 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
764 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702"
765 xmlns:wsa="http://www.w3.org/2005/08/addressing">
766   <S:Header>
767     <wsa:Action>http://docs.oasis-open-org/ws-
768 rx/wsmr/200702/CreateSequence</wsa:Action>
769     <wsa:To>http://docs.oasis-open.org/ws-
770 rx/wsmr/200702/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
771     <wsa:ReplyTo> http://example.org/subscriptionService </wsa:ReplyTo>
772     <wsa:MessageID> http://example.org/id-123-456 </wsa:MessageID>
773     <wsmc:MessagePending pending="true"/>
774   </S:Header>
775   <S:Body>
776     <wsmr:CreateSequence>
777       <wsmr:AcksTo>
778         <wsa:Address> http://example.org/subscriptionService </wsa:Address>
779       </wsmr:AcksTo>
780     </wsmr:CreateSequence>
781   </S:Body>
782 </S:Envelope>

```

783 Notice from the perspective of how the RM Source on the event producer interacts with the RM  
784 Destination of those messages, nothing new is introduced by the use of the `MakeConnection`, the use  
785 of RM protocol is the same as the case where the event consumer is addressable. Note the message  
786 contains a `wsmc:MessagePending` header indicating that additional message are waiting to be  
787 delivered.

788 **Step 4** – The event consumer will respond with a `CreateSequenceResponse` message per normal WS-  
789 Addressing rules:

```

790 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
791 xmlns:wsmr="http://docs.oasis-open.org/ws-rx/wsmr/200702"
792 xmlns:wsa="http://www.w3.org/2005/08/addressing">
793   <S:Header>
794     <wsa:Action>http://docs.oasis-open-org/ws-
795 rx/wsmr/200702/CreateSequenceResponse</wsa:Action>
796     <wsa:To> http://example.org/subscriptionService </wsa:To>
797     <wsa:RelatesTo> http://example.org/id-123-456 </wsa:RelatesTo>
798   </S:Header>
799   <S:Body>
800     <wsmr:CreateSequenceResponse>
801       <wsmr:Identifier> http://example.org/rmid-456 </wsmr:Identifier>

```



```

802     </wsrm:CreateSequenceResponse>
803   </S:Body>
804 </S:Envelope>

```

805 Note, this message is carried on an HTTP request directed to the `wsa:ReplyTo` EPR, and the HTTP  
806 response will be an HTTP 202.

807 **Step 5** – The event consumer checks for another message pending:

```

808 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
809 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
810 xmlns:wsa="http://www.w3.org/2005/08/addressing">
811   <S:Header>
812     <wsa:Action>http://docs.oasis-open.org/ws-
813 rx/wsmc/200702/MakeConnection</wsa:Action>
814     <wsa:To> http://example.org/subscriptionService </wsa:To>
815   </S:Header>
816   <S:Body>
817     <wsmc:MakeConnection>
818       <wsmc:Address>http://docs.oasis-open.org/ws-
819 rx/wsmc/200702/anonymous?id=550e8400-e29b-11d4-a716-
820 446655440000</wsmc:Address>
821     </wsmc:MakeConnection>
822   </S:Body>
823 </S:Envelope>

```

824 Notice this is the same message as the one sent in step 2.

825 **Step 6** – Since there is a message pending for this destination then it is returned on the HTTP response:

```

826 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
827 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
828 xmlns:wsrm="http://docs.oasis-open.org/ws-rx/wsmc/200702"
829 xmlns:wsa="http://www.w3.org/2005/08/addressing">
830   <S:Header>
831     <wsa:Action> http://example.org/eventType1 </wsa:Action>
832     <wsa:To>http://docs.oasis-open.org/ws-
833 rx/wsrm/200702/anonymous?id=550e8400-e29b-11d4-a716-446655440000</wsa:To>
834     <wsrm:Sequence>
835       <wsrm:Identifier> http://example.org/rmid-456 </wsrm:Identifier>
836     </wsrm:Sequence>
837     <wsmc:MessagePending pending="true"/>
838   </S:Header>
839   <S:Body>
840     <!-- event specific data -->
841   </S:Body>
842 </S:Envelope>

```

843 As noted in step 3, the use of the RM protocol does not change when using `MakeConnection`. The  
844 format of the messages, the order of the messages sent and the timing of when to send it remains the  
845 same.

846 **Step 7** – At some later interval, or immediately due to the `MessagePending` header's "pending"  
847 attribute being set to "true", the event consumer will poll again:

```

848 <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
849 xmlns:wsmc="http://docs.oasis-open.org/ws-rx/wsmc/200702"
850 xmlns:wsa="http://www.w3.org/2005/08/addressing">

```

```

851 <S:Header>
852 <wsa:Action>http://docs.oasis-open.org/ws-
853 rx/wsmc/200702/MakeConnection</wsa:Action>
854 <wsa:To> http://example.org/subscriptionService </wsa:To>
855 </S:Header>
856 <S:Body>
857 <wsmc:MakeConnection>
858 <wsmc:Address>http://docs.oasis-open.org/ws-
859 rx/wsmc/200702/anonymous?id=550e8400-e29b-11d4-a716-
860 446655440000</wsmc:Address>
861 </wsmc:MakeConnection>
862 </S:Body>
863 </S:Envelope>

```

864 Notice this is the same message as the one sent in steps 2 and 5. As in steps 3 and 6, the response to  
865 the `MakeConnection` can be any message destined to the specified Endpoint. This allows the event  
866 producer to send not only application messages (events) but RM protocol messages (e.g.  
867 `CloseSequence`, `TerminateSequence` or even additional `CreateSequences`) as needed.

868 **Step 8** – If at any point in time there are no messages pending, in response to a `MakeConnection` the  
869 event producer returns an HTTP 202 back to the event consumer. The process then repeats (back to step  
870 7) until the subscription ends.

## 871 **Appendix D. Acknowledgments**

872 The following individuals have provided invaluable input into the initial contribution:

873 Keith Ballinger(Microsoft), Stefan Batres(Microsoft), Rebecca Bergersen(Iona), Allen  
874 Brown(Microsoft), Kyle Brown(IBM), Michael Conner(IBM), George Copeland(Microsoft),  
875 Francisco Curbera(IBM), Paul Fremantle(IBM), Steve Graham(IBM), Pat Helland(Microsoft), Rick  
876 Hill(Microsoft), Scott Hinkelman(IBM), Tim Holloway(IBM), Efim Hudis(Microsoft), David  
877 Ingham(Microsoft), Gopal Kakivaya(Microsoft), Johannes Klein(Microsoft), Frank Leymann(IBM),  
878 Martin Nally(IBM), Peter Niblett(IBM), Jeffrey Schlimmer(Microsoft), James Snell(IBM), Keith  
879 Stobie(Microsoft), Satish Thatte(Microsoft), Stephen Todd(IBM), Sanjiva Weerawarana(IBM),  
880 Roger Wolter(Microsoft).

881 The following individuals were members of the committee during the development of this specification:

882 Abbie Barbir(Nortel), Charlton Barreto(Adobe), Stefan Batres(Microsoft), Hamid Ben  
883 Malek(Fujitsu), Andreas Bjarlestam(Ericsson), Toufic Boubrez(Layer 7), Doug Bunting(Sun), Lloyd  
884 Burch(Novell), Steve Carter(Novell), Martin Chapman(Oracle), Dave Chappell(Sonic), Paul  
885 Cotton(Microsoft), Glen Daniels(Sonic), Doug Davis(IBM), Blake Dournaee(Intel), Jacques  
886 Durand(Fujitsu), Colleen Evans(Microsoft), Christopher Ferris(IBM), Paul Fremantle(WSO2),  
887 Robert Freund(Hitachi), Peter Furniss(Erebor), Marc Goodner(Microsoft), Alastair  
888 Green(Choreology), Mike Grogan(Sun), Ondrej Hrebicek(Microsoft), Kazunori Iwasa(Fujitsu),  
889 Chamikara Jayalath(WSO2), Lei Jin(BEA), Ian Jones(BT plc), Anish Karmarkar(Oracle), Paul  
890 Knight(Nortel), Dan Leshchiner(Tibco), Mark Little(JBoss), Lily Liu(webMethods), Matt  
891 Lovett(IBM), Ashok Malhotra(Oracle), Jonathan Marsh(Microsoft), Daniel Millwood(IBM), Jeff  
892 Mischkinsky(Oracle), Nilo Mitra(Ericsson), Peter Niblett(IBM), Duane Nickull(Adobe), Eisaku  
893 Nishiyama(Hitachi), Dave Orchard(BEA), Chouthri Palanisamy(NEC), Sanjay Patil(SAP), Gilbert  
894 Pilz(BEA), Martin Raeppe(SAP), Eric Rajkovic(Oracle), Stefan Rossmannith(SAP), Tom  
895 Rutt(Fujitsu), Rich Salz(IBM), Shivajee Samdarshi(Tibco), Vladimir Videlov(SAP), Claus von  
896 Riegen(SAP), Pete Wenzel(Sun), Steve Winkler(SAP), Ümit Yalçınalp(SAP), Nobuyuki  
897 Yamamoto(Hitachi).

## Appendix E. Revision History

Rev	Date	By Whom	What
wd-01	2006-12-31	Doug Davis	Initial version created based on MakeConnection support in the WS-RM spec
wd-02	2007-01-31	Doug Davis	Lots of typos from MarcG Updated WD number and date
wd-02	2007-02-01	Doug Davis	PR015 and PR029 applied

## 899 **Appendix F. Notices**

900 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that  
901 might be claimed to pertain to the implementation or use of the technology described in this document or  
902 the extent to which any license under such rights might or might not be available; neither does it represent  
903 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to  
904 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made  
905 available for publication and any assurances of licenses to be made available, or the result of an attempt  
906 made to obtain a general license or permission for the use of such proprietary rights by implementors or  
907 users of this specification, can be obtained from the OASIS Executive Director.

908 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or  
909 other proprietary rights which may cover technology that may be required to implement this specification.  
910 Please address the information to the OASIS Executive Director.

911 Copyright (C) OASIS Open (2007). All Rights Reserved.

912 This document and translations of it may be copied and furnished to others, and derivative works that  
913 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
914 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
915 this paragraph are included on all such copies and derivative works. However, this document itself may  
916 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as  
917 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights  
918 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it  
919 into languages other than English.

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

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