[MS-NMFMB-Diff]:

.NET Message Framing MSMQ Binding Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
3/12/2010	0.1	Major	First Release.
4/23/2010	0.1.1	Editorial	Changed language and formatting in the technical content.
6/4/2010	0.1.2	Editorial	Changed language and formatting in the technical content.
7/16/2010	0.2	Minor	Clarified the meaning of the technical content.
8/27/2010	0.3	Minor	Clarified the meaning of the technical content.
10/8/2010	0.3	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	0.3	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	0.3	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	0.3	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	0.3	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	0.3	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	0.4	Minor	Clarified the meaning of the technical content.
9/23/2011	0.4	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	1.0	Major	Updated and revised the technical content.
3/30/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	1.1	Minor	Clarified the meaning of the technical content.
10/25/2012	2.0	Major	Updated and revised the technical content.
1/31/2013	2.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	2.1	Minor	Clarified the meaning of the technical content.
11/14/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	3.0	Major	Significantly changed the technical content.
10/16/2015	3.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
7/14/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2017	4.0	Major	Significantly changed the technical content.
6/1/2017	4.0	None	No changes to the meaning, language, or formatting of the technical content.
12/1/2017	4.0	None	No changes to the meaning, language, or formatting of the technical content.
3/13/2019	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document specifies the .NET Message Framing MSMQ Binding Protocol, as well as a collection of Web service policy assertions that define behavior for the interaction with a Web service entity. This set of policy assertions pertains to an endpoint using the .NET Message Framing MSMQ Binding Protocol as the transport. This document does not define any specific Web service endpoints or message exchanges.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

best effort: Indicates that a Message Queuing System makes a best effort to meet the specified message delivery assurance, but does not raise an error if the delivery assurance is not met.

certificate: A certificate is a collection of attributes and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for authentication and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [RFC3280] and [X509] sections 7 and 8.

endpoint: A node that sends or receives a protocol stream.

envelope record: A record that contains data, such as a SOAP message. For more information about envelope records, see [SOAP1.1] and [SOAP1.2-1/2007].

express message: A volatile message that does not persist through queue manager restarts. These express messages provide best-effort, at-most-once delivery assurance.

format name: A name that is used to reference a queue when making calls to API functions.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the GUID. See also universally unique identifier (UUID).

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

initiator: The node that initiates the connection over which a protocol stream flows.

IPv4 address in string format: A string representation of an IPv4 address in dotted-decimal notation, as described in [RFC1123] section 2.1.

message: A data structure representing a unit of data transfer between distributed applications. A message has message properties, which may include message header properties, a message body property, and message trailer properties.

receiver: The node that is the receiver of the protocol stream.

recoverable message: A message that persists through queue manager restarts and provides best-effort, at-most-once delivery assurance.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

web service: A software entity that responds to SOAP messages ([SOAP1.1],.[WSDL]).

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

X.509: An ITU-T standard for public key infrastructure subsequently adapted by the IETF, as specified in [RFC3280].

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MC-MQSRM] Microsoft Corporation, "Message Queuing (MSMQ): SOAP Reliable Messaging Protocol (SRMP)".

[MC-NMF] Microsoft Corporation, ".NET Message Framing Protocol".

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-MQDMPR] Microsoft Corporation, "Message Queuing (MSMQ): Common Data Model and Processing Rules".

[MS-MQMQ] Microsoft Corporation, "Message Queuing (MSMQ): Data Structures".

[MS-MQQB] Microsoft Corporation, "Message Queuing (MSMQ): Message Queuing Binary Protocol".

[MS-WSPOL] Microsoft Corporation, "Web Services: Policy Assertions and WSDL Extensions".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, http://www.ietf.org/rfc/rfc3280.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.rfc-editor.org/rfc/rfc3986.txt

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[WS-Policy] Siddharth, B., Box, D., Chappell, D., et al., "Web Services Policy 1.2 - Framework (WS-Policy)", April 2006, http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

1.2.2 Informative References

[MS-MQOD] Microsoft Corporation, "Message Queuing Protocols Overview".

[MS-NETOD] Microsoft Corporation, "Microsoft .NET Framework Protocols Overview".

1.3 Overview

The .Net Message Framing MSMQ Binding Protocol specifies how the mechanism described in [MC-NMF] for framing messages over any transport protocol can be applied over Message Queue (MSMQ). This protocol specification also includes how to indicate the use of .NET Message Framing over MSMQ as a SOAP transport in Web Services Description Language (WSDL). Finally, the protocol details how the protocol behavior changes if any of the set of policy assertions (as defined in section 1.1 of [MS-WSPOL]) are set on an endpoint. This document specifies the following policy assertions:

- MSMQ Best-Effort

  The MSMQ Best-Effort policy assertion indicates that a Web service endpoint requires messages to be delivered in a best-effort manner, which does not guarantee that the message will be delivered only one time.

- MSMQ Session

  The MSMQ Session policy assertion indicates that a Web service endpoint requires multiple envelope records to be sent within a single message.

- MSMQ Volatile

The MSMQ Volatile policy assertion indicates that a Web service endpoint requires the use of express messages; otherwise, recoverable messages are required.

- MSMQ Authenticated

  The MSMQ Authenticated policy assertion indicates that a Web service endpoint requires MSMQ messages to be authenticated, as described in section 3.1.5.8.3 of [MS-MQQB].

- MSMQ Windows Domain

  The MSMQ Windows Domain policy assertion indicates that a Web service endpoint requires authentication to be performed using the sender's security identifier (SID), as defined in [MS-DTYP] section 2.4.2. If this policy assertion is not set, the service endpoint requires authentication to be performed using X.509 certificates [RFC3280].

These assertions are used to ensure the client is using the Web service through the binding as intended.

The .Net Message Framing MSMQ Binding Protocol makes use of MSMQ protocols to set the proper message attributes in order to conform with policy assertions set by the Web service.

1.4 Relationship to Other Protocols

This protocol uses events defined in Message Queuing (MSMQ): Common Data Model and Processing Rules [MS-MQDMPR] to transfer messages between the client and the destination endpoint.

This protocol uses MSMQ as the transport to send envelope records, as specified in section 2.2.4 of [MC-NMF].

1.5 Prerequisites/Preconditions

The .NET Message Framing MSMQ Binding Protocol requires that both the initiator and receiver satisfy all preconditions stated in section 2.4 of [MS-MQOD]. The **Queue**, as defined in section 3.1.1.2 of [MS-MQDMPR], is used for communication exists and is accessible by both the sender and receiver.

1.6 Applicability Statement

The .NET Message Framing MSMQ Binding Protocol is applicable in scenarios where an initiator and a receiver require a communication mechanism to send and receive envelope records [MC-NMF] over MSMQ. This is the case if the WSDL file contains a SOAP binding element with a transport value of "http://schemas.microsoft.com/soap/msmq" as specified in section 2.1.

1.7 Versioning and Capability Negotiation

This protocol requires .NET Message Framing Protocol version 1.0 [MC-NMF]. When this protocol is implemented by using SOAP, it requires the use of SOAP version 1.1 [SOAP1.1] or SOAP version 1.2 [SOAP1.2-1/2007].

This protocol does not support capability negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

The following sections specify the message transport and the common data types of the .Net Message Framing MSMQ Binding Protocol.

2.1 Transport

The .NET Message Framing MSMQ Binding Protocol requires MSMQ.

An endpoint that uses the .NET Message Framing MSMQ Binding Protocol with [SOAP1.2-1/2007] MUST specify the value of the transport attribute of the <soap12:binding> element [WSDL] to be "http://schemas.microsoft.com/soap/msmq".

An endpoint that uses the .NET Message Framing MSMQ Binding Protocol with [SOAP1.1] MUST specify the value of the transport attribute of the <soap:binding> element [WSDL] to be "http://schemas.microsoft.com/soap/msmq".

2.2 Common Message Syntax

2.2.1 Namespaces

This specification defines and references various XML namespaces using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and is not significant for interoperability.

Prefix	Namespace URI	Reference
msmq	http://schemas.microsoft.com/ws/06/2004/mspolicy/msmq	This specification
soap	http://schemas.xmlsoap.org/wsdl/soap	[WSDL]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[WSDL]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy	[WS-Policy]

2.2.2 Messages

This specification does not define any messages.

2.2.3 Elements

The following sections contain the XML schema description for the policy assertions defined in this document and the schema for the transport that this protocol operates on. The following elements are policy assertions and belong to the wsp:Policy tag, as described in [WS-Policy].

2.2.3.1 MSMQ Best-Effort

```
<msmq:BestEffort
    xmlns:msmq="http://schemas.microsoft.com/ws/06/2004/mspolicy/msmq" />
```

The following describes the content model of the <MsmqBestEffort> element.

/msmq:BestEffort: A Web service endpoint with MSMQ best-effort policy assertion MUST NOT transfer messages as part of a transaction which ensures exactly once guarantee. The client and service instead MUST make a best effort attempt to deliver messages.

2.2.3.2 MSMQ Session

```
<msmq:Session
    xmlns:msmq="http://schemas.microsoft.com/ws/06/2004/mspolicy/msmq" />
```

The following describes the content model of the <MsmqSession> element.

/msmq:Session: A Web service endpoint with the MSMQ session policy assertion MUST require the initiator to create a connection with the service in which numerous messages are sent within a single MSMQ message.

2.2.3.3 MSMQ Volatile

```
<msmq:Volatile
    xmlns:msmq="http://schemas.microsoft.com/ws/06/2004/mspolicy/msmq" />
```

The following describes the content model of the <MsmqVolatile> element.

/msmq:Volatile: A Web service endpoint with MSMQ volatile policy assertion MUST require the client to send express messages.

2.2.3.4 MSMQ Authenticated

```
<msmq:Authenticated
    xmlns:msmq="http://schemas.microsoft.com/ws/06/2004/mspolicy/msmq" />
```

The following describes the content model of the <MsmqAuthenticated> element.

/msmq:Authenticated: A Web service endpoint with MSMQ authenticated policy assertion MUST require MSMQ messages sent on the transport to be authenticated.

2.2.3.5 MSMQ Windows Domain

```
<msmq:WindowsDomain
    xmlns:msmq="http://schemas.microsoft.com/ws/06/2004/mspolicy/msmq" />
```

The following describes the content model of the <MsmqWindowsDomain> element.

/msmq:WindowsDomain: This assertion only applies to policies which also include the MSMQ Authenticated policy assertion. A Web service endpoint with MSMQ Windows Domain policy assertion MUST authenticate the user using Windows logon authentication; otherwise, if this assertion is not set, they MUST authenticate the user using X.509 certificates.

3 Protocol Details

A node can participate in this protocol in one of two roles: initiator or receiver. An initiator begins the process by opening and delivering messages to a queue that the receiver is monitoring. MSMQ is used as the lower-level protocol transport for performing the .NET Message Framing Protocol [MC-NMF].

3.1 Common Details

3.1.1 Abstract Data Model

3.1.1.1 NetMsmqMessage

A **NetMsmqMessage** is a structure that encapsulates the data placed into a **Message**.**Body**, as defined in section 3.1.1.12 of [MS-MQDMPR]. It contains the following attributes:

Preamble: A preamble message as defined in section 2.2.6 of the .NET Message Framing Protocol [MC-NMF].

EnvelopeRecords: A single or multiple envelope record as defined in section 2.2.4 of the .NET Message Framing Protocol [MC-NMF].

EndRecord: An end record as specified in section 2.2.3.9 of the .NET Message Framing Protocol [MC-NMF].

3.1.1.2 net.msmq URI

A **net.msmq URI** is a URI that satisfies the following constraints:

- The scheme component of the URI MUST be net.msmq.
- The URI MUST be a hierarchical URI.
- The authority component of the URI MUST be specified.
- The authority component of the URI MUST NOT include user information.
- The URI MUST NOT include the query URI component.
- The URI MUST NOT include the fragment URI component.

3.1.1.3 MQPCO

An **MQPCO** is a configuration object as defined in section 3.1.3 of [MC-NMF], but with the following constraints:

- The protocol version, section 2.2.3.1 [MC-NMF], specified by the MQPCO MUST be 1.0.
- If the MSMQ Session policy assertion described in section 2.2.3.2 is set, then the communication mode, section 2.2.3.2 [MC-NMF], MUST be Simplex.
- Otherwise, the communication mode MUST be Singleton Sized.
- The Via URI, section 2.2.3.3 of [MC-NMF], specified by the MQPCO MUST be a **net.msmq URI** as specified in 3.1.1.2.

3.1.2 Timers

None.

3.1.3 Initialization

A MQPCO with an uninitialized transport is made available to the protocol as part of a higher-layer triggered event.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

3.1.7.1 Construct Direct Format Name

This event MUST be generated with the following argument:

- *iVia*: A URI, as specified in [RFC3986].

Return Values:

- **rFormatName**: The constructed format name.

The protocol MUST construct a direct format name as specified in section 2.1.2 for [MS-MQMQ], as follows:

- Initialize **rFormat** to "DIRECT=".
- If the host section as defined in section 3.2.2 of [RFC3986], of the Via URI is localhost:
  - Append "OS:" to **rFormat**.
- Otherwise:
  - If the host section of the Via URI is an IPv4 address in string format:
    - Append "TCP:" to **rFormat**.
  - Otherwise:
    - Append "OS:" to **rFormat**.
  - Append the host portion of the Via URI to **rFormat**.
- Append "\" to **rFormat**.
- Append the relative path of the URI as defined in section 3.3 of [RFC3986] to **rFormat**.

- Set *rFormatName* to *rFormat*.

3.1.7.2 Construct Public Format Name

This event MUST be generated with the following argument:

- *iVia*: A URI as specified in [RFC3986].

Return Values:

- **rReturnStatus**: A status code signifying the success or failure of the raised [MS-MQDMPR] events.
- **rFormatName**: The constructed format name.

The protocol MUST construct a public format name as specified in section 2.1.3 of [MS-MQMQ], as follows:

- Initialize **rPathName** to *iVia*.
- Remove the "net.msmq://" prefix from **rPathName** to create a path name
- Generate a Read Directory event as specified in section 3.1.7.1.20 of [MS-MQDMPR], with the following arguments.
  - *iDirectoryObjectType* = "Queue"
  - *iFilter* = An array of the following attribute-filter expressions:
    - Queue.Pathname EQUALS **rPathName**
  - *iAttributeList* : = An array of the following queue attributes:
    - Queue.Identifier
- Set *rReturnStatus* to **rStatus**.
- If the **rStatus** returned by the Read Directory event is not set to Success:
  - Set **rReturnStatus** equal to **rStatus**.
- Set *queueIdentifier* to the Queue.Identifier of the returned queue object.
- Set *rFormat* to the following string where *queueIdentifier* is replaced by its value:
  - "PUBLIC="queueIdentifier
- Set **rFormatName** to **rFormat**.

3.1.7.3 Construct SRMP Format Name

This event MUST be generated with the following argument:

- *iVia* : A URI, as specified in [RFC3986].
- *iSecureSrmp* : A boolean value, indicating whether the transport will use "https" or "http".

Return Values:

- *rFormatName* : The constructed format name.

The initiator MUST construct a direct format name as specified in section 2.1.2 for [MS-MQMQ], as follows:

- Initialize *rFormat* to "DIRECT=".
- If *iSecureSrmp* is set to true:
  - Append "https://" to *rFormat*.
- Otherwise:
  - Append "http://" to *rFormat*.
- Append the host portion of *iVia* to *rFormat*.
- If the port number of the *iVia* URI is set:
  - Append the port number to *rFormat*.
- Append "/msmq/" to *rFormat*.
- Append the relative Path of the URI as defined in section 3.3 of [RFC3986] to *rFormat*.
- Set *rFormatName* to *rFormat*.

3.2 Initiator Details

The initiator MUST use the binding and policy assertions enforced by the Web service in the WSDL to properly construct a valid MQPCO as specified in section 3.1.1.2. To communicate with the receiver the initiator MUST send an MSMQ **Message** [MS-MQDMPR] with a **NetMsmqMessage** in the **Message.Body** to an opened queue.

3.2.1 Abstract Data Model

3.2.1.1 SendQueue

SendQueue is an **OpenQueueDescriptor** as defined in section 3.1.1.16 of [MS-MQDMPR]. The initiator MUST maintain an instance of this element referred to as **rSendQueue**.

3.2.1.2 SendNetMsmqMessage

SendNetMsmqMessage is a NetMsmqMessage structure as defined in section 3.1.1.1. The initiator MUST maintain an instance of this element referred to as **rNetMsmqMessage**.

3.2.1.3 UseActiveDirectory

UseActiveDirectory is a user-configurable element that changes how the protocol constructs a format name. This element MUST be enabled for authentication to be performed as specified in section 3.2.5.1. The initiator MUST maintain an instance of this element referred to as **rUseActiveDirectory**.

Valid values for this element are:

- **True** = Use Public format names, as specified in section 2.1.3 of [MS-MQMQ].
- **False** = Use Direct Format Names as specified in section 2.1.2 of [MS-MQMQ].

3.2.1.4 QueueTransferProtocol

QueueTransferProtocol is a user-configurable element that allows messages to be delivered using the Message Queuing (MSMQ): SOAP Reliable Messaging Protocol (SRMP) [MC-MQSRM]. The initiator MUST maintain an instance of this element referred to as **rQueueTransferProtocol**.

Valid values for this element are:

- **Native** = Use native MSMQ protocols.
- **SRMP** = Use SRMP [MC-MQSRM] over HTTP.
- **SecureSRMP** = Use SRMP [MC-MQSRM] over HTTPS.

3.2.1.5 Message

Message is defined in section 3.1.1.12 of [MS-MQDMPR]. The initiator MUST maintain an instance of this element referred to as *rMsmqMessage*.

3.2.1.6 Transaction

Transaction is defined in section 3.1.1.14 of [MS-MQDMPR]. The initiator MUST maintain an instance of this element referred to as **rTransaction**.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Initialize Session

The initiator MUST open a queue by performing the following:

- Raising the Open Queue for Send event as specified in section 3.2.7.1 with the following parameters set:
  - *iUseActiveDirectory* set to **rUseActiveDirectory**.
  - *iQueueTransferProtocol* set to **rQueueTransferProtocol**.
- If the returned **rReturnStatus** is not equal to MQ_OK (0x00000000), then an error MUST be propagated to the higher layer, and no further processing done.
- Set **rSendQueue** to the returned **rOpenQueue** .

The initiator MUST set **rMsmqMessage** equal to NULL.

3.2.4.2 Send Message

The initiator MUST perform the following steps.

- If **rMsmqMessage** is equal to NULL:

- Construct an MSMQ **Message** as specified in section 3.2.5.1 with the following parameters:
  - *iQueueTransferProtocol* set to **rQueueTransferProtocol.**
- Set *rMsmqMessage* equal to **rReturnMsmqMessage**.
- Set *rTransaction* equal to the returned **rReturnTransaction**.
- Construct a **Preamble Message**, referred to as **rPreamble** using the logic described in section 3.2.4.2 of [MC-NMF].
- Set *rNetMsmqMessage.Preamble* to **rPreamble**.

- Append the envelope record passed from the higher level protocol to **rNetMsmqMessage**.**EnvelopeRecords**.
- If the MSMQ Session policy assertion described in section 2.2.3.2 is not set:
  - Set *rMsmqMessage.Body* to **rNetMsmqMessage**.
  - Deliver **rMsmqMessage** by raising the Enqueue Message event from section 3.1.7.1.9 of [MS-MQDMPR] with the following parameters:
    - *iQueue* set to **rSendQueue**, the opened queue from section 3.2.4.1.
    - *iMessage* set to **rMsmqMessage**.
    - *iTransaction* set to **rTransaction**.
  - If the returned **rStatus**, is not equal to MQ_OK (0x00000000), then an error MUST be propagated to the higher layer.
  - Set *rMsmqMessage* equal to NULL.

3.2.4.3 Session Close

If the MSMQ Session policy assertion described in section 2.2.3.2 is set, the initiator MUST deliver the envelope records together as one MSMQ **Message** by performing the following:

- If **rMsmqMessage** is not equal to NULL:
  - Construct an end record, referred to as **rEndRecord***,* using the logic described in section 3.2.4.5 of [MC-NMF].
  - Set *rNetMsmqMessage.EndRecord* equal to **rEndRecord**.
  - Set *rMsmqMessage.Body* to **rNetMsmqMessage**.
  - Raise the Enqueue Message event from section 3.1.7.1.9 of [MS-MQDMPR] with the following parameters:
    - *iQueue* set to **rSendQueue**, the opened queue from section 3.2.4.1.
    - *iMessage* set to **rMsmqMessage**.
    - *iTransaction* set to **rTransaction**.
  - If the returned **rStatus** is not equal to MQ_OK (0x00000000), then an error MUST be propagated to the higher layer, and no further processing done.

The initiator MUST also close the queue by performing the following:

- Raise the Close Queue event from section 3.1.7.1.6 of [MS-MQDMPR], with the following parameters:
  - *iQueueDesc* set to **rSendQueue**.
- If the returned **rStatus**, is not equal to MQ_OK (0x00000000), then an error MUST be propagated to the higher layer, and no further processing done.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Constructing an MSMQ Message

This event MUST be generated with the following arguments:

- *iQueueTransferProtocol*: A **QueueTransferProtocol** value.

Return Values:

- **rReturnMsmqMessage**: An MSMQ **Message**.
- **rReturnTransaction**: An MSMQ **Transaction**.

The initiator MUST construct a **Message** as defined in section 3.1.1.12 of [MS-MQDMPR]. The **Message** MUST be constructed as specified in the following sections.

The initiator MUST set specific properties on a **Message**, depending on the policy assertions that were set as follows:

- If the Best-Effort policy assertion described in 2.2.3.1 is not set:
  - Raise the Create Transaction event, as specified in section 3.1.7.1.8 Create Transaction of [MS-MQDMPR], with the following parameters:
    - *iTransactionIdentifier* set to a unique transaction identifier GUID, as specified in [MS-DTYP] section 2.3.4.
  - Set **rReturnTransaction** equal to **rTransaction**.
- Otherwise:
  - Set **rReturnTransaction** equal to NULL
- If the Volatile policy assertion described in section 2.2.3.3 is set:
  - Set **Message.DeliveryGuarantee** to **Express** for volatile messages
- Otherwise, if the Volatile policy assertion is not set:
  - Set **Message.DeliveryGuarantee** to **Recoverable** for durable messages.
- If the Authenticated policy assertion described in section 2.2.3.5 is set:
  - Set **Message.PrivacyLevel** to **Enhanced**.
  - Set **Message.EncryptionAlgorithm** to **RC4**.
  - Set **Message.HashAlgorithm** to **SHA256**.<1>
  - If **iQueueTransferProtocol** is set to **SRMP** or **SecureSRMP**:
    - Set **Message.AuthenticationLevel** to **XmlSig**.

- Otherwise:
  - Set **Message.AuthenticationLevel** to **Sig30**.
- If the Windows Domain policy assertion described in section 2.2.3.5 is set:
  - Set **Message.SenderIdentifierType** to **Sid**, as specified in [MS-MQDMPR] section 2.
- Otherwise:
  - Set **Message.SenderIdentifierType** to **None**.
  - Set **Message.SenderCertificate** to a user-provided X.509 certificate as described in [RFC3280].

- Otherwise, if the Authenticated policy assertion is not set:
  - Set **Message.SenderIdentifierType** to **None**.
  - Set **Message.AuthenticationLevel** to **None**.
- Set **rReturnMsmqMessage** to the constructed **Message**.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

3.2.7.1 Open Queue for Send

This event MUST be generated with the following arguments:

- *iUseActiveDirectory*: A **UseActiveDirectory** value.
- *iQueueTransferProtocol*: A **QueueTransferProtocol** value.

Return Values:

- **rOpenQueue**: An OpenQueueDescriptor value for the opened queue.
- **rReturnStatus**: A status code signifying the success or failure of the raised [MS-MQDMPR] events.

The initiator MUST open the queue corresponding to the endpoint specified in the service's WSDL, which also corresponds to the **MQPCO** Via record, referred to as *rVia*. This MUST be performed before any messages are sent. To do so, the initiator MUST perform the following:

- Construct a format name, referred to as **rFormat** as follows:
  - If *iQueueTransferProtocol* is set to **SRMP**
    - Raise the Construct SRMP Format Name Event, as specified in section 3.1.7.3 with the following properties set:
      - *iVia*: Set to **rVia**.
      - *iSecureSrmp*: Set to false.
    - Set *rFormat* to the returned **rFormatName**.

- If *iQueueTransferProtocol* is set to **SecureSRMP**
  - Raise the Construct SRMP Format Name Event, as specified in section 3.1.7.3 with the following properties set:
    - *iVia*: Set to **rVia**.
    - *iSecureSrmp*: Set to true.
  - Set *rFormat* to the returned **rFormatName**.
- If *iQueueTransferProtocol* is set to **Native**
  - If *iUseActiveDirectory* is set to true:
    - Raise the Construct Public Format Name event, as specified in section 3.1.7.2 with the following properties set:
      - *iVia*: Set to **rVia**.
    - If the returned **rReturnStatus** from the Construct Public Format Name event, is not equal to MQ_OK (0x00000000), then set **rReturnStatus** to the same value, and perform no further processing.
    - Set *rFormat* to the returned **rFormatName**.
  - Otherwise, if **iUseActiveDirectory** is set to false:
    - Raise the Construct Direct Format Name event, as specified in section 3.1.7.1 with the following properties set:
      - *iVia*: Set to **rVia**.
    - Set *rFormat* to the returned **rFormatName**.
- Generate an Open Queue event as specified in section 3.1.7.1.5 of [MS-MQDMPR], with the following properties set:
  - *iFormatName*: Set to **rFormat**.
  - *iRequiredAccess*: Set to **QueueAccessType.SendAccess**.
  - *iSharedMode*: Set to **QueueShareMode.DenyNone**, as specified in section 3.1.1.17 of [MS-MQDMPR].
- Set *rOpenQueue* to the returned **rOpenQueueDescriptor**.
- Set *rReturnStatus* to the returned **rStatus**.

3.3 Receiver Details

The receiver listens on the queue discovered at the path name introduced in section 3.1.7.2. The receiver MUST open a queue at this address and process the messages as described in the following sections.

3.3.1 Abstract Data Model

3.3.1.1 ReceiveQueue

ReceiveQueue is an **OpenQueueDescriptor** as defined in section 3.1.1.16 of [MS-MQDMPR]. The receiver MUST maintain an instance of this element referred to as **rReceiveQueue**.

3.3.2 Timers

None.

3.3.3 Initialization

None.

3.3.4 Higher-Layer Triggered Events

3.3.4.1 Initialize Session

The receiver MUST open a queue by performing the following:

- Raising the Open Queue for Receive event, as specified in section 3.3.7.1.
- Set **rReceiveQueue** to the returned **rOpenQueue**.

3.3.4.2 Receive Message

The receiver MUST receive the MSMQ **Message**, referred to as **rMsmqMessage**, by performing the following:

- Raise the Dequeue Message event as specified in section 3.1.7.1.10 of Message Queuing (MSMQ) Common Data Model and Processing Rules [MS-MQDMPR].
    - Set *iQueueDesc* to **rReceiveQueue**.
    - Set *iTimeout* to the amount of time to wait for message arrival in seconds.
    - Set *iCursor* to NULL.
    - If the Best-Effort policy assertion described in 2.2.3.1 is not set:
        - Raise the Create Transaction event, as specified in section 3.1.7.1.8 Create Transaction of [MS-MQDMPR], with the following parameters:
            - *iTransactionIdentifier* set to a unique transaction identifier GUID.
        - Set *iTransaction* to **rTransaction**.
- If the returned **rStatus** is set to MQ_OK (0x00000000), then assign the returned **rMessage** to **rMsmqMessage**.
- Otherwise, an error MUST be propagated to the higher layer, and no further processing done.
    - An error MUST be propagated to the higher layer, and no further processing done.

After the MSMQ **Message** has been received from the MSMQ protocol the receiver MUST do the following:

- Assign **rMsmqMessage**.**Body** to a **NetMsmqMessage**, referred to as **rNetMsmqMessage**.
- Perform the steps for verifying the Preamble Message as specified in section 3.3.4.2 of [MC-NMF], on **rNetMsmqMessage**.**Preamble**.
- Read and propagate the **envelope record** in **rNetMsmqMessage**.**EnvelopeRecords** to a higher layer as specified in section 3.3.4.4 of [MC-NMF].

3.3.4.3 Session Close

The receiver MUST also close the queue by performing the following:

- Raise the Close Queue event from section 3.1.7.1.6 of [MS-MQDMPR], with the following parameters:
  - *iQueueDesc* set to **rReceiveQueue**.
- If the returned **rStatus**, is not equal to MQ_OK (0x00000000), then an error MUST be propagated to the higher layer, and no further processing done.

3.3.5 Message Processing Events and Sequencing Rules

None.

3.3.6 Timer Events

None.

3.3.7 Other Local Events

3.3.7.1 Open Queue for Receive

This event is raised without any arguments supplied.

Return Values:

- **rOpenQueue**: An OpenQueueDescriptor value for the opened queue.
- **rReturnStatus**: A status code signifying the success or failure of the raised [MS-MQDMPR] events.

The receiver MUST open the queue corresponding to the endpoint specified in the service's WSDL, which also corresponds to the **MQPCO** Via record, referred to as **rVia**. This MUST be performed before any messages are received. The receiver MUST perform the following:

- Construct a format name, referred to as **rFormat** as follows:
  - Raise the Construct Direct Format Name event, as specified in section 3.1.7.1 with the following properties set:
    - *iVia*: Set to **rVia**.
  - Set **rFormat** to the returned **rFormatName**.
- Generate an Open Queue event as specified in section 3.1.7.1.5 of [MS-MQDMPR], with the following properties set:
  - *iFormatName*: Set to **rFormat**.

- *iRequiredAccess*: Set to **QueueAccessType.SendAccess**, as specified in section 3.1.1.17 of [MS-MQDMPR].
- *iSharedMode*: Set to **QueueShareMode.DenyNone**, as specified in section 3.1.1.17 of [MS-MQDMPR].

- Set **rOpenQueue** to the returned **rOpenQueueDescriptor**.
- Set **rReturnStatus** to the returned **rStatus**.

4 Protocol Examples

The following packet capture shows the message payload of the Preamble message sent out while using the .NET Message Framing MSMQ Binding Protocol. The specifics of network transport are excluded from this example.

Raw Message

```
00 01 00 01 04 02 2B 6E ......+n
65 74 2E 6D 73 6D 71 3A et.msmq:
2F 2F 6C 6F 63 61 6C 68 //localh
6F 73 74 2F 70 72 69 76 ost/priv
61 74 65 2F 74 72 61 6E ate/tran
73 61 63 74 69 6F 6E 61 sactiona
6C 71 03 07                  lq..
```

Parsed Message

```
NMF: Preamble
     -NMF: Version = 1.0
             Record Type    = Version Record (0X00)
             Major Version     = 1 (0X01)
             Minor Version     = 0 (0X00)
     -NMF: Mode = Duplex Mode
             Record Type = Mode Record (0X01)
             Mode     = Singleton Sized (0X04)
     -NMF: Via = net.msmq://<QueuePathName>/
             Record Type = Via Record (0X02)
             Size = 43 (0X2B)
             Via = net.tcp://localhost/private/transactionalq/
     -NMF: Encoding = Binary Session Encoding
              Record Type = Known Encoding Record (0X03)
              Encoding = Binary Encoding (0X07)
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation the full WSDL with schemas is provided in the following section.

6.1 .Net Message Framing MSMQ Binding Protocol WSDL and Policy Assertions

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="OrderProcessorService"
                  targetNamespace="http://tempuri.org/"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:wsa10="http://www.w3.org/2005/08/addressing"
                  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
                  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
                  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
                  xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
                  xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
                  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
                  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
                  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
                  xmlns:tns="http://tempuri.org/"
                  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:i0="http://Microsoft.ServiceModel.Samples"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
                  xmlns:msmq="http://schemas.microsoft.com/ws/06/2004/mspolicy/msmq">
  <wsp:Policy wsu:Id="NetMsmqBinding_IOrderProcessor_policy">
    <wsp:ExactlyOne>
      <wsp:All>
        <!-- omitted elements -->
        <msmq:MsmqVolatile />
        <msmq:MsmqBestEffort />
        <msmq:MsmqSession />
        <msmq:Authenticated />
        <msmq:WindowsDomain />
        <!-- omitted elements -->
      </wsp:All>
    </wsp:ExactlyOne>
  </wsp:Policy>
  <!-- omitted elements -->
  <wsdl:binding name="NetMsmqBinding_IOrderProcessor" type="i0:IOrderProcessor">
    <wsp:PolicyReference URI="#NetMsmqBinding_IOrderProcessor_policy"/>
    <soap12:binding transport="http://schemas.microsoft.com/soap/msmq"/>
    <wsdl:operation name="SubmitPurchaseOrder">
      <soap12:operation
soapAction="http://SampleNamespace/IOrderProcessor/SubmitPurchaseOrder" style="document"/>
      <wsdl:input>
        <soap12:body use="literal"/>
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="OrderProcessorService">
    <wsdl:port name="NetMsmqBinding_IOrderProcessor"
binding="tns:NetMsmqBinding_IOrderProcessor">
      <soap12:address location="net.msmq://localhost/private/testq"/>
      <wsa10:EndpointReference>
        <wsa10:Address>net.msmq://localhost/private/testq</wsa10:Address>
      </wsa10:EndpointReference>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

7 (Updated Section) Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

The terms "earlier" and "later", when used with a product version, refer to either all preceding versions or all subsequent versions, respectively. The term "through" refers to the inclusive range of versions. Applicable Microsoft products are listed chronologically in this section.

This document specifies version-specific details in the Microsoft .NET Framework. For information about which versions of the .NET Framework are available in each released Windows product or as supplemental software, see [MS-NETOD] section 4.

- Microsoft .NET Framework 3.0
- Microsoft .NET Framework 3.5
- Microsoft .NET Framework 4.0
- Microsoft .NET Framework 4.5
- Microsoft .NET Framework 4.6
- Microsoft .NET Framework 4.7
- Microsoft .NET Framework 4.8

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 3.2.5.1: In Windows releases that support Microsoft .NET Framework 3.5 and later, the MsmqSecureHashAlgorithm enumeration enables an implementation to use any of the hash algorithms that follow, where SHA256 is the default.

```
public enum MsmqSecureHashAlgorithm

{

MD5,

SHA1,

SHA 256,

SHA 512

}
```

The underlying values of the MsmqSecureHashAlgorithm enum members are specified in the table that follows.

Member Name	Value
MD5	int32 (0x00000000)
SHA1	int32 (0x00000001)
SHA256	int32 (0x00000002)
SHA512	int32 (0x00000003)

For information about Windows releases that support the Microsoft .NET Framework versions, including the .NET Framework life-cycle support, see [MS-NETOD] section 4 Microsoft Implementations.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
7 Appendix B: Product Behavior	Added .NET Framework v4.8 to the applicability list.	Major

9 Index

A

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

V

W