

[MS-MQSD]: Message Queuing (MSMQ): Directory Service Discovery Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- This protocol 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 may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- Microsoft does not claim any trade secret rights in this documentation.
- Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. If you are interested in obtaining a patent license, please contact protocol@microsoft.com.
- The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

This protocol documentation is intended for use in conjunction with publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

A protocol specification 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.

Revision Summary

Date	Revision History	Revision Class	Comments
05/11/2007	0.1		MCPD Milestone 4 Initial Availability
08/10/2007	1.0	Major	Updated and revised the technical content.
09/28/2007	1.1	Minor	Updated the technical content.
10/23/2007	1.1.1	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
11/30/2007	2.0	Major	Updated and revised the technical content.
01/25/2008	2.0.1	Editorial	Revised and edited the technical content.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References	5
1.2.1	Normative References	5
1.2.2	Informative References.....	5
1.3	Protocol Overview (Synopsis).....	5
1.4	Relationship to Other Protocols.....	6
1.5	Prerequisites/Preconditions	6
1.6	Applicability Statement	6
1.7	Versioning and Capability Negotiation.....	6
1.8	Vendor-Extensible Fields	6
1.9	Standards Assignments.....	6
2	Messages	7
2.1	Transport	7
2.2	Message Syntax	7
2.2.1	TopologyPacketHeader	7
2.2.2	TopologyClientRequest	7
2.2.3	TopologyServerReply.....	9
3	Protocol Details	12
3.1	Client Details	12
3.1.1	Abstract Data Model	12
3.1.2	Timers	12
3.1.3	Initialization	12
3.1.4	Higher-Layer Triggered Events.....	13
3.1.5	Message Processing Events and Sequencing Rules	13
3.1.6	Timer Events.....	13
3.1.7	Other Local Events	13
3.2	Server Details.....	13
3.2.1	Abstract Data Model	13
3.2.2	Timers	14
3.2.3	Initialization	14
3.2.4	Higher-Layer Triggered Events.....	14
3.2.5	Message Processing Events and Sequencing Rules	14
3.2.6	Timer Events.....	15
3.2.7	Other Local Events	15
4	Protocol Example.....	16
5	Security	18
5.1	Security Considerations for Implementers	18
5.2	Index of Security Parameters	18
6	Appendix A: Windows Behavior	19
7	Index.....	22

1 Introduction

This document specifies the Message Queuing (MSMQ): Directory Service Discovery Protocol used by **MSMQ clients** to discover an accessible executing instance of an **MSMQ Directory Service server**. The MSMQ: Directory Service Discovery Protocol is not used by **MSMQ servers**.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

Active Directory (AD)
Augmented Backus-Naur Form (ABNF)
Globally Unique Identifier (GUID)
Little-Endian
Unicode

The following terms are defined in [\[MS-MQMQ\]](#):

Connected Network
ConnectedNetworkID
Enterprise Site
EnterpriseID
Message Queuing Information Store (MQIS)
MSMQ
MSMQ Client
MSMQ Directory Service
MSMQ Directory Service Server
MSMQ Routing Server
MSMQ Server
MSMQ Site
Primary Enterprise Controller (PEC)
Primary Site Controller (PSC)
SiteID

The following terms are specific to this document:

Internetwork Packet Exchange (IPX): A protocol maintained by Novell's NetWare product that provides connectionless datagram delivery of messages. The **Internetwork Packet Exchange (IPX)** is based on Xerox Corporation's Internetwork Packet protocol, XNS.

Queue: A data structure containing an ordered list of zero or more messages. A **queue** has a head and a tail, and supports a first-in-first-out (FIFO) access pattern. A message is appended to the tail through a write operation (Send) that appends the message and increments the tail pointer. A message is consumed from the head through a destructive read operation (Receive) that deletes the message and increments the head pointer. A message at the head may also be read through a non-destructive read operation (Peek).

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

1.2 References

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[IANAPORT] Internet Assigned Numbers Authority, "Port Numbers", November 2006, <http://www.iana.org/assignments/port-numbers>

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](#)", January 2007.

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](#)", January 2007.

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

[MS-MQMP] Microsoft Corporation, "[Message Queuing \(MSMQ\): Queue Manager Client Protocol Specification](#)", August 2007.

[MS-MQMQ] Microsoft Corporation, "[Message Queuing \(MSMQ\): Data Structures](#)", August 2007.

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

[RFC4234] Crocker, D., Ed. and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005, <http://www.ietf.org/rfc/rfc4234.txt>

1.2.2 Informative References

[IPX] Microsoft Corporation, "Internetwork Packet Exchange (IPX)", <http://msdn2.microsoft.com/en-us/library/ms817906.aspx>

1.3 Protocol Overview (Synopsis)

MSMQ clients use the Message Queuing (MSMQ): Directory Service Discovery Protocol to obtain a current list of network accessible MSMQ servers running the **MSMQ Directory Service**. Once an MSMQ Directory Service Server has been located, other protocols are used to obtain **MSMQ** configuration information such as **queues** and machine names containing MSMQ installations.

MSMQ clients obtain a list of available MSMQ Directory Service Servers by broadcasting (via a connectionless transport) a [TopologyClientRequest \(section 2.2.2\)](#) packet and by receiving [TopologyServerReply \(section 2.2.3\)](#) packets.

Lists of MSMQ Directory Service Servers are contained in **Primary Site Controllers (PSC)** and **Primary Enterprise Controllers (PEC)**; PSCs and PECs generate TopologyServerReply packets in response to TopologyClientRequest packets.

A TopologyClientRequest packet contains a unique identifier of the **enterprise site** of which the MSMQ client is a member, [<1>](#) a unique correlation identifier for the request, and an identifier of the current **MSMQ site** (if any) of which the client is a member. [<2>](#) A TopologyServerReply packet contains the unique request correlation identifier assigned by the MSMQ client and a list of known MSMQ servers. In the case in which the MSMQ client is already a member of the same MSMQ site as

the responding MSMQ server, the TopologyServerReply packet does not contain a list of MSMQ servers.

1.4 Relationship to Other Protocols

The Message Queuing (MSMQ): Directory Service Discovery Protocol depends on the UDP over IP or **IPX <3>** protocols for sending discovery requests and for receiving discovery replies. The MSMQ: Directory Service Discovery Protocol is not used by MSMQ servers.

The [Message Queuing \(MSMQ\): Queue Manager Client Protocol](#), as specified in [MS-MQMP], makes use of the information obtained by an [MS-MQSD] client request.

1.5 Prerequisites/Preconditions

MSMQ clients and MSMQ servers must be configured with the port number **<4>** that has been assigned for use by the MSMQ site for the Message Queuing (MSMQ): Directory Service Discovery Protocol.

1.6 Applicability Statement

The Message Queuing (MSMQ): Directory Service Discovery Protocol is applicable to all versions of MSMQ servers and is the only directory service discovery protocol used by MSMQ version 1 and MSMQ version 2 clients. The MSMQ: Directory Service Discovery Protocol has been deprecated for MSMQ version 3 and MSMQ version 4 clients but must still be implemented by all MSMQ servers for support of MSMQ version 1 and MSMQ version 2 clients.

1.7 Versioning and Capability Negotiation

A version number is present in the [TopologyPacketHeader \(section 2.2.1\)](#) packet, but it is not used.

1.8 Vendor-Extensible Fields

There are no vendor-extensible fields available in the Message Queuing (MSMQ): Directory Service Discovery Protocol.

1.9 Standards Assignments

No standards assignments have been made for the Message Queuing (MSMQ): Directory Service Discovery Protocol and its data structures. Port number 1801 for UDP and TCP has been registered with the Internet Assigned Numbers Authority (IANA) by Microsoft Corporation for MSMQ protocols, as specified in [\[IANAPORT\]](#).

2 Messages

The following sections specify how Message Queuing (MSMQ): Directory Service Discovery Protocol messages are transported and MSMQ: Directory Service Discovery Protocol message syntax.

2.1 Transport

Connectionless communications **MUST** be used, and either UDP over IP or IPX MAY [<5>](#) be used.

2.2 Message Syntax

Message Queuing (MSMQ): Directory Service Discovery Protocol messages are formatted as either UDP or IPX packets.

2.2.1 TopologyPacketHeader

A TopologyPacketHeader packet is sent as the first element of every [TopologyClientRequest \(section 2.2.2\)](#) packet and every [TopologyServerReply \(section 2.2.3\)](#) packet.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version								Type								Reserved															

Version (1 byte): An eight-bit value that specifies the TopologyPacketHeader version. MSMQ clients **MUST** set all bit positions to 0. MSMQ servers **MUST** ignore this field.

Type (1 byte): An eight-bit value that **MUST** specify the type of the packet that follows. A value of 0x01 **MUST** be used to specify a TopologyClientRequest (section 2.2.2). A value of 0x02 **MUST** be used to specify a TopologyServerReply (section 2.2.3).

Value	Meaning
0x01	Specifies a TopologyClientRequest (section 2.2.2).
0x02	Specifies a TopologyServerReply (section 2.2.3).

Reserved (2 bytes): A 16-bit value that is not used, and **MUST** be set to 0x0000 by MSMQ clients, and **MUST** be ignored by MSMQ servers.

2.2.2 TopologyClientRequest

A TopologyClientRequest packet **MUST** be prefixed with a [TopologyPacketHeader](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
TopologyPacketHeader																															
EnterpriseID																															

...
...
...
RequestID
...
...
...
SiteID
...
...
...
IPXNetworkCount (optional)
IPXNetworkNumberArray (variable)
...

TopologyPacketHeader (4 bytes): As specified in section [2.2.1](#).

EnterpriseID (16 bytes): A **GUID** that MUST contain the identifier for the **enterprise** to which the client belongs.

RequestID (16 bytes): A GUID that MUST uniquely identify a request. This value is used to correlate TopologyClientRequest packets to [TopologyServerReply](#) packets. The CorrelationID field of a TopologyServerReply that is generated in response to a TopologyClientRequest packet MUST contain this value.

SiteID (16 bytes): A GUID that MUST contain the identifier for the MSMQ site to which the client belongs.

IPXNetworkCount (4 bytes): A 32-bit integer value in **little-endian** order. When IPX networking is being used, this value MUST specify the number of IPXNetworkNumberArray entries that follow. When IP networking is being used, this field MUST NOT be present. When present, the value MUST be in the range 1 to 32, inclusive.

IPXNetworkNumberArray (variable): An array of 32-bit integer values in little-endian order. Each entry specifies an IPX network address. When IP networking is being used, this array MUST NOT be present. When present, the number of entries is specified by the IPXNetworkCount field.

2.2.3 TopologyServerReply

A TopologyServerReply packet MUST be prefixed with a [TopologyPacketHeader](#).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
TopologyPacketHeader																															
CorrelationID																															
...																															
...																															
...																															
ConnectedNetworkCount																															
ConnectedNetworkMask																															
DirectoryServiceServerSize																															
ConnectedNetworkArray (variable)																															
...																															
RespondingSiteID (optional)																															
...																															
...																															
...																															
DirectoryServiceServerArray (variable)																															

...

TopologyPacketHeader (4 bytes): As specified in section [2.2.1](#).

CorrelationID (16 bytes): A GUID that MUST contain the value from the **RequestID** field from the [TopologyClientRequest](#) packet for which the TopologyServerReply is generated.

ConnectedNetworkCount (4 bytes): A 32-bit integer value in little-endian order. The value specifies the number of **connected network** IDs that are present in the **ConnectedNetworkArray**. The value MUST be in the range 1 to 32, inclusive.

ConnectedNetworkMask (4 bytes): This 32-bit field is treated as 32 Boolean values. This entire field MUST be set to all zeros if the networking protocol being used is IP. When the networking protocol used is IPX, the count of 1-valued bit positions MUST equal the value contained in the **ConnectedNetworkCount** field.

DirectoryServiceServerSize (4 bytes): A 32-bit integer value in little-endian order. The value of this field MUST specify the number of bytes occupied by the **DirectoryServiceServerArray**. If the **SiteID** assigned to the responding server matches the SiteID contained in the TopologyClientRequest, this value MUST be zero, and the **RespondingSiteID** and **DirectoryServiceServerArray** fields MUST NOT be present.

ConnectedNetworkArray (variable): An array of GUIDs that contains a list of directory service server **ConnectedNetworkIDs**. The number of ConnectedNetworkIDs contained in this array (and thus its size) is determined by the value contained in the **ConnectedNetworkCount** field. The entries consist of ConnectedNetworkIDs of MSMQ Directory Service servers.

RespondingSiteID (16 bytes): If the **DirectoryServiceServerSize** field is not zero, this field MUST be present, and MUST contain the SiteID GUID of the responding server. This field MUST NOT be present if the value of the **DirectoryServiceServerSize** field is zero.

DirectoryServiceServerArray (variable): If the **DirectoryServiceServerSize** field is not zero, this field MUST contain an array of **Unicode** characters. The array MUST contain a comma (value 0x2C00) separated list of MSMQ Directory Service servers and an indication of the networking protocol(s) used by the named MSMQ Directory Service server. The first character of each delimited entry MUST specify whether the following named MSMQ Directory Service server supports the IP networking protocol where a character value of 1 (0x3100) indicates support, and a character value of 0 (0x3000) indicates otherwise. The second character of each delimited entry MUST specify whether the following named MSMQ Directory Service server supports the IPX networking protocol where a character value of 1 (0x3100) indicates support, and a character value of 0 (0x3000) indicates otherwise. The third character of each entry is the first character of a variable-length character string containing the name of an MSMQ Directory Service server. Each entry is a machine name, and the comma (value 0x2C00) or null (value 0x0000) characters MUST NOT be allowed as part of the name; the end of an MSMQ Directory Service server name is delimited by a comma (value 0x2C00) or a null character (value 0x0000), which is not part of the MSMQ Directory Service server name.

A **DirectoryServiceServerArray** MUST be formatted according to the following **Augmented Backus-Naur Form (ABNF)** rules:

```
DirectoryServiceServerArray = Entry 0*ContinuedEntry Endlist
Supported = %x31.00 ; A Unicode 1 indicates is supported
NotSupported = %x30.00 ; A Unicode 0 indicates not supported
IP = Supported / NotSupported ; IP networking
IPX = Supported / NotSupported ; IPX networking
R1 = %x01-2b ; Range 1
R2 = %x2c-2c ; Range 2 is x2c only
R3 = %x00-ff ; Range 3
R4 = %x01-ff ; Range 4
R5 = %x2d-ff ; Range 5
R6 = %x00-00 ; Range 6 is x00 only
X1 = R1 R3 ; Two hex digit range 1
X2 = R2 R4 ; Two hex digit range 2
X3 = R5 R3 ; Two hex digit range 3
X4 = R6 R4 ; Two hex digit range 4
NameChar = X1 / X2 / X3 / X4 ; Name character: no commas or nulls
Delim = %x2c.00 ; Use comma delimiter when more follow
EndList = %x00.00 ; Use null for end of list
Entry = IP IPX 1*NameChar ; The layout for the final entry
ContinuedEntry = Delim Entry ; The layout for one continued entry
```

3 Protocol Details

The following sections specify details of the Message Queuing (MSMQ): Directory Service Discovery Protocol, including abstract data models and message processing rules.

3.1 Client Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document.

The client participant of the Directory Service Discovery Protocol MUST maintain the following information:

- A version identifier that MUST contain the value 0 (see section [2.2.1](#)).
- A type identifier that is used to distinguish between client requests and server replies (see section [2.2.1](#)).
- A port number to which the protocol packet is broadcast (see section [3.2.3](#)).
- An **EnterpriseID** that is used to identify the MSMQ Enterprise of which the MSMQ client is a member (see section [2.2.2](#)).[.<6>](#)
- A SiteID that is used to identify the MSMQ Site of which the MSMQ client is a member (see section [2.2.2](#)).[.<7>](#)
- A list of IP and/or IPX network addresses that are used by the MSMQ client (see section [2.2.2](#)).[.<8>](#)
- A count of the IPX network addresses that are used by the MSMQ client (see section [2.2.2](#)).
- A list of directory service server names (see section [2.2.3](#)).[.<9>](#)

3.1.2 Timers

There is a 15 second time-out value for receiving responses to the broadcast of the [TopologyClientRequest](#).

3.1.3 Initialization

When the client process is initialized, it attempts to obtain its local server SiteID and its EnterpriseID.[.<10>](#) If the client process is unable to obtain its local server SiteID and its EnterpriseID, it must log an error and terminate.

When the client process successfully obtains the SiteID and EnterpriseID, the MSMQ client broadcasts a [TopologyClientRequest](#) if any of the following conditions are met: the client is attempting to complete its first initialization after installation; the client IP address has changed since the last successful initialization; or a client IP address is a Remote Access Service address.

When a client successfully completes initialization and a broadcast is required, then it proceeds as described in section [3.1.5](#). If initialization does not complete successfully, the client MUST attempt to initialize again at periodic intervals. [<11>](#)

3.1.4 Higher-Layer Triggered Events

There are no higher-layer triggered events.

3.1.5 Message Processing Events and Sequencing Rules

If the conditions described in section [3.1.3](#) are met, then a [TopologyClientRequest](#) is broadcast on every network address assigned to the client machine. If valid responses are received on multiple assigned network addresses, the information stored will be what is received from the last assigned network address providing a valid response.

If an improperly formatted [TopologyServerReply](#) is received in response to the broadcast of the [TopologyClientRequest](#), it MUST be ignored, and processing MUST proceed as if the response had not been received.

When a properly formatted [TopologyServerReply](#) is received from the current local server controller site, no new information is provided, and the local list of Directory Service servers is not updated.

When a properly formatted [TopologyServerReply](#) is received from a controller server site that is not the current local server site, the client MUST wait for either a second [TopologyServerReply](#) packet to be received or a receive time out to occur. When a receive time out occurs, or the second properly formatted [TopologyServerReply](#) packet received is from a site other than the current local server site, the first [TopologyServerReply](#) packet information MUST be used to update the local lists. When the second properly formatted [TopologyServerReply](#) is received from the current local server site, the local list of Directory Service servers MUST NOT be updated.

3.1.6 Timer Events

If a response to a [TopologyClientRequest](#) broadcast packet is not received within 15 seconds, a return code indicating that a directory server list has not been obtained is returned to the invoker of the internal function that broadcast the [TopologyClientRequest](#) packet.

3.1.7 Other Local Events

There are no local events.

3.2 Server Details

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with what is described in this document.

The server participant of the Directory Service Discovery Protocol MUST maintain the following information:

- A version identifier that MUST contain the value 0 (see section [2.2.1](#)).

- A type identifier that is used to distinguish between client requests and server replies (see section [2.2.1](#)).
- A port number to be used to listen for the protocol packet (see section [3.2.3](#)).
- A SiteID that is used to identify the MSMQ Site of which the MSMQ server is a member (see section [2.2.3](#)).<12>
- A list of ConnectedNetworkIDs (see section [2.2.3](#)).
- A list of directory service server names (see section [2.2.3](#)).<13>

3.2.2 Timers

There are no timers.

3.2.3 Initialization

When the server begins execution, it MUST use the agreed on directory service discovery port (which is by default, 1801 for IP networking and 876 for IPX networking) to listen for client requests. Typically, a sockets programming *select* operation is used as the means for listening for the client requests.

3.2.4 Higher-Layer Triggered Events

There are no higher-layer triggered events.

3.2.5 Message Processing Events and Sequencing Rules

On receiving a [TopologyClientRequest](#) message, the server sends a [TopologyServerReply](#) packet and then immediately returns to listening for another TopologyClientRequest message. In the event of a network error during a send operation, the server MUST close and re-open the socket and attempt to listen for incoming packets. If the socket cannot be opened, the server thread MUST terminate.

In the event of a network error during a receive operation, the server MUST ignore the incoming data and return to listening for incoming data. The receive operation MUST be prepared to receive a fully populated TopologyClientRequest, and MUST acquire the actual length of the data packet received. If the length of the data packet received is less than the length of a TopologyClientRequest, including the **IPXNetworkCount** field and at least one **IPXNetworkNumberArray** field entry, the TopologyClientRequest **IPXNetworkCount** and **IPXNetworkNumberArray** fields should be considered as not present, and MUST be ignored.

The value of the TopologyServerReply **ConnectedNetworkCount** field MUST be set to the number of entries to be placed into the TopologyServerReply **ConnectedNetworkArray** field. The value MUST be in the range 1 to 32, inclusive.

If the TopologyClientRequest received is an IPX packet, the TopologyServerReply **ConnectedNetworkMask** field MUST have an individual bit set to 1 for every entry that is placed into the TopologyServerReply **ConnectedNetworkArray** field. The number of bits set to 1 MUST be equal to the value in the TopologyServerReply **ConnectedNetworkCount** field.

The TopologyServerReply **DirectoryServiceServerSize** field MUST be set to the size in bytes of the data that is placed into the TopologyServerReply **DirectoryServiceServerArray** field.

If the SiteID of the receiving server matches the SiteID value in the received TopologyClientRequest **SiteID** field: the first and only entry of the TopologyServerReply **ConnectedNetworkArray** MUST

be set to the value of the responding server's MSMQ Directory Service server's **ConnectedNetworkID**, a value of zero MUST be placed in the **TopologyServerReply** **DirectoryServiceServerSize** field, and the **RespondingSiteID** and **DirectoryServiceServerArray** fields MUST NOT be included in the **TopologyServerReply** response packet.

If the **SiteID** of the receiving server does not match the **SiteID** value in the received **TopologyClientRequest** **SiteID** field:

- Available **ConnectedNetworkID** entries MUST be placed into the **TopologyServerReply** **ConnectedNetworkArray** field. The **RespondingSiteID** field in the **TopologyServerReply** MUST be populated with the receiver's **SiteID**.
- The **DirectoryServiceServerArray** in the **TopologyServerReply** MUST be populated with the locally-known list of Directory Service servers, and its size MUST be placed in the **DirectoryServiceServerSize** field.

3.2.6 Timer Events

There are no timer events.

3.2.7 Other Local Events

There are no other local events.

4 Protocol Example

A client broadcasts a [TopologyClientRequest](#) packet and receives zero or more [TopologyServerReply](#) packets.

In a trivial example, the only response is from the current site server. In this case, the value in the DirectoryServiceServerCount field of the TopologyServerReply contains a zero value.

In a more typical example, there are two or more responses: one from the PSC of the site to which the client belongs and one or more responses from other PSCs.

The following sequence diagram shows a typical example.

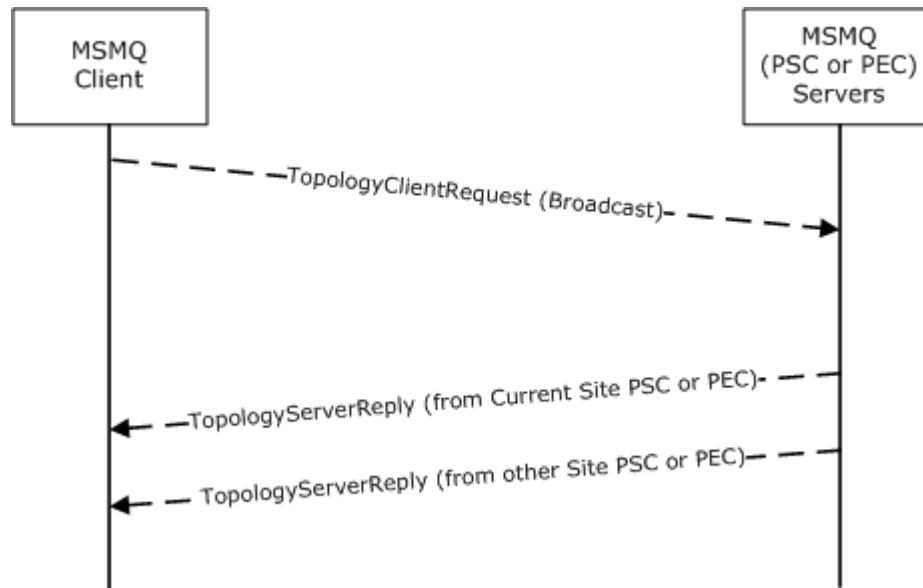


Figure 1: Typical protocol example

Examples of the content of a TopologyClientRequest and two TopologyServerReply responses sent over TCP/IP are presented below.

The TopologyClientRequest data packet has the following content.

Offset in packet	Hexadecimal data value
00000000	00 01 00 00 61 BA EA E6 C6 D1 DB 11 BA AC 00 03
00000010	FF 4E 2D 22 03 A1 91 F2 3C E3 4F AB A9 30 BE 3A
00000020	33 E4 32 DD F6 1B C5 DC AD D4 43 45 87 39 71 56
00000030	8E 8F 91 28

The first TopologyServerReply data packet is from the local site server and does not contain a list of MSMQ Directory Service servers. The packet has the following content.

Offset in packet	Hexadecimal data value
00000000	00 02 00 00 03 A1 91 F2 3C E3 4F AB A9 30 BE 3A
00000010	33 E4 32 DD 01 00 00 00 00 00 00 00 00 00 00 00
00000020	62 BA EA E6 C6 D1 DB 11 BA AC 00 03 FF 4E 2D 22

The second TopologyServerReply data packet is from a site server of which the client is not a member, and thus contains a list of MSMQ Directory Service servers; this packet is ignored because the requestor has received a reply from a server in the site of which the requestor is a member. The packet has the following content.

Offset in packet	Hexadecimal data value
00000000	00 02 00 00 03 A1 91 F2 3C E3 4F AB A9 30 BE 3A
00000010	33 E4 32 DD 01 00 00 00 00 00 00 00 00 12 00 00 00
00000020	62 BA EA E6 C6 D1 DB 11 BA AC 00 03 FF 4E 2D 22
00000030	60 BA EA E6 C6 D1 DB 11 BA AC 00 03 FF 4E 2D 22
00000040	31 00 30 00 6E 00 74 00 34 00 70 00 65 00 63 00
00000050	00 00

5 Security

The following sections specify security considerations for implementers of the Message Queuing (MSMQ): Directory Service Discovery Protocol.

5.1 Security Considerations for Implementers

The Message Queuing (MSMQ): Directory Service Discovery Protocol has no explicit security facilities. External security such as IPsec can be used to encrypt the packets as they flow in an IP network.

5.2 Index of Security Parameters

None used.

6 Appendix A: Windows Behavior

Support for the IPX networking protocol is available only in versions 1 and 2 of MSMQ; all other versions do not support IPX.

The information in this specification is applicable to the following versions of Windows:

- Windows NT
- Windows 95
- Windows 2000

The information in this specification is applicable to the following versions of Windows servers:

- Windows NT
- Windows 2000 Server
- Windows Server 2003
- Windows Server 2008

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

[<1> Section 1.3:](#) For a non-clustered MSMQ installation, the enterprise site identifier for an MSMQ site is a GUID obtained by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache\EnterpriseId.

For a clustered MSMQ installation, the enterprise site identifier for an MSMQ site is a GUID obtained by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\ClusteredQMs\servicename\Parameters\MachineCache\EnterpriseId where servicename is replaced by the Windows service names of the installed MSMQ clusters.

[<2> Section 1.3:](#) For a non-clustered MSMQ installation, the MSMQ site identifier for an MSMQ site is a GUID obtained by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache\SiteId.

For a clustered MSMQ installation, the MSMQ site identifier for an MSMQ site is a GUID obtained by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\ClusteredQMs\servicename\Parameters\MachineCache\SiteId where servicename is replaced by the Windows service names of the installed MSMQ clusters.

[<3> Section 1.4:](#) The IPX protocol is not supported on Windows 95.

[<4> Section 1.5:](#) On Windows machines, the value for the directory service discovery port number is read from the Windows registry. If the MSMQ client or MSMQ server is unable to read the value from the registry, default values of 1801 and 876 are used for IP and IPX, respectively.

For a non-clustered MSMQ installation, the IP port number is acquired by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MsmqIpPort.

For a clustered MSMQ installation, the IP port number is acquired by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\ClusteredQMs\servicename\Parameters\MsmqIPPort where servicename is replaced by the Windows service names of the installed MSMQ clusters.

For a non-clustered MSMQ installation, the IPX port number is acquired by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MsmqIPxPort.

For a clustered MSMQ installation, the IPX port number is acquired by reading the registry with a value name of HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\ClusteredQMs\servicename\Parameters\MsmqIPxPort where servicename is replaced by the Windows service names of the installed MSMQ clusters.

[<5> Section 2.1:](#) IPX is supported by MSMQ only on MSMQ version 1 and MSMQ version 2, regardless of the Windows operating system level. .

[<6> Section 3.1.1:](#) On Windows clients, the values for SiteID and EnterpriseID are read from the Windows registry.

[<7> Section 3.1.1:](#) On Windows clients, the values for SiteID and EnterpriseID are read from the Windows registry.

[<8> Section 3.1.1:](#) This list is generated for IPX by calling the sockets function getsockopt with parameters as follows:

CALL getsockopt with socket_descriptor, NSPROTO_IPX, IPX_ADDRESS, &addressAdapter, &cbOpt

The IPX adapter list is in the buffer denoted by addressAdapter pointer.

This list is generated for IP by calling gethostbyname function with parameters as follows:

CALL gethostbyname with client_machine_name RETURNING host_structure

The list of IP addresses is the h_addr_list in the returned value (if not null).

[<9> Section 3.1.1:](#) Windows clients maintain this information as a DirectoryServiceServerArray (see section [2.2.3](#)). This information is used by the [Message Queuing \(MSMQ\): Queue Manager Client Protocol](#), as specified in [\[MS-MQMP\]](#). This information is stored in the Windows registry at HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache\MQISServer.

[<10> Section 3.1.3:](#) On Windows clients, the values for SiteID and EnterpriseID are read from the Windows registry.

[<11> Section 3.1.3:](#) On Windows clients, if an EnterpriseID and a SiteID are found, the initialization is attempted every 2 minutes until successful.

[<12> Section 3.2.1:](#) In MSMQ version 1, the value for SiteID is read from the Windows registry; for MSMQ version 2, the value for SiteID is read from the **Message Queuing Information Store (MQIS)** database.

[<13> Section 3.2.1:](#) The list of server names is stored in the Directory Service. Windows NT employs the Message Queuing Information Store (MQIS) database, which runs on top of SQL Server as the Directory Service. In Windows 2000, Windows XP, Windows Server 2003, Windows Vista and Windows Server 2008, the MSMQ Directory Service is exposed by **Active Directory**.

In Windows NT and Windows 2000, the **MSMQ Routing Server** attempts to read the list of server names from the Windows Registry at

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSMQ\Parameters\MachineCache\MQISServer. If the MSMQ Routing Server does not find this key in the Windows Registry, it reads the list of server names from the MQIS database. The list of server names is populated and maintained by an MSMQ administrator.

7 Index

A

Abstract data model

[client](#)

[server](#)

[Applicability](#)

C

[Capability negotiation](#)

Client

[abstract data model](#)

[higher-layer triggered events](#)

[initialization](#)

[local events](#)

[message processing](#)

[overview](#)

[sequencing rules](#)

[timer events](#)

[timers](#)

D

Data model - abstract

[client](#)

[server](#)

E

[Example](#)

F

[Fields - vendor-extensible](#)

G

[Glossary](#)

H

Higher-layer triggered events

[client](#)

[server](#)

I

[Implementers - security considerations](#)

[Informative references](#)

Initialization

[client](#)

[server](#)

[Introduction](#)

L

Local events

[client](#)

[server](#)

M

Message processing

[client](#)

[server](#)

Messages

[overview](#)

[syntax](#)

[transport](#)

N

[Normative references](#)

O

[Overview \(synopsis\)](#)

P

[Parameters - security](#)

[Preconditions](#)

[Prerequisites](#)

R

References

[informative](#)

[normative](#)

[overview](#)

[Relationship to other protocols](#)

S

[Security](#)

Sequencing rules

[client](#)

[server](#)

Server

[abstract data model](#)

[higher-layer triggered events](#)

[initialization](#)

[local events](#)

[message processing](#)

[overview](#)

[sequencing rules](#)

[timer events](#)

[timers](#)

[Standards assignments](#)

[Syntax - message](#)

T

Timer events

[client](#)

[server](#)

Timers

[client](#)

[server](#)

[TopologyClientRequest packet](#)

[TopologyPacketHeader packet](#)

[TopologyServerReply packet](#)

[Transport - message](#)

Triggered events - higher-layer

[client](#)

[server](#)

V

[Vendor-extensible fields](#)

[Versioning](#)

W

[Windows behavior](#)