

# [MS-MQMQ]: Message Queuing (MSMQ): Data Structures

---

## 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](mailto: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		MCPPE Milestone 4 Initial Availability
08/10/2007	1.0	Major	Updated and revised the technical content.
09/28/2007	1.0.1	Editorial	Revised and edited the technical content.
10/23/2007	1.0.2	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
11/30/2007	1.0.3	Editorial	Revised and edited the technical content.
01/25/2008	1.0.4	Editorial	Revised and edited the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>8</b>
1.1	Glossary .....	8
1.2	References .....	13
1.2.1	Normative References .....	13
1.2.2	Informative References.....	14
1.3	Relationship to Protocols and Other Structures .....	15
1.4	Applicability Statement .....	15
1.5	Versioning and Localization.....	15
1.6	Vendor-Extensible Fields .....	15
<b>2</b>	<b>Definitions and Structures .....</b>	<b>16</b>
2.1	MSMQ Queue Names .....	16
2.1.1	Path Names .....	16
2.1.2	Direct Format Names .....	16
2.1.3	Public Format Names.....	17
2.1.4	Private Format Names .....	18
2.1.5	Distribution List Format Names .....	18
2.1.6	Machine, Connector, and Multicast Format Names .....	19
2.1.7	Multiple-Element Format Names .....	19
2.2	Structures.....	20
2.2.1	MQDSPUBLICKEY .....	20
2.2.2	MQDSPUBLICKEYS .....	20
2.2.3	SECURITY_INFORMATION .....	21
2.2.4	TA_ADDRESS .....	21
2.2.4.1	IP Address.....	22
2.2.4.2	IPX Address.....	23
2.2.5	SEQUENCE_INFO .....	23
2.2.5.1	SEQUENCE_INFO (Packet).....	23
2.2.6	QUEUE_FORMAT_TYPE .....	24
2.2.7	QUEUE_FORMAT .....	25
2.2.8	OBJECTID .....	27
2.2.9	DL_ID .....	27
2.2.10	MULTICAST_ID.....	27
2.2.11	QUEUE_SUFFIX_TYPE.....	28
2.2.12	PROPVARIANT Type Constants.....	28
2.2.12.1	VARTYPE.....	29
2.2.13	PROPVARIANT .....	30
2.2.13.1	tag_inner_PROPVARIANT.....	30
2.2.13.2	PROPVARIANT .....	31
2.2.14	VARIANT_BOOL.....	31
2.2.15	BLOB .....	31
2.2.16	COUNTEDARRAY.....	32
2.2.16.1	CAUB.....	32
2.2.16.2	CAUI .....	32
2.2.16.3	CAL.....	32
2.2.16.4	CAUL.....	32
2.2.16.5	CAUH .....	33
2.2.16.6	CACLSID.....	33
2.2.16.7	CALPWSTR .....	33
2.2.16.8	CAPROPVARIANT.....	34
2.2.17	ULARGE_INTEGER .....	34
2.3	PROPID .....	34

2.3.1	Queue Property Identifiers .....	35
2.3.1.1	PROPID_Q_INSTANCE .....	35
2.3.1.2	PROPID_Q_TYPE .....	35
2.3.1.3	PROPID_Q_PATHNAME .....	35
2.3.1.4	PROPID_Q_JOURNAL .....	35
2.3.1.5	PROPID_Q_QUOTA .....	35
2.3.1.6	PROPID_Q_BASEPRIORITY .....	36
2.3.1.7	PROPID_Q_JOURNAL_QUOTA .....	36
2.3.1.8	PROPID_Q_LABEL .....	36
2.3.1.9	PROPID_Q_CREATE_TIME .....	36
2.3.1.10	PROPID_Q_MODIFY_TIME .....	36
2.3.1.11	PROPID_Q_AUTHENTICATE .....	36
2.3.1.12	PROPID_Q_PRIV_LEVEL .....	37
2.3.1.13	PROPID_Q_TRANSACTION .....	37
2.3.1.14	PROPID_Q_QMID .....	37
2.3.1.15	PROPID_Q_MASTERID .....	37
2.3.1.16	PROPID_Q_FULL_PATH .....	38
2.3.1.17	PROPID_Q_NAME_SUFFIX .....	38
2.3.1.18	PROPID_Q_PATHNAME_DNS .....	38
2.3.1.19	PROPID_Q_MULTICAST_ADDRESS .....	38
2.3.1.20	PROPID_Q_ADS_PATH .....	39
2.3.1.21	PROPID_Q_SECURITY .....	39
2.3.1.22	PROPID_Q_OBJ_SECURITY .....	39
2.3.1.23	PROPID_Q_SECURITY_INFORMATION .....	39
2.3.2	Machine Property Identifiers .....	39
2.3.2.1	PROPID_QM_SITE_ID .....	39
2.3.2.2	PROPID_QM_MACHINE_ID .....	40
2.3.2.3	PROPID_QM_PATHNAME .....	40
2.3.2.4	PROPID_QM_ENCRYPTION_PK .....	40
2.3.2.5	PROPID_QM_CNS .....	40
2.3.2.6	PROPID_QM_OUTFRS .....	40
2.3.2.7	PROPID_QM_INFRS .....	40
2.3.2.8	PROPID_QM_SERVICE .....	41
2.3.2.9	PROPID_QM_QUOTA .....	41
2.3.2.10	PROPID_QM_JOURNAL_QUOTA .....	41
2.3.2.11	PROPID_QM_MACHINE_TYPE .....	41
2.3.2.12	PROPID_QM_CREATE_TIME .....	41
2.3.2.13	PROPID_QM_MODIFY_TIME .....	42
2.3.2.14	PROPID_QM_FOREIGN .....	42
2.3.2.15	PROPID_QM_OS .....	42
2.3.2.16	PROPID_QM_FULL_PATH .....	42
2.3.2.17	PROPID_QM_SITE_IDS .....	43
2.3.2.18	PROPID_QM_OUTFRS_DN .....	43
2.3.2.19	PROPID_QM_INFRS_DN .....	43
2.3.2.20	PROPID_QM_SERVICE_ROUTING .....	43
2.3.2.21	PROPID_QM_SERVICE_DSSERVER .....	43
2.3.2.22	PROPID_QM_SERVICE_DEPCLIENTS .....	44
2.3.2.23	PROPID_QM_ENCRYPTION_PK_BASE .....	44
2.3.2.24	PROPID_QM_ENCRYPTION_PK_ENHANCED .....	44
2.3.2.25	PROPID_QM_PATHNAME_DNS .....	44
2.3.2.26	PROPID_QM_OBJ_SECURITY .....	44
2.3.2.27	PROPID_QM_SECURITY_INFORMATION .....	45
2.3.2.28	PROPID_QM_ENCRYPT_PKS .....	45
2.3.2.29	PROPID_QM_SIGN_PKS .....	45
2.3.2.30	PROPID_QM_OWNER_SID .....	45

2.3.2.31	PROPID_QM_GROUP_IN_CLUSTER.....	45
2.3.2.32	PROPID_QM_SECURITY.....	46
2.3.2.33	PROPID_QM_SIGN_PK .....	46
2.3.2.34	PROPID_QM_ENCRYPT_PK.....	46
2.3.2.35	PROPID_QM_UPGRADE_DACL .....	46
2.3.3	Site Property Identifiers .....	46
2.3.3.1	PROPID_S_PATHNAME .....	46
2.3.3.2	PROPID_S_SITEID.....	46
2.3.3.3	PROPID_S_GATES .....	47
2.3.3.4	PROPID_S_PSC.....	47
2.3.3.5	PROPID_S_INTERVAL1 .....	47
2.3.3.6	PROPID_S_INTERVAL2.....	47
2.3.3.7	PROPID_S_FULL_NAME .....	47
2.3.3.8	PROPID_S_NT4_STUB.....	47
2.3.3.9	PROPID_S_FOREIGN.....	48
2.3.3.10	PROPID_S_SECURITY .....	48
2.3.3.11	PROPID_S_PSC_SIGNPK .....	48
2.3.3.12	PROPID_S_SECURITY_INFORMATION.....	48
2.3.4	Connected Network Property Identifiers .....	49
2.3.4.1	PROPID_CN_SECURITY .....	49
2.3.5	Enterprise Object Property Identifiers .....	49
2.3.5.1	PROPID_E_NAMESTYLE .....	49
2.3.5.2	PROPID_E_CSP_NAME .....	49
2.3.5.3	PROPID_E_ID .....	49
2.3.5.4	PROPID_E_LONG_LIVE.....	49
2.3.5.5	PROPID_E_VERSION.....	50
2.3.5.6	PROPID_E_SECURITY .....	50
2.3.6	User Object Property Identifiers .....	50
2.3.6.1	PROPID_U_SID .....	50
2.3.6.2	PROPID_U_SIGN_CERT .....	50
2.3.6.3	PROPID_U_DIGEST .....	50
2.3.6.4	PROPID_U_ID .....	50
2.3.7	Sitelink Property Identifiers .....	51
2.3.7.1	PROPID_L_NEIGHBOR1.....	51
2.3.7.2	PROPID_L_NEIGHBOR2.....	51
2.3.7.3	PROPID_L_COST .....	51
2.3.7.4	PROPID_L_ID .....	51
2.3.7.5	PROPID_L_GATES_DN .....	51
2.3.7.6	PROPID_L_NEIGHBOR1_DN .....	51
2.3.7.7	PROPID_L_NEIGHBOR2_DN .....	52
2.3.7.8	PROPID_L_DESCRIPTION .....	52
2.3.7.9	PROPID_L_FULL_PATH .....	52
2.3.7.10	PROPID_L_ACTUAL_COST .....	52
2.3.7.11	PROPID_L_GATES .....	52
2.3.8	Settings Property Identifiers .....	53
2.3.8.1	PROPID_SET_NAME.....	53
2.3.8.2	PROPID_SET_SERVICE.....	53
2.3.8.3	PROPID_SET_QM_ID .....	53
2.3.8.4	PROPID_SET_FULL_PATH .....	53
2.3.8.5	PROPID_SET_NT4 .....	54
2.3.8.6	PROPID_SET_MASTERID .....	54
2.3.8.7	PROPID_SET_SITENAME .....	54
2.3.8.8	PROPID_SET_SERVICE_ROUTING.....	54
2.3.8.9	PROPID_SET_SERVICE_DSSERVER .....	54
2.3.8.10	PROPID_SET_SERVICE_DEPCLIENTS.....	55

2.3.8.11	PROPID_SET_OLDSERVICE .....	55
2.3.9	MQUser Property Identifiers .....	55
2.3.9.1	PROPID_MQU_SID.....	55
2.3.9.2	PROPID_MQU_SIGN_CERT.....	56
2.3.9.3	PROPID_MQU_DIGEST .....	56
2.3.9.4	PROPID_MQU_ID .....	56
2.3.9.5	PROPID_MQU_SECURITY.....	56
2.3.10	Computer Property Identifiers.....	56
2.3.10.1	PROPID_COM_FULL_PATH .....	56
2.3.10.2	PROPID_COM_SAM_ACCOUNT .....	56
2.3.10.3	PROPID_COM_ACCOUNT_CONTROL .....	57
2.3.10.4	PROPID_COM_DNS_HOSTNAME .....	57
2.3.10.5	PROPID_COM_SID.....	57
2.3.10.6	PROPID_COM_SIGN_CERT.....	57
2.3.10.7	PROPID_COM_DIGEST .....	57
2.3.10.8	PROPID_COM_ID.....	58
2.3.11	Management Machine Property Identifiers .....	58
2.3.11.1	PROPID_MGMT_MSMQ_ACTIVEQUEUES.....	58
2.3.11.2	PROPID_MGMT_MSMQ_PRIVATEQ .....	58
2.3.11.3	PROPID_MGMT_MSMQ_DSSERVER .....	58
2.3.11.4	PROPID_MGMT_MSMQ_CONNECTED .....	58
2.3.11.5	PROPID_MGMT_MSMQ_TYPE.....	59
2.3.11.6	PROPID_MGMT_MSMQ_BYTES_IN_ALL_QUEUES.....	59
2.3.12	Management Queue Property Identifiers.....	59
2.3.12.1	PROPID_MGMT_QUEUE_PATHNAME .....	59
2.3.12.2	PROPID_MGMT_QUEUE_FORMATNAME .....	59
2.3.12.3	PROPID_MGMT_QUEUE_TYPE.....	59
2.3.12.4	PROPID_MGMT_QUEUE_LOCATION .....	60
2.3.12.5	PROPID_MGMT_QUEUE_XACT .....	60
2.3.12.6	PROPID_MGMT_QUEUE_FOREIGN.....	60
2.3.12.7	PROPID_MGMT_QUEUE_MESSAGE_COUNT .....	61
2.3.12.8	PROPID_MGMT_QUEUE_BYTES_IN_QUEUE .....	61
2.3.12.9	PROPID_MGMT_QUEUE_JOURNAL_MESSAGE_COUNT .....	61
2.3.12.10	PROPID_MGMT_QUEUE_BYTES_IN_JOURNAL .....	61
2.3.12.11	PROPID_MGMT_QUEUE_STATE.....	61
2.3.12.12	PROPID_MGMT_QUEUE_NEXTHOPS.....	62
2.3.12.13	PROPID_MGMT_QUEUE_EOD_LAST_ACK.....	62
2.3.12.14	PROPID_MGMT_QUEUE_EOD_LAST_ACK_TIME .....	63
2.3.12.15	PROPID_MGMT_QUEUE_EOD_LAST_ACK_COUNT .....	63
2.3.12.16	PROPID_MGMT_QUEUE_EOD_FIRST_NON_ACK .....	63
2.3.12.17	PROPID_MGMT_QUEUE_EOD_LAST_NON_ACK.....	63
2.3.12.18	PROPID_MGMT_QUEUE_EOD_NEXT_SEQ .....	63
2.3.12.19	PROPID_MGMT_QUEUE_EOD_NO_READ_COUNT .....	63
2.3.12.20	PROPID_MGMT_QUEUE_EOD_NO_ACK_COUNT.....	64
2.3.12.21	PROPID_MGMT_QUEUE_EOD_RESEND_TIME .....	64
2.3.12.22	PROPID_MGMT_QUEUE_EOD_RESEND_INTERVAL .....	64
2.3.12.23	PROPID_MGMT_QUEUE_EOD_RESEND_COUNT .....	64
2.3.12.24	PROPID_MGMT_QUEUE_EOD_SOURCE_INFO.....	64
2.3.12.25	PROPID_MGMT_QUEUE_CONNECTION_HISTORY.....	65
2.3.12.26	PROPID_MGMT_QUEUE_SUBQUEUE_COUNT.....	67
2.3.12.27	PROPID_MGMT_QUEUE_SUBQUEUE_NAMES.....	67
2.4	Active Directory Properties .....	67
2.4.1	msMQ-Recipient-FormatName .....	67
2.4.2	mSMQDigestsMig .....	67
2.4.3	mSMQMigrated.....	67

2.4.4	mSMQSignCertificatesMig .....	67
2.4.5	mSMQSiteGatesMig.....	68
2.5	Unused Active Directory Properties .....	68
2.6	PROPID to Active Directory Schema Mapping.....	68
2.6.1	Distribution Lists and Alias Classes .....	70
2.7	Error Codes .....	71
<b>3</b>	<b>Appendix A: PROPVARIANT IDL.....</b>	<b>81</b>
<b>4</b>	<b>Appendix B: Windows Behavior .....</b>	<b>84</b>
<b>5</b>	<b>Index.....</b>	<b>85</b>

# 1 Introduction

Message Queuing (MSMQ): Data Structures contains common definitions and data structures that are used in various protocols in the set of **Microsoft Message Queuing** protocols. The documentation for individual protocols contains references to this document, as needed.

## 1.1 Glossary

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

**Active Directory (AD)**  
**Globally Unique Identifier (GUID)**  
**Interface Definition Language (IDL)**  
**Microsoft Interface Definition Language (MIDL)**  
**Relative Identifier (RID)**  
**Remote Procedure Call (RPC)**  
**RPC Protocol Sequence**  
**RPC Transfer Syntax**  
**Security Identifier (SID)**  
**Universally Unique Identifier (UUID)**

The following terms are specific to this document:

**Active Queue:** A **queue** that contains **messages** or is currently opened by an application. Active queues may be **public queues**, **private queues**, or **outgoing queues**.

**Administration Queue:** A messaging **queue** that receives **Message Queuing (MSMQ)** system-generated acknowledgment **messages**. An administration queue is available to MSMQ applications for checking **message** status.

**Backup Site Controller (BSC):** An **MSMQ Directory Service** role played by an **MSMQ server**. A BSC contains a read-only copy of the directory for a **site**. A BSC may satisfy directory lookup requests, but cannot satisfy directory change requests. There may be zero or more BSCs in a **site**.

**Connected Network:** A network of computers in which any two computers can communicate directly through a common transport protocol (for example, TCP/IP or SPX/IPX). A computer can belong to multiple connected networks.

**ConnectedNetworkID:** A **GUID** that has been assigned to a particular **MSMQ Connected Network** and that is unique to that **Connected Network**.

**Connector Application:** An application that runs on an **MSMQ connector server** and translates both outgoing and incoming **messages** sent between a **Message Queuing** computer and a foreign messaging system.

**Connector Queue:** A **queue** used by a **connector server**. **Messages** sent to **foreign queues** are temporarily stored in a **connector queue** before they are retrieved by the **connector application**.

**Connector Server:** A **Message Queuing routing server** that is configured to send **messages** between a **Message Queuing site** and one or more foreign sites. A **connector server** has a **connector application** running on it and two **connector queues** for each foreign site: one used for **transactional messages** and one used for **non-transactional messages**.



**Cursor:** A data structure providing sequential access over a **message queue**. A cursor has a current pointer that lies between the head and tail pointer of the **queue**. The pointer can be moved forward or backward through an operation on the cursor (Next). A **message** at the current pointer can be accessed through a non-destructive read (Peek) or a destructive read (Receive) operation.

**Dead-Letter Queue:** A **queue** that contains **messages** that were sent from a host with a request for negative source journaling and that could not be delivered. **Message Queuing** provides a transactional dead-letter queue and a non-transactional dead-letter queue.

**Dependent Client:** An **MSMQ client** that does not have a **local queue** manager service. A dependent client has an **MSMQ client** run-time library but must rely on an **MSMQ** node operating in the role of supporting server to access **MSMQ** message queuing functionality.

**Direct Format Name:** A name that is used to reference a **public queue** or a **private queue** without accessing the **MSMQ Directory Service**. Message Queuing can use the physical, explicit location information provided by direct format names to send **messages** directly to their destinations. For more information, see section [2.1](#).

**Distribution List:** An **Active Directory** object that can contain explicit references only to destinations published in **Active Directory**; that is, to **public queues**, **queue aliases**, and other distribution lists, but not to private and URL-named queues.

**Enterprise:** A unit of administration of a network of **MSMQ clients** and **MSMQ servers**. An enterprise consists of an **MSMQ Directory Service**, one or more **connected networks** and one or more **MSMQ sites**.

**EnterpriseID:** A **GUID** that has been assigned to a particular **MSMQ enterprise** and is unique to that **enterprise**.

**Enterprise Site:** An **MSMQ site** that has a **Primary Enterprise Controller** as its **Primary Site Controller**.

**External Transaction:** An atomic transaction context dispensed by a transaction coordinator other than an **MSMQ queue manager**, such as by a distributed transaction coordinator (DTC), and used by an **MSMQ queue manager** to coordinate its state changes with state changes in other resource managers. For more information, see [\[MS-DTCO\]](#).

**Foreign Queue:** A messaging queue that resides on a computer that does not run an **MSMQ** messaging application.

**Format Name:** A name used to reference a **queue** when making calls to API functions.

**Independent Client:** A **Message Queuing** computer that runs the **Message Queuing** service, hosts **queues**, sends and receives **messages**, and can operate by default while disconnected from the network. These clients do not require synchronous access to a **Message Queuing** server to send and receive **messages**.

**Internal Transaction:** An atomic transaction context dispensed by an **MSMQ Queue Manager** instance that can be used to atomically commit or rollback state changes within that **MSMQ Queue Manager**. The dispensing **MSMQ Queue Manager** instance is the transaction coordinator and also the only resource manager participant supported by the transaction context. An internal transaction cannot, therefore, be used to coordinate state changes with other resource managers, including other **MSMQ Queue Manager** instances.

**Local Queue:** For **independent clients**, a **queue** located on the same server as the client. For **dependent clients**, a local queue is a **queue** located at the supporting server.

**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.

**Message Body:** A distinguished **message property** that represents the application payload.

**Message Header:** See **Message Packet Header**.

**Message Packet:** A byte buffer that is the physical representation of the **message** in the **queue manager** and on the wire.

**Message Packet Header:** The set of **message properties** in a **message packet** that precede the **message body** property. Also called a **message header**.

**Message Packet Trailer:** The set of **message properties** in a **message packet** that follow the **message body** property. Also called a **message trailer**.

**Message Property:** A data structure that contains a **property identifier** and a value, and that is associated with a **message**.

**Message 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. **Messages** are appended to the tail through a write operation (Send) that appends the **message** and increments the tail pointer. **Messages** are 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).

**Message Queuing Information Store (MQIS):** The directory service store used by **MSMQ Directory Service**.

**Message Trailer:** See **Message Packet Trailer**.

**Microsoft Message Queuing (MSMQ):** A communications service that provides asynchronous and reliable **message** passing between distributed client applications. In **Message Queuing**, clients send **messages** to **queues** and consume **messages** from **queues**. The **queues** provide persistence of the **messages**, enabling the sending and receiving client applications to operate asynchronously from one another.

**MSMQ:** See **Microsoft Message Queuing (MSMQ)**.

**MSMQ 1.0 Digital Signature:** A digital signature based on a hash of the following **message properties**: Correlation identifier, application specific identifier, **message body**, **message** label, response **queue**, **administration queue**. This signature type is supported by all versions of **Message Queuing**.

**MSMQ 2.0 Digital Signature:** A digital signature that is more robust than the **MSMQ 1.0 digital signature** and is based on a hash of the following **message properties**: Correlation identifier, application specific identifier, **message body**, **message** label, response **queue**, **administration queue**, Destination **queue property**, Source computer identifier, Delivery mode property, Message priority property, Message journaling property, Acknowledgment level property, Message class property, Body type property, and Connector type property. This signature type is not supported by **MSMQ** version 1.

**MSMQ 3.0 Digital Signature:** A digital signature that is used only for **messages** sent to distribution lists or [multiple-element format names](#) and is based on a hash of the following **message properties**: Correlation identifier, application specific identifier, **message body**, **message** label response **queue**, **administration queue**, Extension information property,

Destination **queue**, **format name** property, Source computer identifier, Delivery mode property, Message priority property, Message journaling property, Acknowledgment level property, Message class property, Body type property, and Connector type property. This signature type is not supported by **MSMQ** version 1 or **MSMQ** version 2.

**MSMQ Client:** An **MSMQ** client is an **MSMQ** node that does not act in any of the **MSMQ server** roles as defined by **MSMQ server**. There are two types of **MSMQ** clients: **dependent clients** and **independent clients**.

**MSMQ Connector Server:** See **Connector Server**.

**MSMQ Directory Service:** A network directory service that provides directory information, including key distribution, to **MSMQ**. It initially shipped in the Windows NT 4.0 Option Pack as part of **MSMQ**. This directory service predates, and is superseded by **Active Directory (AD)**.

**MSMQ Directory Service Server:** An **MSMQ server** that provides **MSMQ Directory Service**. The server can act in either of the **MSMQ Directory Service** roles: **Primary Site Controller (PSC)** or **Backup Site Controller (BSC)**.

**MSMQ Mixed-Mode:** When upgrading from **MSMQ** 1.0 in Windows NT 4.0 to **MSMQ** 2.0 in Windows 2000, a transitional mode known as mixed-mode environment is supported. Although not intended as a final deployment strategy, there is full support for this mixed-mode, which allows **MSMQ** 1.0 controller servers to coexist in the same **enterprise** with **MSMQ** 2.0 servers, supporting both **MSMQ** 1.0 and **MSMQ** 2.0 clients. In mixed-mode, the **MSMQ** replication service is used to synchronize **MQIS** with **Active Directory (AD)**.

**MSMQ Object:** Any one of the objects stored by **MSMQ** in its directory service. An object has a class name and a set of properties.

**MSMQ Queue Manager:** An **MSMQ** service hosted on a client or server machine that provides **queued** messaging services. **Queue managers** manage **queues** deployed on the local computer, and provide asynchronous transfer of **messages** to **queues** located on other computers. A **queue manager** is identified by a **globally unique identifier (GUID)**.

**MSMQ Routing Server:** A role played by an **MSMQ server**. An **MSMQ** routing server implements store and forward messaging. A routing server may provide connectivity between different **connected networks** within a **site**, or may provide session concentration between **sites**.

**MSMQ Server:** A server that acts in one or more of the **MSMQ Directory Service** roles: **Primary Enterprise Controller (PEC)**, **Primary Site Controller (PSC)**, or **Backup Site Controller (BSC)**; or in the role of **MSMQ routing server** or **MSMQ connector server**.

**MSMQ Site:** A network of computers, typically physically collocated, that have high connectivity as measured in terms of latency (low) and throughput (high). A site is represented by a site object in the directory service. An **MSMQ** site maps one-to-one with an **Active Directory** site when **Active Directory** provides directory services to **MSMQ**.

**MSMQ Site Gate:** An **MSMQ routing server** through which all intersite messaging traffic flows.

**MSMQ Site Link:** A communication link between two sites. A site link is represented by a site link object in the directory service. Site links may have associated link costs. Site links with their associated costs can be used to compute lowest-cost routing paths for store-and-forward messaging.

**MSMQ Supporting Server:** An **MSMQ** node that makes its local **MSMQ queue manager** service available to **MSMQ dependent clients**.

**Nontransactional Message:** A **message** that is sent outside of a transaction.

**Order Queue:** A messaging queue that is used to monitor the arrival order of **messages** that are sent as part of a transaction.

**Outgoing Queue:** A temporary internal **queue** that holds **messages** for a remote destination **queue**. The **path name** and **format name** of an outgoing **queue** are identical to the **path name** and **format name** of the corresponding destination **queue**. An outgoing queue is distinguished from its corresponding destination **queue** by the fact that the outgoing queue is located on the sending computer.

**Path Name:** The path name of a **queue** includes the name of the computer where the **messages** for the **queue** are stored, and an optional PRIVATE\$ key word indicating if the **queue** is private, followed by the name of the **queue**. Path names can also refer to subqueues; for more information, see section [2.1](#).

**Primary Enterprise Controller (PEC):** An **MSMQ Directory Service** role played by an **MSMQ server**. The PEC acts as the authority for the **enterprise** configuration information stored in the directory. There is only one PEC in an **enterprise**. The PEC also acts in the role of **Primary Site Controller (PSC)** for the **site** to which it belongs.

**Primary Site Controller (PSC):** An **MSMQ Directory Service** role played by an **MSMQ server**. The PSC acts as the authority for the directory information for the **site** to which it belongs. The PSC may satisfy directory lookup requests and directory change requests. There is only one PSC per **site**.

**Private Queue:** An application-defined **message queue** that is not registered in the **MSMQ Directory Service**. A private queue is deployed on a particular **queue manager**.

**Property Identifier:** A DWORD value associated with an **MSMQ object** property that defines the property type and its semantic meaning.

**Public Queue:** An application-defined **message queue** that is registered in the **MSMQ Directory Service**. A public queue may be deployed at any **queue manager**.

**Queue:** An object that holds **messages** passed between applications, or **messages** passed between **Message Queuing** and applications. In general, applications can send **messages** to queues and read **messages** from queues.

**Queue Alias:** An **Active Directory** object used to reference queues that might not be listed in **Active Directory**. Queue aliases are published in **Active Directory**.

**Queue Journal:** A **queue** that contains copies of the **messages** sent from a host when positive source journaling is requested.

**Queue Manager:** A message queuing service that manages **queues** deployed on a computer. A queue manager may also provide asynchronous transfer of **messages** to **queues** deployed on other queue managers.

**Queue Property:** A data structure that contains a **property identifier** and a value, and is associated with a **message queue**.

**Remote Queue:** For **independent clients**, a **queue** located at a remote server. For **dependent clients**, a **queue** located at a server other than the supporting server.

**Remote Read:** The act of reading (receiving) **messages** from a **remote queue**.

**SiteID:** A **GUID** that has been assigned to a particular **MSMQ site** and is unique to that **site**.

**SOAP Reliable Messaging Protocol (SRMP):** A published specification defining an open general-purpose extension of WS-Routing that adds reliability to the Web Services Routing Protocol (WS-Routing) and to SOAP. **Message Queuing** uses this protocol to format the packets in **messages** sent over HTTP/HTTPS or to a multicast address.

**Subqueue:** A **message queue** that is logically associated, through a naming hierarchy, with a parent **message queue**. Subqueues may be used to partition **messages** within the **queue**. For example, a **queue journal** may be a subqueue that holds a copy of each **message** consumed from its parent **queue**.

**System Queue:** An MSMQ-defined **message queue**.

**Transactional Message:** A **message** sent as part of a transaction. Transaction **messages** must be sent to **transactional queues**.

**Transactional Queue:** A **queue** that contains only **transactional messages**.

**Unit of Work:** A set of individual operations that **MSMQ** must successfully complete before any of the individual **MSMQ** operations can be considered complete.

**XML Digital Signature:** When authentication is requested for an HTTP **message** or a multicast **message**, which is also sent in **SRMP**, it is automatically signed using an XML digital signature. The value in the signature value is calculated by hashing by using the SHA1 hashing algorithm, encrypting the hash with the public key in the certificate attached to the **message**, and converting the result to base64 encoding. This signature type is not supported by **MSMQ** version 1 or **MSMQ** version 2.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described 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](mailto: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.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, <http://www.opengroup.org/public/pubs/catalog/c706.htm>

[MS-ADTS] Microsoft Corporation, "[Active Directory Technical Specification](#)", June 2007.

[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-MQDS] Microsoft Corporation, "[Message Queuing \(MSMQ\): Directory Service Protocol Specification](#)", July 2007.

[MS-MQMA] Microsoft Corporation, "[Message Queuing \(MSMQ\): Architecture Protocol Specification](#)", August 2007.

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

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

[MS-RDPBCGR] Microsoft Corporation, "[Remote Desktop Protocol: Basic Connectivity and Graphics Remoting Specification](#)", June 2007.

[MS-RPCE] Microsoft Corporation, "[Remote Procedure Call Protocol Extensions](#)", January 2007.

[MS-SAMR] Microsoft Corporation, "[Security Account Manager \(SAM\) Remote Protocol Specification \(Client-to-Server\)](#)", June 2007.

[RFC1035] Mockapetris, R., "Domain Names - Implementation and Specification", RFC 1035, November 1987, <http://www.ietf.org/rfc/rfc1035.txt>

[RFC1123] Braden, R., "Requirements for Internet Hosts–Application and Support", RFC 1123, October 1989, <http://www.ietf.org/rfc/rfc1123.txt>

[RFC1321] Rivest, R. "The MD5 Message-Digest Algorithm", RFC 1321, April 1992, <http://www.ietf.org/rfc/rfc1321.txt>

[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>

[RFC2251] Wahl, M., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997, <http://www.ietf.org/rfc/rfc2251.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>

[RFC4514] Network Working Group, Zeilenga, K., Ed, "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", RFC 4514, June 2006, <http://www.ietf.org/rfc/rfc4514.txt>

[RFC4516] Network Working Group, Smith, M., Ed, and Howes, T., "Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator", RFC 4516, June 2006, <http://www.ietf.org/rfc/rfc4516.txt>

## 1.2.2 Informative References

[IANAIMA] Internet Assigned Numbers Authority, "Internet Multicast Addresses", March 2007, <http://www.iana.org/assignments/multicast-addresses>

[LDAP] Microsoft Corporation, "About Lightweight Directory Access Protocol", <http://msdn2.microsoft.com/en-us/library/aa366075.aspx>

If you have any trouble finding [LDAP], please check [here](#).

[MIDLINF] Microsoft Corporation, "MIDL Language Reference", <http://msdn2.microsoft.com/en-us/library/aa367088.aspx>

[MSDN-MQEIC] Microsoft Corporation, "Message Queuing Error and Information Codes", <http://msdn2.microsoft.com/en-gb/library/ms700106.aspx>

[MS-DTCO] Microsoft Corporation, "[MSDTC Connection Manager: OleTx Transaction Protocol Specification](#)", July 2007.

### **1.3 Relationship to Protocols and Other Structures**

The data types in this document are used by protocols in the set of Microsoft Message Queuing protocols as specified in [\[MS-MQMA\]](#).

### **1.4 Applicability Statement**

An applicability statement is not applicable.

### **1.5 Versioning and Localization**

Neither versioning or localization information is applicable.

### **1.6 Vendor-Extensible Fields**

There are no vendor-extensible fields.



## 2 Definitions and Structures

This section discusses data structures that are used by various protocols in the set of Microsoft Message Queuing protocols.

### 2.1 MSMQ Queue Names

The following sections describe the various ways to designate Message Queuing **queues**.

#### 2.1.1 Path Names

The path of a **public queue** consists of the name of the computer hosting the queue and the name of the queue separated by a backward slash in the form "ComputerName\QueueName". The names of **private queues** are prefixed by the string "private\$" separated by a backward slash. Thus, the path of a private queues has the form "ComputerName\ private\$\QueueName".

A queue **path name** MUST conform to the following format in ABNF notation:

```
QueuePathName = (Computer "\" QueueName |
                  Computer "\"private$" QueueName)
Computer       = 1*256(VCHAR)
```

For MSMQ 1.0:

```
QueueName = 1*124(%x21-3A | %x3C-5B | %x5D-7f)
           ; Exclude backslash and semicolon
```

For MSMQ 2.0–MSMQ 3.0:

```
QueueName = 1*124(%x21 | %x23-2A | %x2D-3A | %x3C-5B | %x5D-7f)
           ; Exclude backslash, semicolon, plus, comma, double quote
```

For MSMQ 4.0:

```
QueueName = 1*124(%x21 | %x23-2A | %x2D-3A | %x3C-5B | %x5D-7f)
           [";" Subqueue]

Subqueue   = 1*32(%x21 | %x23-2A | %x2D-3A | %x3C-5B | %x5D-7f)
```

#### 2.1.2 Direct Format Names

Direct format names are used to reference public or private queues without accessing the directory service. Message Queuing can use the information provided by direct format names to send



**messages** directly to their destinations. Thus, direct format names can be used to send and receive messages in workgroup mode, send messages to computers on the Internet, and send messages directly to a computer.

A direct format name **MUST** conform to the following format in ABNF notation.

```
DirectName = PrivateQueuePath | PublicQueuePath | MachineQueuePath

PrivateQueuePath = "DIRECT=" Protocol ":"
                  ProtocolAddressSpecification "\"PRIVATE$"
                  QueueName [";JOURNAL"]

PublicQueuePath = "DIRECT=" Protocol ":"
                 ProtocolAddressSpecification "\" QueueName
                 [";JOURNAL"]

MachineQueuePath = "DIRECT=" Protocol ":"
                  ProtocolAddressSpecification "\"SYSTEM$;"
                  ("JOURNAL" | "DEADLETTER" | "DEADXACT")

Subqueue=1*32(%x21 | %x23-2A | %x2D-3A | %x3C-5B / %x5D-7f)

Protocol = "TCP" | "OS" | "HTTP" | "HTTPS" | "IPX"
```

Where:

- <QueueName> is a queue path name from [Path names \(section 2.1.1\)](#).
- <ProtocolAddressSpecification> is the protocol-specific address format as defined in the following table.

Protocol	Description	Protocol address specification
TCP	Connection-oriented TCP over IP	Internet address notation (IP address)
IPX	Connection-oriented SPX over IPX	Network number and host number (separated by the colon (":") character)
OS	Connection using the native computer-naming convention	Any computer name supported by the underlying operating system

### 2.1.3 Public Format Names

Public format names and [direct format names](#) are used to reference public queues. When a public format name is used, Message Queuing uses its internal routing algorithm to define the route to the destination queue.

Public format names contain the string "PUBLIC=" followed by the identifier assigned to the queue when it was created. This identifier is the **globally unique identifier (GUID)** listed for the queue object in Active Directory.

A public format name **MUST** conform to the following format in Augmented Backus-Naur Form (ABNF) notation:

```

PublicName = "PUBLIC=" QueueGuid

QueueGuid = Guid

Guid       = 8HexDig %x2D 3(4HexDig %x2D) 12HexDig

HexDig     = Digit | "A" | "B" | "C" | "D" | "E" | "F"

Digit      = %x30-39

```

#### 2.1.4 Private Format Names

Private format names are used to reference private queues. When a private format name is used, Message Queuing uses its internal routing algorithm to define the route to the destination queue.

When Message Queuing detects a private format name, it does not refer to the directory service for information about the queue. However, it does use the directory service to lookup information about the computer for routing purposes.

Private format names contain the string "PRIVATE=" followed by the identifier of the computer where the queue is registered and a hexadecimal number that identifies the queue.

A private format name **MUST** conform to the following format in ABNF notation.

```

PrivateName = "PRIVATE=" ComputerGuid "\" 1*8HEXDIG [";JOURNAL"]

ComputerGuid = Guid

```

where:

<Guid> is a **GUID**, as specified in section [2.1.3](#).

#### 2.1.5 Distribution List Format Names

Distribution list format names are used to reference **distribution lists** (group objects) stored in Active Directory (as specified in [\[MS-ADTS\]](#)). Distribution list format names contain the string "DL=" followed by the distribution list identifier. This identifier is the GUID listed for the distribution list (group) object in Active Directory. The following is the general format used to reference a distribution list with optional inclusion of the Active Directory domain name.

The name **MUST** conform to the following format in ABNF notation.

```

DistributionListName = "DL=" DistributionListGuid ["@" DomainName]

DistributionListGuid = Guid

DomainName           = 1*VCHAR

VCHAR                 = %x21-7E

```

Where:

<Guid> is a **GUID**, as specified in section [2.1.3](#).

### 2.1.6 Machine, Connector, and Multicast Format Names

Machine format names are used to reference computer journals and **dead-letter queues** for a specific computer (for MSMQ 2.0 and MSMQ 3.0, [direct format names](#) also can be used for this purpose). Connector format names are used to reference the **connector queues** on a **connector server**. Multicast address format names (introduced in MSMQ 3.0) reference multiple destination queues.

These names MUST conform to the following format in ABNF notation.

```
Machine    = "MACHINE=" Guid
Connector  = "CONNECTOR=" Guid
Multicast  = "MULTICAST=" Address
Address    = 1*3Digit 3("1*3Digit) ":" Port
Port       = 1*5Digit
```

Where:

<Guid> is a **GUID**, as specified in section [2.1.3](#).

### 2.1.7 Multiple-Element Format Names

A multiple-element format name is formed as a concatenation of one or more [public](#), [private](#), [direct](#), or [distribution-list format names](#) separated by commas. Thus, different kinds of **format names** used in Message Queuing can be used together as elements of a multiple-element format name.

The following example shows a multiple-element format name that contains a direct format name, a public format name, and a distribution-list format name.

```
DIRECT=ComputerAddress\QueueName,PUBLIC=QueueGUID,DL=DL_GUID
```

**Note** A multiple-element format name containing an element that points to a public queue or a distribution list cannot be used when there is no access to Active Directory.

Multiple-element format names cannot contain the format names of read-only queues, such as **queue journals**, computer journals, or dead-letter queues. An error is returned if the format name of a read-only queue is included in the multiple-element format name.

The name MUST conform to the following format in ABNF notation.

```

MultipleElementName = FormatName * ("," FormatName)

FormatName = DirectName | PublicName | PrivateName |
             DistributionListName | Machine | Connector |
             Multicast

```

## 2.2 Structures

### 2.2.1 MQDSPUBLICKEY

The MQDSPUBLICKEY structure defines a public key certificate.

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
ulKeyLen																															
ulProviderLen																															
ulProviderType																															
aBuf (variable)																															
...																															

**ulKeyLen (4 bytes):** An unsigned 32-bit integer that MUST contain the size in bytes of the MQDSPUBLICKEY structure.

**ulProviderLen (4 bytes):** An unsigned 32-bit integer that MUST contain the size in bytes of the provider name, including the terminating null.

**ulProviderType (4 bytes):** An unsigned 32-bit integer that MUST contain an enumerated constant for the provider-type code. The only value supported is PROV\_RSA\_FULL (0x00000001).

**aBuf (variable):** A buffer containing the provider name concatenated with the public key certificate. The provider name is a Unicode string, null terminated. The public key certificate is formatted as specified in [\[MS-RDPBCGR\]](#) section 2.2.1.4.3.1.1.1.

### 2.2.2 MQDSPUBLICKEYS

The MQDSPUBLICKEYS structure defines a set of [MQDSPUBLICKEY \(section 2.2.1\)](#) structures.

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
ulLen																															
cNumofKeys																															
aPublicKeys (variable)																															
...																															

**ulLen (4 bytes):** An unsigned 32-bit integer that MUST contain the size in bytes of the MQDSPUBLICKEYS structure.

**cNumofKeys (4 bytes):** An unsigned 32-bit integer that MUST contain the count of MQDSPUBLICKEY (section 2.2.1) structures in the array aPublicKeys.

**aPublicKeys (variable):** An array of MQDSPUBLICKEY (section 2.2.1) structures.

### 2.2.3 SECURITY\_INFORMATION

A SECURITY\_INFORMATION value applies to a [SECURITY\\_DESCRIPTOR](#) (as specified in [\[MS-DTYP\]](#) section 2.4.6). The value is constructed from zero or more bit flags from the following table.

Value	Meaning
OWNER_SECURITY_INFORMATION 0x00000001	Owner identifier of the object
GROUP_SECURITY_INFORMATION 0x00000002	Primary group identifier
DACL_SECURITY_INFORMATION 0x00000004	Discretionary access control list (ACL) of the object
SACL_SECURITY_INFORMATION 0x00000008	System ACL of the object

This type is declared as follows:

```
typedef DWORD SECURITY_INFORMATION;
```

### 2.2.4 TA\_ADDRESS

The **TA\_ADDRESS** structure defines a single transport address of a specific type.

```
typedef struct _TA_ADDRESS {
```

```

USHORT AddressLength;
USHORT AddressType;
UCHAR Address[1];
} TA_ADDRESS,
*PTA_ADDRESS;

```

**AddressLength:** An unsigned 16-bit integer that **MUST** contain the size in bytes of the **Address** field. The value **MUST** be one of the following (by address type):

Address Type Prefix	Value
IP_	4
IPX_	10
FOREIGN_	16

**AddressType:** An unsigned 16-bit integer that **MUST** contain one of the values in the following table.

Value	Meaning
IP_ADDRESS_TYPE 0x0001	The <b>Address</b> field contains a 32-bit IP address.
IP_RAS_ADDRESS_TYPE 0x0002	The <b>Address</b> field contains a 32-bit IP address.
IPX_ADDRESS_TYPE 0x0003	The <b>Address</b> field contains a 4-byte netnum followed by a 6-byte nodenum. The netnum identifies the IPX network. The nodenum represents the IPX node address.
FOREIGN_ADDRESS_TYPE 0x0005	The <b>Address</b> field contains the globally unique identifier (GUID) of a <b>connected network</b> object.

**Address:** The array of bytes that contains the address value.

This **MUST** be one of the following:

- An IP address (as specified in section [2.2.4.1](#)).
- An IPX address (as specified in section [2.2.4.2](#))
- A FOREIGN address is a **GUID** object (as specified in [\[MS-DTYP\]](#) section **2.3.4**).

### 2.2.4.1 IP Address

The IP Address packet is a numerical representation of an IPv4 address.

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
IP_Address																															

**IP\_Address (4 bytes):** A 32-bit unsigned integer.

#### 2.2.4.2 IPX Address

The IPX Address packet identifies a remote destination on a Novell Netware network.

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
Netnum																															
nodenum																															
...																															

#### 2.2.5 SEQUENCE\_INFO

The **SEQUENCE\_INFO** structure stores the sequence information about the applicable message in a message stream sent from a given sending computer to a given destination queue.

```
typedef struct tagSEQUENCE_INFO {
    LONGLONG SeqID;
    ULONG SeqNo;
    ULONG PrevNo;
} SEQUENCE_INFO;
```

**SeqID:** Specifies a sequence identifier. This value **MUST** be set to the sequence identifier of the message being acknowledged.

**SeqNo:** Specifies a sequence number of a message within a stream. This value **MUST** be set to the sequence number of the message being acknowledged.

**PrevNo:** Specifies a sequence number. This value **MUST** be set to the sequence number of the previous message received in the stream.

##### 2.2.5.1 SEQUENCE\_INFO (Packet)

The **SEQUENCE\_INFO (Packet)** stores the sequence information about the applicable message in a message stream sent from a given sending computer to a given destination queue.

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
SeqID																															
...																															
SeqNo																															
PrevNo																															

**SeqID (8 bytes):** A 64-bit signed integer that specifies a sequence identifier. MUST be set to the sequence identifier of the message being acknowledged.

**SeqNo (4 bytes):** A 32-bit unsigned long integer that specifies a sequence number of a message within a stream. MUST be set to the sequence number of the message being acknowledged.

**PrevNo (4 bytes):** A 32-bit unsigned long integer that specifies a sequence number. MUST be set to the sequence number of the previous message received in the stream.

## 2.2.6 QUEUE\_FORMAT\_TYPE

The **QUEUE\_FORMAT\_TYPE** enumeration identifies the type of name format being used.

```
typedef enum __QUEUE_FORMAT_TYPE
{
    QUEUE_FORMAT_TYPE_UNKNOWN = 0,
    QUEUE_FORMAT_TYPE_PUBLIC = 1,
    QUEUE_FORMAT_TYPE_PRIVATE = 2,
    QUEUE_FORMAT_TYPE_DIRECT = 3,
    QUEUE_FORMAT_TYPE_MACHINE = 4,
    QUEUE_FORMAT_TYPE_CONNECTOR = 5,
    QUEUE_FORMAT_TYPE_DL = 6,
    QUEUE_FORMAT_TYPE_MULTICAST = 7,
    QUEUE_FORMAT_TYPE_SUBQUEUE = 8
} QUEUE_FORMAT_TYPE;
```

**QUEUE\_FORMAT\_TYPE\_UNKNOWN:** The format type is unknown.

**QUEUE\_FORMAT\_TYPE\_PUBLIC:** The [QUEUE\\_FORMAT \(section 2.2.7\)](#) structure contains a **GUID** (as specified in [\[MS-DTYP\]](#) section 2.3.4) that identifies a queue.

**QUEUE\_FORMAT\_TYPE\_PRIVATE:** The **QUEUE\_FORMAT** (section 2.2.7) structure contains an [OBJECTID](#) that identifies a queue.

**QUEUE\_FORMAT\_TYPE\_DIRECT:** The **QUEUE\_FORMAT** structure contains a direct name string that identifies a queue.

**QUEUE\_FORMAT\_TYPE\_MACHINE:** The **QUEUE\_FORMAT** structure contains a **GUID** (as specified in [\[MS-DTYP\]](#) section 2.3.4) that identifies a queue.



**QUEUE\_FORMAT\_TYPE\_CONNECTOR:** The **QUEUE\_FORMAT** structure contains a **GUID** (as specified in [MS-DTYP] section **2.3.4**) that identifies a connector queue. This is not supported by all protocols.

**QUEUE\_FORMAT\_TYPE\_DL:** The **QUEUE\_FORMAT** structure contains a **GUID** (as specified in [MS-DTYP] section **2.3.4**) that identifies a distribution list (DL). This is not supported by all protocols.

**QUEUE\_FORMAT\_TYPE\_MULTICAST:** The **QUEUE\_FORMAT** structure contains a **GUID** (as specified in [MS-DTYP] section **2.3.4**) that identifies a multicast address. This is not supported by all protocols.

**QUEUE\_FORMAT\_TYPE\_SUBQUEUE:** The **QUEUE\_FORMAT** structure contains a direct name string that identifies a **subqueue**.

**Note** **QUEUE\_FORMAT\_TYPE\_SUBQUEUE** was introduced in MSMQ version 4.

## 2.2.7 QUEUE\_FORMAT

The **QUEUE\_FORMAT** structure describes the type of queue being managed and an identifier for that queue.

```
typedef struct _QUEUE_FORMAT {
    unsigned char m_qft;
    unsigned char m_SuffixAndFlags;
    unsigned short m_reserved;
    [switch_is(m_qft)] union {
        [case(QUEUE_FORMAT_TYPE_PUBLIC)]
            GUID m_gPublicID;
        [case(QUEUE_FORMAT_TYPE_PRIVATE)]
            OBJECTID m_oPrivateID;
        [case(QUEUE_FORMAT_TYPE_DIRECT)]
            LPWSTR m_pDirectID;
        [case(QUEUE_FORMAT_TYPE_MACHINE)]
            GUID m_gMachineID;
        [case(QUEUE_FORMAT_TYPE_CONNECTOR)]
            GUID m_GConnectorID;
        [case(QUEUE_FORMAT_TYPE_DL)]
            DL_ID m_DlID;
        [case(QUEUE_FORMAT_TYPE_MULTICAST)]
            MULTICAST_ID m_MulticastID;
        [case(QUEUE_FORMAT_TYPE_SUBQUEUE)]
            LPWSTR m_pDirectSubqueueID;
    } m_qft;
} QUEUE_FORMAT;
```

**m\_qft:** The type of queue format name.

**m\_SuffixAndFlags:** This member is broken into two subfields: **Suffix Type** is located in the four least-significant bits, and **Flags** is located in the four most-significant bits.

0	1	2	3	4	5	6	7
Flags				Suffix type			

Flags	Meaning
QUEUE_FORMAT_FLAG_NOT_SYSTEM 0x00	The specified queue is not a <b>system queue</b> .
QUEUE_FORMAT_FLAG_SYSTEM 0x80	The specified queue is a system queue.

Suffix Type	Meaning
QUEUE_SUFFIX_TYPE_NONE 0x00	No suffix is specified. The <b>Flags</b> subfield MUST be set to 0x00. The <b>m_qft</b> member MUST NOT be set to 0x04.
QUEUE_SUFFIX_TYPE_JOURNAL 0x01	A journal suffix. The <b>Flags</b> subfield MUST be set to 0x80. The <b>m_qft</b> member MUST NOT be set to 0x05, 0x06, or 0x07.
QUEUE_SUFFIX_TYPE_DEADLETTER 0x02	A dead-letter suffix. The <b>Flags</b> subfield MUST be set to 0x80. The <b>m_qft</b> member MUST NOT be set to 0x01, 0x02, 0x05, 0x06, or 0x07.
QUEUE_SUFFIX_TYPE_DEADXACT 0x03	A transacted dead-letter suffix. The <b>Flags</b> subfield MUST be set to 0x80. The <b>m_qft</b> member MUST be set to 0x03 or 0x04.
QUEUE_SUFFIX_TYPE_XACTONLY 0x04	A transaction-only suffix. The <b>m_qft</b> member MUST be set to 0x05.
QUEUE_SUFFIX_TYPE_SUBQUEUE 0x05	A subqueue suffix. The <b>Flags</b> subfield MUST be 0x00. The <b>m_qft</b> member MUST be set to 0x08.

**m\_reserved:** The integer value used for padding. The client SHOULD set this value to 0. The server MUST not use it.

**\_m\_qft:** Based on the value of **m\_qft**.

**m\_gPublicID:** A **GUID** (as specified in [\[MS-DTYP\]](#) section 2.3.4) of a public queue. Selected when **m\_qft** is set to 0x01.

**m\_oPrivateID:** An **OBJECTID** of a private queue; members MUST be used as specified in **OBJECTID**. Selected when **m\_qft** is set to 0x02.

**m\_pDirectID:** Name identifier of a direct queue (as specified in section 2.1 for the ABNF). It is selected when **m\_qft** is set to 0x03.

**m\_gMachineID:** The **GUID** (as specified in [\[MS-DTYP\]](#) section 2.3.4) of a machine. It is selected when **m\_qft** is set to 0x04.

**m\_GConnectorID:** The **GUID** (as specified in [\[MS-DTYP\]](#) section 2.3.4) of a connector queue. It is selected when **m\_qft** is set to 0x05.

**m\_DIID:** The identifier of a distribution list. It is selected when **m\_qft** is set to 0x06.

**m\_MulticastID:** The identifier of a multicast queue. It is selected when **m\_qft** is set to 0x07.

**m\_pDirectSubqueueID:** The identifier of a subqueue. Selected when **m\_qft** is set to 0x08.

The value MUST conform to the ABNF for DirectName and contain the optional <Subqueue> element, as specified in [2.1](#).

The full QUEUE\_FORMAT IDL is specified in [\[MS-MQMR\] Appendix A](#).

### 2.2.8 OBJECTID

The **OBJECTID** structure uniquely distinguishes a repository object from all other repository objects represented in a repository database. Each repository object instance has two identifiers: an object identifier and an internal identifier.

```
typedef struct _OBJECTID {
    GUID Lineage;
    DWORD Uniquifier;
} OBJECTID;
```

**Lineage:** The identifying object globally in the scope. MUST be a unique **GUID** (as specified in [\[MS-DTYP\]](#) section **2.3.4**) value when the original object is created. Across objects in a repository database, no two objects can have the same **Lineage**.

**Uniquifier:** The numeric object identifier that is unique within the scope of the server. A single global object can have multiple **Uniquifier** modifiers.

### 2.2.9 DL\_ID

The **DL\_ID** structure defines a distribution list queue identifier.

```
typedef struct _DL_ID {
    GUID m_DlGuid;
    LPWSTR m_pwzDomain;
} DL_ID;
```

**m\_DlGuid:** The **GUID** (as specified in [\[MS-DTYP\]](#) section **2.3.4**) of the distribution list queue.

**m\_pwzDomain:** The Active Directory domain of the distribution list queue. MUST be a null-terminated Unicode string.

### 2.2.10 MULTICAST\_ID

The **MULTICAST\_ID** structure defines a multicast queue identifier.

```
typedef struct _MULTICAST_ID {
    ULONG m_address;
    ULONG m_port;
} MULTICAST_ID;
```

**m\_address:** The IP address of the queue.

**m\_port:** The port to which the queue is attached.

### 2.2.11 QUEUE\_SUFFIX\_TYPE

```
typedef enum
{
    QUEUE_SUFFIX_TYPE_NONE = 0,
    QUEUE_SUFFIX_TYPE_JOURNAL = 1,
    QUEUE_SUFFIX_TYPE_DEADLETTER = 2,
    QUEUE_SUFFIX_TYPE_DEADXACT = 3,
    QUEUE_SUFFIX_TYPE_XACTONLY = 4,
    QUEUE_SUFFIX_TYPE_SUBQUEUE = 5
} QUEUE_SUFFIX_TYPE;
```

**QUEUE\_SUFFIX\_TYPE\_NONE:** There is no suffix.

**QUEUE\_SUFFIX\_TYPE\_JOURNAL:** Refers to the queue journal of the queue identified by the unnamed union in the [QUEUE\\_FORMAT \(section 2.2.7\)](#) structure.

**QUEUE\_SUFFIX\_TYPE\_DEADLETTER:** Refers to the non-transacted dead-letter queue of the computer identified by the union in the **QUEUE\_FORMAT** (section 2.2.7) structure.

**QUEUE\_SUFFIX\_TYPE\_DEADXACT:** Refers to the transacted dead-letter queue of the computer identified by the union in the **QUEUE\_FORMAT** (section 2.2.7) structure.

**QUEUE\_SUFFIX\_TYPE\_XACTONLY:** Refers to the transacted connector queue of the connector identified by the union in the **QUEUE\_FORMAT** (section 2.2.7) structure.

**QUEUE\_SUFFIX\_TYPE\_SUBQUEUE:** Refers to the subqueue that is the direct name identified by the union in the **QUEUE\_FORMAT** (section 2.2.7) structure.

### 2.2.12 PROPVARIANT Type Constants

The following values are used in the discriminant field, **vt**, of the [PROPVARIANT \(section 2.2.13\)](#) type.

The PROPVARIANT (section 2.2.13) type constants are defined in the **VARENUM** enumeration, as follows:

```
typedef enum
{
    VT_EMPTY = 0,
    VT_NULL = 1,
    VT_I2 = 2,
    VT_I4 = 3,
    VT_BOOL = 11,
    VT_VARIANT = 12,
    VT_I1 = 16,
    VT_UI1 = 17,
    VT_UI2 = 18,
```

```

VT_UI4 = 19,
VT_I8 = 20,
VT_UI8 = 21,
VT_LPWSTR = 31,
VT_BLOB = 65,
VT_CLSID = 72,
VT_VECTOR = 0x1000
} VARENUM;

```

**VT\_EMPTY:** (0x0000): The type of the contained field is undefined. When this flag is specified, the PROPVARIANT (section 2.2.13) MUST not contain a data field.

**VT\_NULL:** (0x0001): The type of the contained field is NULL. When this flag is specified, the PROPVARIANT (section 2.2.13) MUST not contain a data field.

**VT\_I2:** (0x0002): The type of the contained field MUST be a 2-byte signed integer.

**VT\_I4:** (0x0003): The type of the contained field MUST be a 4-byte signed integer.

**VT\_BOOL:** (0x000B): The type of the contained field MUST be [VARIANT\\_BOOL \(section 2.2.14\)](#).

**VT\_VARIANT:** (0x000C): The type of the contained field MUST be [CAPROPVARIANT \(section 2.2.16.8\)](#). It MUST appear with the bit flag VT\_VECTOR.

**VT\_I1:** (0x0010): The type of the contained field MUST be a 1-byte integer.

**VT\_UI1:** (0x0011): The type of the contained field MUST be a 1-byte unsigned integer.

**VT\_UI2:** (0x0012): The type of the contained field MUST be a 2-byte unsigned integer.

**VT\_UI4:** (0x0013): The type of the contained field MUST be a 4-byte unsigned integer.

**VT\_I8:** (0x0014): The type of the contained field MUST be an 8-byte signed integer.

**VT\_UI8:** (0x0015): The type of the contained field MUST be an 8-byte unsigned integer.

**VT\_LPWSTR:** (0x001F): The type of the contained field MUST be an **LPWSTR** (as specified in [\[MS-DTYP\]](#) section **2.2.34**), a null-terminated Unicode string.

**VT\_BLOB:** (0x0041): The type of the contained field MUST be a [BLOB \(section 2.2.15\)](#).

**VT\_CLSID:** (0x0048): The type of the contained field MUST be a pointer to a **GUID** (as specified in [\[MS-DTYP\]](#) section **2.3.4**) value.

**VT\_VECTOR:** (0x1000): The type of the contained field MUST be combined with other values by using the bitwise OR operation to indicate a counted field. The type of the contained field MUST be a [COUNTEDARRAY \(section 2.2.16\)](#).

### 2.2.12.1 VARTYPE

The **VARTYPE** holds [VARENUM \(section 2.2.12\)](#) enumerated values.

This type is declared as follows:

```
typedef unsigned short VARTYPE;
```

### 2.2.13 PROPVARIANT

The [PROPVARIANT \(section 2.2.13.2\)](#) is a container for a union that can hold many types of data.

#### 2.2.13.1 tag\_inner\_PROPVARIANT

```
typedef struct _tag_inner_PROPVARIANT {
    VARTYPE vt;
    UCHAR wReserved1;
    UCHAR wReserved2;
    ULONG wReserved3;
    [switch is(vt)] union {
        [case(VT_UI1)]
            UCHAR bVal;
        [case(VT_I2)]
            SHORT iVal;
        [case(VT_UI2)]
            USHORT uiVal;
        [case(VT_I4)]
            LONG lVal;
        [case(VT_UI4)]
            ULONG ulVal;
        [case(VT_BOOL)]
            VARIANT_BOOL boolVal;
        [case(VT_CLSID)]
            GUID* puuid;
        [case(VT_BLOB)]
            BLOB blob;
        [case(VT_LPWSTR)]
            LPWSTR pwszVal;
        [case(VT_VECTOR|VT_UI1)]
            CAUB caub;
        [case(VT_VECTOR|VT_UI2)]
            CAUI caui;
        [case(VT_VECTOR|VT_I4)]
            CAL cal;
        [case(VT_VECTOR|VT_UI4)]
            CAUL caul;
        [case(VT_VECTOR|VT_UI8)]
            CAUH cauh;
        [case(VT_VECTOR|VT_CLSID)]
            CACLSID cauuid;
        [case(VT_VECTOR|VT_LPWSTR)]
            CALPWSTR calpwstr;
        [case(VT_VECTOR|VT_VARIANT)]
            CAPROPVARIANT capropvar;
    } _varUnion;
} tag_inner_PROPVARIANT;
```

**vt:** MUST be set to one of the values as specified in section [2.2.12](#).

**wReserved1:** MAY be set to 0 and MUST be ignored by the recipient.

**wReserved2:** MAY be set to 0 and MUST be ignored by the recipient.

**wReserved3:** MAY be set to 0 and MUST be ignored by the recipient.

**\_varUnion:** MUST contain an instance of the type according to the value in the **vt** field.

### 2.2.13.2 PROPVARIANT

The following is the type definition for **PROPVARIANT**.

This type is declared as follows:

```
typedef tag_inner_PROPVARIANT PROPVARIANT;
```

### 2.2.14 VARIANT\_BOOL

The **VARIANT\_BOOL** type specifies Boolean values.

The values MUST be defined as:

Name/Value	Value	Description
VARIANT_TRUE	0xFFFF	MUST indicate a Boolean value of TRUE.
VARIANT_FALSE	0x0	MUST indicate a Boolean value of FALSE.

This type is declared as follows:

```
typedef short VARIANT_BOOL;
```

### 2.2.15 BLOB

The **BLOB** structure defines a counted array of unsigned characters.

```
typedef struct tagBLOB {  
    unsigned long cbSize;  
    [size_is(cbSize)] unsigned char* pBlobData;  
} BLOB;
```

**cbSize:** A 32-bit unsigned long that specifies the size of the array of unsigned characters pointed to by **pBlobData**.

**pBlobData:** An array of 8-bit unsigned characters.

## 2.2.16 COUNTEDARRAY

A COUNTEDARRAY specifies a counted array of types.

### 2.2.16.1 CAUB

The **CAUB** structure defines a counted array of unsigned characters.

```
typedef struct tagCAUB {
    unsigned long cElems;
    [size_is(cElems)] unsigned char* pElems;
} CAUB;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of unsigned characters.

### 2.2.16.2 CAUI

The **CAUI** structure defines a counted array of unsigned short integers.

```
typedef struct tagCAUI {
    unsigned long cElems;
    [size_is(cElems)] unsigned short* pElems;
} CAUI;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of unsigned short integers.

### 2.2.16.3 CAL

The **CAL** structure defines a counted array of long integers.

```
typedef struct tagCAL {
    unsigned long cElems;
    [size_is(cElems)] long* pElems;
} CAL;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of long integers.

### 2.2.16.4 CAUL

The **CAUL** structure defines a counted array of unsigned long integers.



```
typedef struct tagCAUL {
    unsigned long cElems;
    [size_is(cElems)] unsigned long* pElems;
} CAUL;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of unsigned long integers.

#### 2.2.16.5 CAUH

The **CAUH** structure defines a counted array of [ULARGE\\_INTEGER \(section 2.2.17\)](#) values.

```
typedef struct tagCAUH {
    unsigned long cElems;
    [size_is(cElems)] ULARGE_INTEGER* pElems;
} CAUH;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of **ULARGE\_INTEGER** (section 2.2.17) values.

#### 2.2.16.6 CACLSID

The **CACLSID** structure defines a counted array of **GUID** (as specified in [\[MS-DTYP\]](#) section **2.3.4**) values.

```
typedef struct tagCACLSID {
    unsigned long cElems;
    [size_is(cElems)] GUID* pElems;
} CACLSID;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of **GUID** (as specified in [\[MS-DTYP\]](#) section **2.3.4**) values.

#### 2.2.16.7 CALPWSTR

The **CALPWSTR** structure defines a counted array of **LPWSTR** (as specified in [\[MS-DTYP\]](#) section **2.2.35**) values.

```
typedef struct tagCALPWSTR {
    unsigned long cElems;
    [size_is(cElems)] LPWSTR* pElems;
} CALPWSTR;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of **LPWSTR** (as specified in [\[MS-DTYP\]](#) section **2.2.35**) values.

### 2.2.16.8 CAPROPVARIANT

The **CAPROPVARIANT** structure defines a counted array of [PROPVARIANT \(section 2.2.13.2\)](#) values.

```
typedef struct tagCAPROPVARIANT {
    unsigned long cElems;
    [size_is(cElems)] PROPVARIANT* pElems;
} CAPROPVARIANT;
```

**cElems:** MUST be set to the total number of elements of the array.

**pElems:** An array of **PROPVARIANT** (section 2.2.13.2) values.

### 2.2.17 ULARGE\_INTEGER

The **ULARGE\_INTEGER** structure defines a large integer.

```
typedef struct _ULARGE_INTEGER {
    ULONGLONG QuadPart;
} ULARGE_INTEGER;
```

**QuadPart:** A **ULONGLONG** (as specified in [\[MS-DTYP\]](#) section **2.2.55**) value.

## 2.3 PROPID

When making MSMQ-related API function calls, object properties are specified by providing an array of **property identifiers** (a unique **PROPID** value). The associated property values are specified (or returned) in a related array of [PROPVARIANT](#) structures. The values (in decimal), their [PROPVARIANT](#) types (as specified in section [2.2.12](#)), and their associated symbolic names are listed in the **PROPID** sub-sections. Related properties are grouped together within each **PROPID** subsection.

A **PROPID** is an unsigned 32-bit value.

This type is declared as follows:

```
typedef ULONG PROPID;
```

Each directory object type and management type has a set of properties associated with it. The following sections define the property identifier ranges and the properties associated within each range.

Unless otherwise specified, properties are valid for all MSMQ versions.

### 2.3.1 Queue Property Identifiers

Queue properties specify attributes of individual queue objects.

#### 2.3.1.1 PROPID\_Q\_INSTANCE

Value: 101

Variant type: [VT\\_CLSID](#)

Description: Globally unique identifier (GUID) for the queue.

#### 2.3.1.2 PROPID\_Q\_TYPE

Value: 102

Variant type: [VT\\_CLSID](#)

Description: A user-defined value that indicates the type of service that the queue provides. The value is optionally specified at queue creation and can be changed after the queue has been created.

#### 2.3.1.3 PROPID\_Q\_PATHNAME

Value: 103

Variant type: [VT\\_LPWSTR](#)

Description: The path of the queue. The value is specified at queue creation and is immutable thereafter. The value MUST conform to the ABNF for path name (as specified in section [2.1.1](#)).

#### 2.3.1.4 PROPID\_Q\_JOURNAL

Value: 104

Variant type: [VT\\_UI1](#)

Description: A value that specifies how MSMQ tracks messages removed from the queue. MUST be one of the following:

Value	Constant	Description
0	MQ_JOURNAL_NONE	The default. Target journaling is not requested. Messages removed from the destination queue are no longer available.
1	MQ_JOURNAL	Target journaling is requested. Copies of messages are stored in the journal of the queue whenever a receiving application removes a message.

#### 2.3.1.5 PROPID\_Q\_QUOTA

Value: 105

Variant type: [VT\\_UI4](#)

Description: Maximum size (in kilobytes) of a queue.[<1>](#)

### 2.3.1.6 PROPID\_Q\_BASEPRIORITY

Value: 106

Variant type: [VT\\_I2](#)

Description: Priority level of the queue. PROPID\_Q\_BASEPRIORITY applies only to public queues that can be located through the directory service (using a public format name). The base priority of private queues, as well as public queues accessed directly, is always 0. The value MUST be set to a valid priority level. Priority levels are integer values between -32768 (0x8000) and +32767 (0x7fff). The default priority level is 0.

### 2.3.1.7 PROPID\_Q\_JOURNAL\_QUOTA

Value: 107

Variant type: [VT\\_UI4](#)

Description: Maximum size (in kilobytes) of the queue journal. Value may be in the range 0 to 0xffffffff. [<2>](#)

### 2.3.1.8 PROPID\_Q\_LABEL

Value: 108

Variant type: [VT\\_LPWSTR](#)

Description: A descriptive label (maximum 124 characters) for the queue.

### 2.3.1.9 PROPID\_Q\_CREATE\_TIME

Value: 109

Variant type: [VT\\_I4](#)

Description: The time when the queue was created. Time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

### 2.3.1.10 PROPID\_Q\_MODIFY\_TIME

Value: 110

Variant type: [VT\\_I4](#)

Description: The time when the queue properties were last modified. The time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

### 2.3.1.11 PROPID\_Q\_AUTHENTICATE

Value: 111

Variant type: [VT\\_UI1](#)

Description: Authentication level of the queue; MUST be one of the following values.

Value	Constant
0x0	MQ_AUTHENTICATE_NONE
0x01	MQ_AUTHENTICATE

#### 2.3.1.12 PROPID\_Q\_PRIV\_LEVEL

Value: 112

Variant type: [VT\\_UI4](#)

Description: Privacy level of the queue; MUST be one of the following values.

Value	Constant
0x00000000	MQ_PRIV_LEVEL_NONE
0x00000001	MQ_PRIV_LEVEL_OPTIONAL
0x00000002	MQ_PRIV_LEVEL_BODY

MQ\_PRIV\_LEVEL\_NONE : The queue accepts only non-private (clear) messages.

MQ\_PRIV\_LEVEL\_BODY : The queue accepts only private (encrypted) messages.

MQ\_PRIV\_LEVEL\_OPTIONAL : The default. The queue does not enforce privacy. It accepts private (encrypted) messages and non-private (clear) messages.

#### 2.3.1.13 PROPID\_Q\_TRANSACTION

Value: 113

Variant type: [VT\\_UI1](#)

Description: Transaction level of the queue; MUST be one of the following values.

Value	Constant
0x0	MQ_TRANSACTIONAL_NONE
0x01	MQ_TRANSACTIONAL

#### 2.3.1.14 PROPID\_Q\_QMID

Value: 115

Variant type: [VT\\_CLSID](#)

Description: Contains the globally unique identifier (GUID) of the **queue manager** that hosts the queue.

#### 2.3.1.15 PROPID\_Q\_MASTERID

Value: 116

Variant type: [VT\\_CLSID](#)

Description: Contains the GUID of the **Message Queuing Information Store (MQIS)** server that owns the queue; relevant only where MQIS servers are deployed.

#### 2.3.1.16 PROPID\_Q\_FULL\_PATH

Value: 121

Variant type: [VT\\_LPWSTR](#)

Description: Contains the distinguished name (DN) of the queue object in Active Directory (as specified in [\[MS-ADTS\]](#)).

**Note** This property identifier was introduced in MSMQ 2.0.

#### 2.3.1.17 PROPID\_Q\_NAME\_SUFFIX

Value: 123

Variant type: [VT\\_LPWSTR](#)

Description: Contains the suffix of the queue name if the name exceeds 64 characters (the length of the **Common-Name** attribute in Active Directory).

**Note** This property identifier was introduced in MSMQ 2.0.

#### 2.3.1.18 PROPID\_Q\_PATHNAME\_DNS

Value: 124

Variant type: [VT\\_LPWSTR](#)

Description: Contains the Domain Name Systems (DNS) name prefixed path of the queue. The value MUST conform to the ABNF for QueuePathName (as specified in section [2.1.1](#)), where the Computer name is the DNS name of the hosting computer.

**Note** This property identifier was introduced in MSMQ 2.0.

#### 2.3.1.19 PROPID\_Q\_MULTICAST\_ADDRESS

Value: 125

Variant type: [VT\\_LPWSTR](#)

Description: IP multicast address associated with the queue. The property value MUST contain a string that contains a valid multicast address conforming to the ABNF.

```
MulticastAddress = Address ":" Port
Address = 3DIGIT 3("." 1*3DIGIT)
Port=1*5DIGIT
```

The address MUST be in the class D range from 224.0.0.0 to 239.255.255.255. However, only certain ranges of addresses in this range are unreserved and available for sending multicast

messages. For more information and the current list of reserved multicast addresses, see [\[IANAIMA\]](#). There are no restrictions on the port number.

**Note** This property identifier was introduced in MSMQ 3.0.

### 2.3.1.20 PROPID\_Q\_ADS\_PATH

Value: 126

Variant type: [VT\\_LPWSTR](#)

Description: Contains the Active Directory path to the public queue object stored in Active Directory. The value MUST conform to the ABNF for ldapurl (as specified in [\[RFC4516\]](#)).

The following example shows a possible Active Directory path of the queue "MyComp\MyQueue".

LDAP://MyLDAPServer/CN=MyQueue,CN=msmq,CN=MyComp,CN=Computers,DC=MyDomain,DC=MyCompany,DC=COM

**Note** This property identifier was introduced in MSMQ 3.0.

### 2.3.1.21 PROPID\_Q\_SECURITY

Value: 1101

Variant type: [VT\\_BLOB](#)

Description: A security object in Windows NT 4.0 format. The BLOB layout is that of **SECURITY\_DESCRIPTOR**, as specified in [\[MS-DTYP\]](#) section **2.4.6**.

### 2.3.1.22 PROPID\_Q\_OBJ\_SECURITY

Value: 1102

Variant type: [VT\\_BLOB](#)

Description: A security object in Windows NT 4.0 format. The BLOB layout is that of **SECURITY\_DESCRIPTOR** (as specified in [\[MS-DTYP\]](#) section **2.4.6**).

**Note** This property identifier was introduced in MSMQ 2.0.

### 2.3.1.23 PROPID\_Q\_SECURITY\_INFORMATION

Value: 1103

Variant type: [VT\\_UI4](#)

Description: Contains options related to setting or retrieving a security descriptor. It contains [SECURITY\\_INFORMATION](#) (section 2.2.3).

## 2.3.2 Machine Property Identifiers

Machine object property identifiers describe a queue manager.

### 2.3.2.1 PROPID\_QM\_SITE\_ID

Value: 201

Variant type: [VT\\_CLSID](#)

Description: Contains the site identifier globally unique identifier (GUID) of the site in which the queue manager is located.

#### **2.3.2.2 PROPID\_QM\_MACHINE\_ID**

Value: 202

Variant type: [VT\\_CLSID](#)

Description: A GUID that uniquely identifies the queue manager for the computer.

#### **2.3.2.3 PROPID\_QM\_PATHNAME**

Value: 203

Variant type: [VT\\_LPWSTR](#)

Description: The name of the computer where the queue manager is located.

#### **2.3.2.4 PROPID\_QM\_ENCRYPTION\_PK**

Value: 205

Variant type: [VT\\_UI1](#) | [VT\\_VECTOR](#)

Description: The public encryption key of the computer. This property is superseded by [PROPID\\_QM\\_ENCRYPTION\\_PK\\_BASE](#) if present.

#### **2.3.2.5 PROPID\_QM\_CNS**

Value: 207

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: Contains an array of Connected Network identifiers for the connected networks that the queue manager supports.

#### **2.3.2.6 PROPID\_QM\_OUTFRS**

Value: 208

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: An array of Globally Unique Identifiers (GUIDs) of routing servers that act as outgoing interfaces for all MSMQ messages that a given machine sends.

#### **2.3.2.7 PROPID\_QM\_INFRS**

Value: 209

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: An array of Globally Unique Identifiers (GUIDs) of routing servers that act as incoming interfaces for all MSMQ messages that a given machine receives.



### 2.3.2.8 PROPID\_QM\_SERVICE

Value: 210

Variant type: [VT\\_UI4](#)

Description: Indicates the type of service that a given machine supports. The possible values are:

Value	Meaning
0x00000000	The machine does not support any service.
0x00000001	The machine is an <b>MSMQ Routing Server</b> .
0x00000002	The machine is a <b>Backup Site Controller (BSC)</b> .
0x00000004	The machine is a <b>Primary Site Controller (PSC)</b> .
0x00000008	The machine is a <b>Primary Enterprise Controller (PEC)</b> .
0x00000010	The machine is a <b>Remote Access Service (RAS) Server</b> .

### 2.3.2.9 PROPID\_QM\_QUOTA

Value: 214

Variant type: [VT\\_UI4](#)

Description: The disk quota for all queues located at the queue manager. Valid range: 0 to max unsigned 32-bit (0xffffffff).

### 2.3.2.10 PROPID\_QM\_JOURNAL\_QUOTA

Value: 215

Variant type: [VT\\_UI4](#)

Description: Contains the system-wide journal storage quota in kilobytes. Range restrictions are identical to [PROPID\\_QM\\_QUOTA \(section 2.3.2.9\)](#).

### 2.3.2.11 PROPID\_QM\_MACHINE\_TYPE

Value: 216

Variant type: [VT\\_LPWSTR](#)

Description: A description of the operating system version and the MSMQ version. MAY be an empty string [<3>](#), or a version string [<4>](#).

### 2.3.2.12 PROPID\_QM\_CREATE\_TIME

Value: 217

Variant type: [VT\\_LPWSTR](#)

Description: The time when the directory object was created. Time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

### 2.3.2.13 PROPID\_QM\_MODIFY\_TIME

Value: 218

Variant type: [VT\\_LPWSTR](#)

Description: The time when the directory object was last modified. The time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

### 2.3.2.14 PROPID\_QM\_FOREIGN

Value: 219

Variant type: [VT\\_UI1](#)

Description: Indicates if the queue manager is a foreign system that services **foreign queues**. The value **MUST** be one of the following:

Constant	Value
FOREIGN_MACHINE	0x01
MSMQ_MACHINE	0x00

### 2.3.2.15 PROPID\_QM\_OS

Value: 220

Variant type: [VT\\_UI4](#)

Description: A value indicating the operating system type of the queue manager. The value **MUST** be one of the following:

String	Value	Description
MSMQ_OS_NONE	0x00000000	Unknown operating system type
MSMQ_OS_FOREIGN	0x00000100	Not a Windows operating system type
MSMQ_OS_95	0x00000200	Windows 95
MSMQ_OS_NTW	0x00000300	Windows Client
MSMQ_OS_NTS	0x00000400	Windows Server
MSMQ_OS_NTE	0x00000500	Windows Server 2003, Enterprise Edition

### 2.3.2.16 PROPID\_QM\_FULL\_PATH

Value: 221

Variant type: [VT\\_LPWSTR](#)

Description: The distinguishedName for the MSMQ Configuration object. The name **MUST** conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

**Note** Not valid for MSMQ 1.0.

#### 2.3.2.17 PROPID\_QM\_SITE\_IDS

Value: 222

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: Contains an array of site identifiers for sites to which the computer belongs.

**Note** Not valid for MSMQ 1.0.

#### 2.3.2.18 PROPID\_QM\_OUTFRS\_DN

Value: 223

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: An array of distinguished names for MSMQ routing servers through which all outgoing traffic for this computer should be routed. Each name MUST conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

**Note** Not valid for MSMQ 1.0.

#### 2.3.2.19 PROPID\_QM\_INFRS\_DN

Value: 224

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: An array of distinguished names for MSMQ routing servers through which all incoming traffic to this computer should be routed. Each name MUST conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

**Note** Not valid for MSMQ 1.0.

#### 2.3.2.20 PROPID\_QM\_SERVICE\_ROUTING

Value: 227

Variant type: [VT\\_UI1](#)

Description: Indicates whether the queue manager is configured as a routing server. This value SHOULD be settable only by the MSMQ installer. The value MUST be one of the following:

Value	Meaning
0x00	The queue manager is NOT configured as a routing server.
0x01	The queue manager is configured as a routing server.

**Note** Not valid for MSMQ 1.0.

#### 2.3.2.21 PROPID\_QM\_SERVICE\_DSSERVER

Value: 228

Variant type: [VT\\_UI1](#)

Description: Indicates whether the installed version of Microsoft Message Queuing (MSMQ) provides **MSMQ Directory Service (MQDS)** services. This property value is stored in **Active Directory (AD)** as a Boolean.

### 2.3.2.22 PROPID\_QM\_SERVICE\_DEPCLIENTS

Value: 229

Variant type: [VT\\_UI1](#)

Description: Indicates whether the installed version of Microsoft Message Queuing (MSMQ) provides MSMQ **dependent client** services. This property value is stored in Active Directory (AD) as a Boolean.

### 2.3.2.23 PROPID\_QM\_ENCRYPTION\_PK\_BASE

Value: 231

Variant type: [VT\\_UI1](#) | [VT\\_VECTOR](#)

Description: Contains the public encryption key of the computer.

**Note** Not valid for MSMQ 1.0.

### 2.3.2.24 PROPID\_QM\_ENCRYPTION\_PK\_ENHANCED

Value: 232

Variant type: [VT\\_UI1](#) | [VT\\_VECTOR](#)

Description: Contains the enhanced (128-bit) public encryption key of the computer.

**Note** Not valid for MSMQ 1.0.

### 2.3.2.25 PROPID\_QM\_PATHNAME\_DNS

Value: 233

Variant type: [VT\\_LPWSTR](#)

Description: Contains the Domain Name System (DNS) name of the computer.

**Note** Not valid for MSMQ 1.0.

### 2.3.2.26 PROPID\_QM\_OBJ\_SECURITY

Value: 234

Variant type: [VT\\_BLOB](#)

Description: Contains the security descriptor of the MSMQ Configuration object. The BLOB layout is that of **SECURITY\_DESCRIPTOR**, as specified in [\[MS-DTYP\]](#), section [2.4.6](#).

**Note** Not valid for MSMQ 1.0.

### 2.3.2.27 PROPID\_QM\_SECURITY\_INFORMATION

Value: 237

Variant type: [VT\\_UI4](#)

Description: Contains options related to setting or retrieving a security descriptor. The value MUST conform to [SECURITY\\_INFORMATION \(section 2.2.3\)](#).

### 2.3.2.28 PROPID\_QM\_ENCRYPT\_PKS

Value: 238

Variant type: [VT\\_BLOB](#)

Description: The computer's public key certificates used for signing formatted as an [MQDSPUBLICKEYS \(section 2.2.2\)](#) structure.

### 2.3.2.29 PROPID\_QM\_SIGN\_PKS

Value: 239

Variant type: [VT\\_BLOB](#)

Description: The computer's public key certificates used for signing, formatted as an [MQDSPUBLICKEYS \(section 2.2.2\)](#) structure.

### 2.3.2.30 PROPID\_QM\_OWNER\_SID

Value: 241

Variant type: [VT\\_BLOB](#)

Description: Contains the **SID** of the user who ran the setup program. It is passed from the MSMQ service that created the MSMQ Configuration object, so that the server can add it with full control to the discretionary access control list (DACL) of the newly created object. The **SID** layout is specified in [\[MS-DTYP\]](#) section **2.4.2**.

**Note** This property identifier was introduced in MSMQ 2.0.

### 2.3.2.31 PROPID\_QM\_GROUP\_IN\_CLUSTER

Value: 242

Variant type: [VT\\_UI1](#)

Description: Indicates that the MSMQ installation is in a group that is part of a cluster.

Used when creating the MSMQ Configuration objects. The value MUST be one of the following:

Constant	Value
MSMQ_GROUP_NOT_IN_CLUSTER	0x00
MSMQ_GROUP_IN_CLUSTER	0x01

**Note** This property identifier was introduced in MSMQ 2.0.

#### 2.3.2.32 PROPID\_QM\_SECURITY

Value: 1201

Variant type: [VT\\_BLOB](#)

Description: Security information in Windows NT 4.0 format. The layout of the **BLOB** is specified in [\[MS-DTYP\]](#) section 2.4.6.

#### 2.3.2.33 PROPID\_QM\_SIGN\_PK

Value: 1202

Variant type: [VT\\_BLOB](#)

Description: The computer's public key certificates used for signing, formatted as an [MQDSPUBLICKEYS \(section 2.2.2\)](#) structure. This property can be specified only at object creation time.

#### 2.3.2.34 PROPID\_QM\_ENCRYPT\_PK

Value: 1203

Variant type: [VT\\_BLOB](#)

Description: The computer's public key certificates used for encryption, formatted as an [MQDSPUBLICKEYS \(section 2.2.2\)](#) structure. This property can be specified only at object creation time.

#### 2.3.2.35 PROPID\_QM\_UPGRADE\_DACL

Value: 1205

Variant type: [VT\\_BLOB](#)

Description: A dummy [PROPID](#). It is used only in a set property operation to request that the Primary Enterprise Controller (PEC) update the discretionary access control list (DACL) of the calling computer. The **BLOB** MAY be empty. The server MUST ignore the value.

### 2.3.3 Site Property Identifiers

Site property identifiers pertain to the site object.

#### 2.3.3.1 PROPID\_S\_PATHNAME

Value: 301

Variant type: [VT\\_LPWSTR](#)

Description: Contains the name of the site.

#### 2.3.3.2 PROPID\_S\_SITEID

Value: 302

Variant type: [VT\\_CLSID](#)

Description: Contains the identifier of the site.

#### 2.3.3.3 PROPID\_S\_GATES

Value: 303

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: Contains the GUIDs of the MSMQ Configuration objects of the MSMQ servers that are the gates for this site.

#### 2.3.3.4 PROPID\_S\_PSC

Value: 304

Variant type: [VT\\_LPWSTR](#)

Description: Contains the computer name of the Primary Site Controller (PSC) for the site.

#### 2.3.3.5 PROPID\_S\_INTERVAL1

Value: 305

Variant type: [VT\\_UI2](#)

Description: In **MSMQ mixed-mode**, the default replication time (in seconds) within an **MSMQ Site**. The default is 2 seconds.

#### 2.3.3.6 PROPID\_S\_INTERVAL2

Value: 306

Variant type: [VT\\_UI2](#)

Description: In MSMQ mixed-mode, the default replication time (in seconds) between MSMQ Sites. The default is 10 seconds.

#### 2.3.3.7 PROPID\_S\_FULL\_NAME

Value: 309

Variant type: [VT\\_LPWSTR](#)

Description: Contains the distinguished name (DN) of the site in Active Directory. (The name format is specified in [\[MS-ADTS\]](#).)

**Note** Not valid for MSMQ 1.0.

#### 2.3.3.8 PROPID\_S\_NT4\_STUB

Value: 310

Variant type: [VT\\_UI2](#)

Description: Specifies whether the site was migrated from a Windows NT 4.0 Message Queuing Information Store (MQIS) database. The value MUST be one of the following:

Value	Description
0x01	Site was migrated from MQIS.
0x00	Site was not migrated.

**Note** Not valid for MSMQ 1.0.

#### 2.3.3.9 PROPID\_S\_FOREIGN

Value: 311

Variant type: [VT\\_UI1](#)

Description: Specifies whether the site is used as a definition of an external messaging system. The value MUST be one of the following:

Value	Description
0x01	Site is an external system.
0x00	Site is not external.

**Note** Not valid for MSMQ 1.0.

#### 2.3.3.10 PROPID\_S\_SECURITY

Value: 1301

Variant type: [VT\\_BLOB](#)

Description: A security object in Windows NT 4.0 format. The [BLOB](#) layout is that of **SECURITY\_DESCRIPTOR**, as specified in [\[MS-DTYP\]](#) section **2.4.6**.

#### 2.3.3.11 PROPID\_S\_PSC\_SIGNPK

Value: 1302

Variant type: [VT\\_BLOB](#)

Description: Contains the signing key of the Primary Site Controller (PSC) formatted as an [MQDSPUBLICKEYS](#) (section 2.2.2) structure.

#### 2.3.3.12 PROPID\_S\_SECURITY\_INFORMATION

Value: 1303

Variant type: [VT\\_UI4](#)

Description: The [SECURITY\\_INFORMATION](#) (section 2.2.3) associated with setting or retrieving a security descriptor.

**Note** Not valid for MSMQ 1.0 or MSMQ 2.0.



## 2.3.4 Connected Network Property Identifiers

Connected Network object properties contain attributes of a connected network.

### 2.3.4.1 PROPID\_CN\_SECURITY

Value: 1501

Variant type: [VT\\_BLOB](#)

Description: A security object in Windows NT 4.0 format. The [BLOB](#) layout is that of **SECURITY\_DESCRIPTOR**, as specified in [\[MS-DTYP\]](#) section **2.4.6**.

## 2.3.5 Enterprise Object Property Identifiers

**Enterprise** object properties pertain to enterprise-wide settings.

### 2.3.5.1 PROPID\_E\_NAMESTYLE

Value: 602

Variant type: [VT\\_UI1](#)

Description: Indicates whether weakened security is enabled. Value MUST be one of the following:

Value	Description
0x00	Weakened security is not enabled.
0x01	Weakened security is enabled.

### 2.3.5.2 PROPID\_E\_CSP\_NAME

Value: 603

Variant type: [VT\\_LPWSTR](#)

Description: The type of cryptographic provider used by MSMQ. The default value is "Microsoft Enhanced RSA and AES Cryptographic Provider".

### 2.3.5.3 PROPID\_E\_ID

Value: 609

Variant type: [VT\\_CLSID](#)

Description: The GUID identifier for the directory object instance.

### 2.3.5.4 PROPID\_E\_LONG\_LIVE

Value: 616

Variant type: [VT\\_UI4](#)

Description: The default value for the time, in seconds, that a message has to reach a queue when sending MSMQ messages.

### 2.3.5.5 PROPID\_E\_VERSION

Value: 617

Variant type: [VT\\_UI2](#)

Description: The version number of MSMQ Directory Service (MQDS) information.

### 2.3.5.6 PROPID\_E\_SECURITY

Value: 1601

Variant type: [VT\\_BLOB](#)

Description: The security descriptor of the MSMQ services object in Windows NT 4.0 format. The [BLOB](#) layout is that of **SECURITY\_DESCRIPTOR**, as specified in [\[MS-DTYP\]](#) section **2.4.6**.

## 2.3.6 User Object Property Identifiers

User object properties are used by MSMQ during management of user certificates that are stored in the MSMQ Directory Service.

### 2.3.6.1 PROPID\_U\_SID

Value: 701

Variant type: [VT\\_BLOB](#)

Description: Contains the user's security identifier (SID). The [BLOB](#) layout of a **SID** is specified in [\[MS-DTYP\]](#) section **2.4.2**.

### 2.3.6.2 PROPID\_U\_SIGN\_CERT

Value: 702

Variant type: [VT\\_BLOB](#)

Description: Contains an X.509 encoded certificate for the user object as specified in [\[RFC3280\]](#).

### 2.3.6.3 PROPID\_U\_DIGEST

Value: 705

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: Contains an array of certificate digests. Each digest is computed as the MD5 hash of the encoded certificate. Each array element **MUST** contain the 16-byte output of the MD5 algorithm, as specified in [\[RFC1321\]](#).

### 2.3.6.4 PROPID\_U\_ID

Value: 706

Variant type: [VT\\_CLSID](#)

Description: The GUID identifying the user object.

### 2.3.7 Sitelink Property Identifiers

Sitelink properties define the cost of routing a message from one site to another.

#### 2.3.7.1 PROPID\_L\_NEIGHBOR1

Value: 801

Variant type: [VT\\_CLSID](#)

Description: Contains the GUID of one of the routing sites.

#### 2.3.7.2 PROPID\_L\_NEIGHBOR2

Value: 802

Variant type: [VT\\_CLSID](#)

Description: Contains the GUID of the other routing site.

#### 2.3.7.3 PROPID\_L\_COST

Value: 803

Variant type: [VT\\_UI4](#)

Description: Contains the cost of the link. Each routing link is assigned a relative cost, which may reflect the speed or the monetary cost of the underlying physical communication link. The default value is 1; and costs can range from 1 to 999999, inclusive.

#### 2.3.7.4 PROPID\_L\_ID

Value: 806

Variant type: [VT\\_CLSID](#)

Description: Contains the GUID of the site link object.

#### 2.3.7.5 PROPID\_L\_GATES\_DN

Value: 807

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: Contains the distinguished names of the MSMQ Configuration object of the computers that are site gates on the link. Each name MUST conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

**Note** Not valid for MSMQ 1.0 or MSMQ 2.0.

#### 2.3.7.6 PROPID\_L\_NEIGHBOR1\_DN

Value: 808

Variant type: [VT\\_LPWSTR](#)

Description: Contains the distinguished name of one site on the link. The name MUST conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

**Note** Not valid for MSMQ 1.0.

#### 2.3.7.7 PROPID\_L\_NEIGHBOR2\_DN

Value: 809

Variant type: [VT\\_LPWSTR](#)

Description: Contains the distinguished name of the other site on the link. The name MUST conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

**Note** Not valid for MSMQ 1.0.

#### 2.3.7.8 PROPID\_L\_DESCRIPTION

Value: 810

Variant type: [VT\\_LPWSTR](#)

Description: Contains the description of the routing link.

**Note** Not valid for MSMQ 1.0.

#### 2.3.7.9 PROPID\_L\_FULL\_PATH

Value: 811

Variant type: [VT\\_LPWSTR](#)

Description: Contains the distinguished name of the routing link object in the Active Directory. The name MUST conform to ABNF: distinguishedName, as described in [\[RFC4514\]](#).

**Note** Not valid for MSMQ 1.0.

#### 2.3.7.10 PROPID\_L\_ACTUAL\_COST

Value: 812

Variant type: [VT\\_UI4](#)

Description: Contains the untranslated link cost. The value MUST be in the range from 1 to 999999, inclusive.

**Note** Not valid for MSMQ 1.0.

#### 2.3.7.11 PROPID\_L\_GATES

Value: 813

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: Contains the GUIDs of the MSMQ Configuration objects of the computers that are site gates on the link.

**Note** Not valid for MSMQ 1.0.

### 2.3.8 Settings Property Identifiers

Setting objects represent MSMQ Routing Servers or **MSMQ Directory Service Servers**.

**Note** All Settings properties are not valid for MSMQ 1.0.

#### 2.3.8.1 PROPID\_SET\_NAME

Value: 5101

Variant type: [VT\\_LPWSTR](#)

Description: Contains the **Common-Name** attribute, which MUST always be set to the string "MSMQ Settings".

#### 2.3.8.2 PROPID\_SET\_SERVICE

Value: 5102

Variant type: [VT\\_UI4](#)

Description: Contains a value that identifies the type of service. The value MUST be one of the following:

Value	Description
0x00000000	None
0x00000001	Routing server (SRV)
0x00000002	Backup Site Controller (BSC)
0x00000004	Primary Site Controller (PSC)
0x00000008	Primary Enterprise Controller (PEC)

#### 2.3.8.3 PROPID\_SET\_QM\_ID

Value: 5103

Variant type: [VT\\_CLSID](#)

Description: Contains the GUID of the computer's MSMQ Configuration object.

#### 2.3.8.4 PROPID\_SET\_FULL\_PATH

Value: 5105

Variant type: [VT\\_LPWSTR](#)

Description: Contains the distinguished name of the MSMQ Settings object in the Active Directory. The name MUST conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

### 2.3.8.5 PROPID\_SET\_NT4

Value: 5106

Variant type: [VT\\_UI1](#)

Description: Specifies whether the server is MSMQ 1.0.

The value MUST be one of the following:

Value	Description
0x01	Server is MSMQ 1.0.
0x00	Server is not MSMQ 1.0.

### 2.3.8.6 PROPID\_SET\_MASTERID

Value: 5107

Variant type: [VT\\_CLSID](#)

Description: Contains the GUID of the site.

### 2.3.8.7 PROPID\_SET\_SITENAME

Value: 5108

Variant type: [VT\\_LPWSTR](#)

Description: Contains the site name.

### 2.3.8.8 PROPID\_SET\_SERVICE\_ROUTING

Value: 5109

Variant type: [VT\\_UI1](#)

Description: Specifies whether the server is a routing server. The value MUST be one of the following:

Value	Description
0x01	Server is a routing server.
0x0	Server is not a routing server.

### 2.3.8.9 PROPID\_SET\_SERVICE\_DSSERVER

Value: 5110

Variant type: [VT\\_UI1](#)

Description: Specifies whether the **MSMQ server** provides access to the Active Directory for MSMQ server 2.0 clients. The value MUST be set to one of the following:

Value	Description
0x01	Server provides access to Active Directory.
0x00	Server does not provide access to Active Directory.

#### 2.3.8.10 PROPID\_SET\_SERVICE\_DEPCLIENTS

Value: 5111

Variant type: [VT\\_UI1](#)

Description: Specifies whether the server can be a supporting server for dependent clients. The value **MUST** be set to one of the following:

Value	Description
0x01	Server can be a supporting server.
0x00	Server cannot be a supporting server.

#### 2.3.8.11 PROPID\_SET\_OLDSERVICE

Value: 5112

Variant type: [VT\\_UI4](#)

Description: Contains a value that identifies the type of service. The value **MUST** be set to one of the following:

Value	Description
0x00000000	None
0x00000001	Routing server (SRV)
0x00000002	Backup Site Controller (BSC)
0x00000004	Primary Site Controller (PSC)
0x00000008	Primary Enterprise Controller (PEC)

### 2.3.9 MQUser Property Identifiers

These properties represent attributes of users who migrated to Active Directory from the Microsoft Message Queuing Information Store (MQIS).

**Note** These values are not valid for MSMQ server 1.0.

#### 2.3.9.1 PROPID\_MQU\_SID

Value: 5401

Variant type: [VT\\_BLOB](#)

Description: The migrated user's SID, as specified in [\[MS-DTYP\]](#) section **2.4.2**.

### 2.3.9.2 PROPID\_MQU\_SIGN\_CERT

Value: 5402

Variant type: [VT\\_BLOB](#)

Description: Contains an X.509 encoded certificate for the migrated user object as specified in [\[RFC3280\]](#).

### 2.3.9.3 PROPID\_MQU\_DIGEST

Value: 5405

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: Contains an array of certificate digests. Each digest is computed as the MD5 hash of the encoded certificate. Each array element MUST contain the 16-byte output of the MD5 algorithm, as specified in [\[RFC1321\]](#).

### 2.3.9.4 PROPID\_MQU\_ID

Value: 5406

Variant type: [VT\\_CLSID](#)

Description: Contains the GUID of the MQUser object.

### 2.3.9.5 PROPID\_MQU\_SECURITY

Value: 5407

Variant type: [VT\\_BLOB](#)

Description: A security object in Windows NT 4.0 format. The [BLOB](#) layout is that of **SECURITY\_DESCRIPTOR**, as specified in [\[MS-DTYP\]](#) section **2.4.6**.

## 2.3.10 Computer Property Identifiers

Computer properties contain attributes of the computer object.

**Note** These values are not valid for MSMQ server 1.0.

### 2.3.10.1 PROPID\_COM\_FULL\_PATH

Value: 5201

Variant type: [VT\\_LPWSTR](#)

Description: Contains the distinguished name of the computer. The name MUST conform to ABNF: distinguishedName, as specified in [\[RFC4514\]](#).

### 2.3.10.2 PROPID\_COM\_SAM\_ACCOUNT

Value: 5202

Variant type: [VT\\_LPWSTR](#)



Description: Identifies a property that contains the name of the computer account in Active Directory. Contains the name of the computer object. The value is represented as the computer name (truncated to 19 characters) followed by a "\$" character. For example, the property value for a computer with the name "MyComputer" is "MyComputer\$".

### **2.3.10.3 PROPID\_COM\_ACCOUNT\_CONTROL**

Value: 5204

Variant type: [VT\\_UI4](#)

Description: Contains user account control attributes, as specified in [\[MS-SAMR\]](#). The value MUST be a bit mask computed as a logical OR of a set of UF\_FLAG codes, as specified in [\[MS-SAMR\]](#) section **2.2.1.13**.

### **2.3.10.4 PROPID\_COM\_DNS\_HOSTNAME**

Value: 5205

Variant type: [VT\\_LPWSTR](#)

Description: Contains the Domain Name Service (DNS) host name attribute of the computer object. The value MUST contain the DNS name of the computer.

### **2.3.10.5 PROPID\_COM\_SID**

Value: 5206

Variant type: [VT\\_BLOB](#)

Description: Contains the SID of the computer object. This property is read from the Active Directory during creation of an MSMQ server Configuration object, and is used to add the computer SID to the MSMQ server configuration Discretionary Access Control List (DACL). The [BLOB](#) layout of **SID** is specified in [\[MS-DTYP\]](#) section **2.4.2**.

### **2.3.10.6 PROPID\_COM\_SIGN\_CERT**

Value: 5207

Variant type: [VT\\_BLOB](#)

Description: Contains an X.509 encoded certificate for the computer object. The X.509 encoded certificate is specified in [\[RFC3280\]](#).

### **2.3.10.7 PROPID\_COM\_DIGEST**

Value: 5208

Variant type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: Contains an array of certificate digests. The digest is computed as the MD5 hash of the encoded certificate. Each value MUST contain the 16-byte output of the MD5 algorithm, as specified in [\[RFC1321\]](#).

### 2.3.10.8 PROPID\_COM\_ID

Value: 5209

Variant type: [VT\\_CLSID](#)

Description: Contains the **GUID** (as specified in [\[MS-DTYP\]](#) section **2.3.4**) of the computer object.

### 2.3.11 Management Machine Property Identifiers

Management machine property identifiers provide values that identify properties that describe local administration of MSMQ machines.

#### 2.3.11.1 PROPID\_MGMT\_MSMQ\_ACTIVEQUEUES

Value: 1

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: A list of all the active queue names on the computer. Each name **MUST** conform to the ABNF for a format name, as specified in section [2.1](#).

#### 2.3.11.2 PROPID\_MGMT\_MSMQ\_PRIVATEQ

Value: 2

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: A list of the path names of all the private queues registered on the computer.

#### 2.3.11.3 PROPID\_MGMT\_MSMQ\_DSSERVER

Value: 3

Variant type: [VT\\_LPWSTR](#)

Description: The name of the current MSMQ Directory Service server for the computer. The pointer to a null-terminated string that specifies the computer name of the discovered server. The returned computer name is prefixed with "\\".

ABNF format:

DS = "\\\" Name

Alpha = %x41-5A / %x61-7A

Name = 1\*Alpha

#### 2.3.11.4 PROPID\_MGMT\_MSMQ\_CONNECTED

Value: 4

Variant type: [VT\\_LPWSTR](#)

Description: The value that indicates whether the queue manager on the computer has been disconnected from the network. The value MUST be one of the following strings.

Value	Constant
"CONNECTED"	MSMQ_CONNECTED
"DISCONNECTED"	MSMQ_DISCONNECTED

#### 2.3.11.5 PROPID\_MGMT\_MSMQ\_TYPE

Value: 5

Variant type: [VT\\_LPWSTR](#)

Description: The version and build information for the computer operating system and MSMQ installation.

#### 2.3.11.6 PROPID\_MGMT\_MSMQ\_BYTES\_IN\_ALL\_QUEUES

Value: 6

Variant type: [VT\\_I8](#)

Description: The number of message bytes stored in all the queues on the computer.

**Note** Not valid for MSMQ 1.0 and MSMQ 2.0.

#### 2.3.12 Management Queue Property Identifiers

Management **queue property** identifiers provide values that identify properties for monitoring the MSMQ installation and the queues on a computer, which allows applications to manage these resources programmatically.

##### 2.3.12.1 PROPID\_MGMT\_QUEUE\_PATHNAME

Value: 1

Variant type: [VT\\_LPWSTR](#)

Description: The path name of the queue. The path name format is specified in section [2.1.1](#).

##### 2.3.12.2 PROPID\_MGMT\_QUEUE\_FORMATNAME

Value: 2

Variant type: [VT\\_LPWSTR](#)

Description: The format name of the queue, as specified in section [2.1](#).

##### 2.3.12.3 PROPID\_MGMT\_QUEUE\_TYPE

Value: 3

Variant type: [VT\\_LPWSTR](#)

Description: The type of the queue. The value MUST be one of the following strings.

Value	Constant
"PUBLIC"	MGMT_QUEUE_TYPE_PUBLIC
"PRIVATE"	MGMT_QUEUE_TYPE_PRIVATE
"MACHINE"	MGMT_QUEUE_TYPE_MACHINE
"CONNECTOR"	MGMT_QUEUE_TYPE_CONNECTOR
"MULTICAST"	MGMT_QUEUE_TYPE_MULTICAST

#### 2.3.12.4 PROPID\_MGMT\_QUEUE\_LOCATION

Value: 4

Variant type: [VT\\_LPWSTR](#)

Description: The value that indicates whether the queue is located on the computer. The value MUST be one of the following strings.

Value	Constant
"LOCAL"	MGMT_QUEUE_LOCAL_LOCATION
"REMOTE"	MGMT_QUEUE_REMOTE_LOCATION

#### 2.3.12.5 PROPID\_MGMT\_QUEUE\_XACT

Value: 5

Variant type: [VT\\_LPWSTR](#)

Description: The value that indicates whether the queue is transactional. The value MUST be one of the following strings.

Value	Constant
"UNKNOWN"	MGMT_QUEUE_UNKNOWN_TYPE
"YES"	MGMT_QUEUE_TRANSACTIONAL_TYPE
"NO"	MGMT_QUEUE_NOT_TRANSACTIONAL_TYPE

#### 2.3.12.6 PROPID\_MGMT\_QUEUE\_FOREIGN

Value: 6

Variant type: [VT\\_LPWSTR](#)

Description: The string that indicates whether the queue is a foreign queue. The value MUST be one of the following strings.

Value	Constant
"UNKNOWN"	MGMT_QUEUE_UNKNOWN_TYPE
"YES"	MGMT_QUEUE_FOREIGN_TYPE
"NO"	MGMT_QUEUE_NOT_FOREIGN_TYPE

### 2.3.12.7 PROPID\_MGMT\_QUEUE\_MESSAGE\_COUNT

Value: 7

Variant type: [VT\\_UI4](#)

Description: The number of messages in the queue.

### 2.3.12.8 PROPID\_MGMT\_QUEUE\_BYTES\_IN\_QUEUE

Value: 8

Variant type: [VT\\_UI4](#)

Description: The number of message bytes for all messages in the queue.

**Note** This property identifier is available only in MSMQ 3.0 and later versions. It replaces PROPID\_MGMT\_QUEUE\_JOURNAL\_USED\_QUOTA from MSMQ 1.0 and MSMQ 2.0.

### 2.3.12.9 PROPID\_MGMT\_QUEUE\_JOURNAL\_MESSAGE\_COUNT

Value: 9

Variant type: [VT\\_UI4](#)

Description: The number of messages in the queue journal.

### 2.3.12.10 PROPID\_MGMT\_QUEUE\_BYTES\_IN\_JOURNAL

Value: 10

Variant type: [VT\\_UI4](#)

Description: The number of message bytes for all messages in the queue journal.

**Note** This property identifier is available only in MSMQ 3.0 and later versions. It replaces PROPID\_MGMT\_QUEUE\_JOURNAL\_USED\_QUOTA from MSMQ 1.0 and MSMQ 2.0.

### 2.3.12.11 PROPID\_MGMT\_QUEUE\_STATE

Value: 11

Variant type: [VT\\_LPWSTR](#)

Description: The connection state of the **outgoing queue**. The value MUST be one of the following strings.

Value	Constant
"LOCAL CONNECTION"	MGMT_QUEUE_STATE_LOCAL
"INACTIVE"	MGMT_QUEUE_STATE_NONACTIVE
"WAITING"	MGMT_QUEUE_STATE_WAITING
"NEED VALIDATION"	MGMT_QUEUE_STATE_NEED_VALIDATE
"ONHOLD"	MGMT_QUEUE_STATE_ONHOLD
"CONNECTED"	MGMT_QUEUE_STATE_CONNECTED
"DISCONNECTING"	MGMT_QUEUE_STATE_DISCONNECTING
"DISCONNECTED"	MGMT_QUEUE_STATE_DISCONNECTED

### 2.3.12.12 PROPID\_MGMT\_QUEUE\_NEXTHOPS

Value: 12

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: The address, or a list of possible addresses, for routing messages to the destination queue in the next hop. If the queue is in the process of being connected, a list of possible addresses is returned. Each element conforms to the following ABNF:

```

Address = IPAddress/ForeignAddress/IPv6Address

IPAddress = "IP=" AddDigits 3("." AddDigits)

ForeignAddress = "FOREIGN=" Guid

IPv6Address = "IPv6=" [4HexDig] 7(":" [4HexDig])

Guid = 8HexDig %x2D 3(4HexDig %x2D) 12HexDig

HexDig = Digit / "A" / "B" / "C" / "D" / "E" / "F"

Digit = %x30-39

AddDigits = 1*3Digit

```

### 2.3.12.13 PROPID\_MGMT\_QUEUE\_EOD\_LAST\_ACK

Value: 13

Variant type: [VT\\_BLOB](#)

Description: The sequence information about the last message sent from the computer to the queue for which an order acknowledgment was received. The [BLOB](#) layout of the **SEQUENCE\_INFO** structure is specified in section [2.2.5](#).

#### **2.3.12.14 PROPID\_MGMT\_QUEUE\_EOD\_LAST\_ACK\_TIME**

Value: 14

Variant type: [VT\\_I4](#)

Description: The date and time when the last order acknowledgment for a message sent from the computer to the queue was received. Time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

#### **2.3.12.15 PROPID\_MGMT\_QUEUE\_EOD\_LAST\_ACK\_COUNT**

Value: 15

Variant type: [VT\\_UI4](#)

Description: The number of times that the last order acknowledgment for a message sent from the computer to the queue was received.

#### **2.3.12.16 PROPID\_MGMT\_QUEUE\_EOD\_FIRST\_NON\_ACK**

Value: 16

Variant type: [VT\\_BLOB](#)

Description: The sequence information about the first message sent from the computer to the queue for which no order acknowledgment was received. The [BLOB](#) layout of the **SEQUENCE\_INFO** structure is specified in section [2.2.5](#).

#### **2.3.12.17 PROPID\_MGMT\_QUEUE\_EOD\_LAST\_NON\_ACK**

Value: 17

Variant type: [VT\\_BLOB](#)

Description: The sequence information about the last message that was sent from the computer to the queue for which no order acknowledgment was received. The [BLOB](#) layout of the **SEQUENCE\_INFO** structure is specified in section [2.2.5](#).

#### **2.3.12.18 PROPID\_MGMT\_QUEUE\_EOD\_NEXT\_SEQ**

Value: 18

Variant type: [VT\\_BLOB](#)

Description: The sequence information about the next message to be sent from the computer to the queue. The [BLOB](#) layout of the **SEQUENCE\_INFO** structure is specified in section [2.2.5](#).

#### **2.3.12.19 PROPID\_MGMT\_QUEUE\_EOD\_NO\_READ\_COUNT**

Value: 19

Variant type: [VT\\_UI4](#)

Description: The number of messages sent from the computer to the queue for which an order acknowledgment was received, but for which a receive acknowledgment message was not received.

### 2.3.12.20 PROPID\_MGMT\_QUEUE\_EOD\_NO\_ACK\_COUNT

Value: 20

Variant type: [VT\\_UI4](#)

Description: The number of messages sent from the computer to the queue for which no order acknowledgment was received.

### 2.3.12.21 PROPID\_MGMT\_QUEUE\_EOD\_RESEND\_TIME

Value: 21

Variant type: [VT\\_I4](#)

Description: The time at which MSMQ will attempt to send a message from the computer to the queue again. Time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

### 2.3.12.22 PROPID\_MGMT\_QUEUE\_EOD\_RESEND\_INTERVAL

Value: 22

Variant type: [VT\\_UI4](#)

Description: The resend interval (in seconds) for the messages in the outgoing queue for which no order acknowledgment was received.

### 2.3.12.23 PROPID\_MGMT\_QUEUE\_EOD\_RESEND\_COUNT

Value: 23

Variant type: [VT\\_UI4](#)

Description: The number of times that the last message in the corresponding outgoing queue on the computer was sent.

### 2.3.12.24 PROPID\_MGMT\_QUEUE\_EOD\_SOURCE\_INFO

Value: 24

Variant type: [VT\\_VARIANT](#) | [VT\\_VECTOR](#)

Description: The array of arrays of information about the **transactional messages** sent from all source computers to the queue on the target computer.

The array contains the following six items. Each item is an array; there is one entry in each array for each message.

- **Format name**

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: Each entry is a format name of a queue, as specified in section [2.1](#).

- **Sender ID**



Variant type: [VT\\_CLSID](#) | **VT\_VECTOR**

Description: Each entry is a GUID of the sender of the message.

- **Sequence ID**

Variant type: [VT\\_UI8](#) | **VT\_VECTOR**

Description: Each entry is a number to distinguish a sequence from other sequences.

- **Sequence number**

Variant type: [VT\\_UI4](#) | **VT\_VECTOR**

Description: Each entry is a sequence number.

- **Last access time**

Variant type: [VT\\_I4](#) | **VT\_VECTOR**

Description: Each entry was the time when the queue was accessed. Time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

- **Message reject count**

Variant type: **VT\_UI4** | **VT\_VECTOR**

Description: Each entry is the number of times that a message was rejected.

### 2.3.12.25 PROPID\_MGMT\_QUEUE\_CONNECTION\_HISTORY

Value: 25

Variant type: 12 | 0x1000 - [VT\\_VARIANT](#) | [VT\\_VECTOR](#)

Description: The queue connection state history. The array contains the following four items.

- **Status**

Variant type: [VT\\_UI4](#)

Description: The connection status and cause of the failure. The contents MUST be one of the following values.

Value	Description
0x00000000	Connection is in the process of establishment; no failures have occurred.
0x00000001	Connection establishment packet has been received.
0x00000002	Connection has been successfully established and is ready to send messages.
0x80000000	Exact reason for failure cannot be determined.
0x80000001	Ping failure.
0x80000002	Create socket failure.

Value	Description
0x80000003	Bind socket failure.
0x80000004	Connect socket failure.
0x80000005	TCP is not enabled.
0x80000006	Send operation on a socket failed.
0x80000007	Send operation failed because connection is not ready.
0x80000008	Domain Name Service (DNS) failure.
0x80000009	Could not validate server certificate in HTTPS scenario.
0x8000000A	Connection limit reached, cannot establish new session to a specific destination.
0x8000000B	Connection refused by other side due to any reason (quota, invalid packet, connection limit reached).
0x8000000C	Absence of MSMQ Directory Service server connectivity prevents getting routing data.
0x8000000D	Failure due to low memory.

- **Time at which the failure occurred**

Variant type: **VT\_I4**

Description: The time is represented as the number of seconds elapsed since midnight (00:00:00), January 1, 1970 (Coordinated Universal Time).

- **Error indicator**

Variant type: **VT\_I4**

Description: A value other than 0 indicates an error.

- **List of addresses**

Variant type: [VT\\_LPWSTR](#) | **VT\_VECTOR**

Description: An address or a list of possible addresses, for routing messages to the destination queue in the next hop. Each element MUST conform to the following ABNF:

```

Address = Name / IPaddress
Name = "\\\" 1*Alpha
Alpha = %x41-5A / %x61-7A
IPaddress = AddDigits 3("." AddDigits)
Digit = %x30-39
AddDigits = 1*3Digit

```

### 2.3.12.26 PROPID\_MGMT\_QUEUE\_SUBQUEUE\_COUNT

Value: 26

Variant type: [VT\\_UI4](#)

Description: The count of the number of subqueues for a given queue.

### 2.3.12.27 PROPID\_MGMT\_QUEUE\_SUBQUEUE\_NAMES

Value: 27

Variant type: [VT\\_LPWSTR](#) | [VT\\_VECTOR](#)

Description: The list of subqueue names (as specified in section [2.1](#)) for a given queue.

## 2.4 Active Directory Properties

The following are Active Directory (AD) properties with no equivalent MSMQ properties.

### 2.4.1 msMQ-Recipient-FormatName

Type: String

Description: Used as the recipient format name in the msMQ-Custom-Recipient class. It is a **Unicode** string with a size between 1 to 255 characters, inclusive.

### 2.4.2 mSMQDigestsMig

Type: String

Description: Used in MSMQ mixed-mode, contains the previous value of [mSMQDigests](#), which corresponds to the [PROPID\\_U\\_DIGEST](#) property in MSMQ and contains an array of certificate digests. Each digest is computed as the MD5 hash of the encoded certificate. Each array element MUST contain the 16-byte output of the MD5 algorithm, as specified in [\[RFC1321\]](#).

### 2.4.3 mSMQMigrated

Type: Boolean

Description: The [mSMQMigrated](#) attribute contains MSMQ mixed-mode information. It is used only in migration.

### 2.4.4 mSMQSignCertificatesMig

Type: [VT\\_BLOB](#)

Description: In MSMQ mixed-mode, the attribute contains the previous value of [mSMQSignCertificates](#). MSMQ supports migration from MSMQ Directory Service (MQDS) version 1.0 to Windows 2000 DS, and mixed mode specifies a state in which one or more of the DS servers were not upgraded to Windows 2000. Used only in migration. The MSMQ property for [mSMQSignCertificates](#) is [PROPID\\_U\\_SIGN\\_CERT](#), which contains an X.509 encoded certificate for the user object as specified in [\[RFC3280\]](#).

### 2.4.5 mSMQSiteGatesMig

Type: [VT\\_CLSID](#) | [VT\\_VECTOR](#)

Description: In MSMQ mixed-mode, the previous value of [mSMQSiteGates](#). The MSMQ property for [mSMQSiteGates](#) is [PROPID\\_L\\_GATES](#), which contains the GUIDs of the MSMQ Configuration objects of the computers that are **MSMQ Site Gates** on an **MSMQ Site Link**.

## 2.5 Unused Active Directory Properties

The following Active Directory (AD) properties are present in the AD schema but are not used in any MSMQ version:

- [mSMQSiteID](#)
- [mSMQPrevSiteGates](#)
- [MSMQ-SecuredSource](#)
- [mSMQComputerType](#)
- [mSMQLabel](#) (has been replaced with [mSMQLabelEx](#))
- [mSMQSiteName](#) (has been replaced with [mSMQSiteNameEx](#))

## 2.6 PROPID to Active Directory Schema Mapping

The implementation of the MSMQ Directory Service (MQDS) has evolved over MSMQ versions. In MSMQ 1.0, the Directory Service is implemented as the Message Queue Information Service and is accessed via the [Message Queuing \(MSMQ\): Directory Service Protocol](#), as specified in [MS-MQDS].

In MSMQ 2.0 and later versions, the Directory Service is implemented using Active Directory (AD) and can also be accessed directly via the Lightweight Directory Access Protocol (for more information, see [LDAP](#)). The mapping between PROPID and Active Directory Schema classes (as specified in [\[MS-ADTS\]](#)) is in the following table.

Directory Service properties	Active Directory class	Active Directory properties
<a href="#">PROPID_Q_INSTANCE</a> <a href="#">PROPID_Q_TYPE</a> <a href="#">PROPID_Q_PATHNAME</a> <a href="#">PROPID_Q_JOURNAL</a> <a href="#">PROPID_Q_QUOTA</a> <a href="#">PROPID_Q_BASEPRIORITY</a> <a href="#">PROPID_Q_JOURNAL_QUOTA</a> <a href="#">PROPID_Q_LABEL</a> <a href="#">PROPID_Q_CREATE_TIME</a> <a href="#">PROPID_Q_MODIFY_TIME</a> <a href="#">PROPID_Q_AUTHENTICATE</a> <a href="#">PROPID_Q_PRIV_LEVEL</a> <a href="#">PROPID_Q_TRANSACTION</a> <a href="#">PROPID_Q_MASTERID</a> <a href="#">PROPID_Q_SECURITY</a>	mSMQQueue (MSMQ-Queue)	<a href="#">objectGUID</a> <a href="#">mSMQQueueType</a> <a href="#">distinguishedName</a> <a href="#">mSMQJournal</a> <a href="#">mSMQQueueQuota</a> <a href="#">mSMQBasePriority</a> <a href="#">mSMQQueueJournalQuota</a> <a href="#">mSMQLabelEx</a> <a href="#">whenCreated</a> <a href="#">whenChanged</a> <a href="#">mSMQAuthenticate</a> <a href="#">mSMQPrivacyLevel</a> <a href="#">mSMQTransactional</a> <a href="#">mSMQOwnerID</a> <a href="#">nTSecurityDescriptor</a>

Directory Service properties	Active Directory class	Active Directory properties
<a href="#">PROPID_Q_FULL_PATH</a> <a href="#">PROPID_Q_NAME_SUFFIX</a> <a href="#">PROPID_Q_MULTICAST_ADDRESS</a>		<a href="#">distinguishedName</a> <a href="#">mSMQQueueNameExt</a> <a href="#">MSMQ-MulticastAddress</a>
<a href="#">PROPID_QM_MACHINE_ID</a> <a href="#">PROPID_QM_PATHNAME</a> <a href="#">PROPID_QM_SERVICE_ROUTING</a> <a href="#">PROPID_QM_SERVICE_DEPCLIENTS</a> <a href="#">PROPID_QM_SERVICE_DSSERVER</a> <a href="#">PROPID_QM_QUOTA</a> <a href="#">PROPID_QM_JOURNAL_QUOTA</a> <a href="#">PROPID_QM_MACHINE_TYPE</a> <a href="#">PROPID_QM_CREATE_TIME</a> <a href="#">PROPID_QM_MODIFY_TIME</a> <a href="#">PROPID_QM_FOREIGN</a> <a href="#">PROPID_QM_OS</a> <a href="#">PROPID_QM_SECURITY</a> <a href="#">PROPID_QM_SIGN_PK</a> <a href="#">PROPID_QM_ENCRYPT_PK</a> <a href="#">PROPID_QM_FULL_PATH</a> <a href="#">PROPID_QM_SITE_IDS</a> <a href="#">PROPID_QM_OUTFRS_DN</a> <a href="#">PROPID_QM_INFRS_DN</a> <a href="#">PROPID_QM_SERVICE</a>	mSMQConfiguration (MSMQ-Configuration)	<a href="#">objectGUID</a> <a href="#">cn</a> <a href="#">mSMQRoutingServices</a> <a href="#">mSMQDependentClientServices</a> <a href="#">mSMQDsServices</a> <a href="#">mSMQQuota</a> <a href="#">mSMQJournalQuota</a> <a href="#">mSMQComputerTypeEx</a> <a href="#">whenCreated</a> <a href="#">whenChanged</a> <a href="#">mSMQForeign</a> <a href="#">mSMQOSType</a> <a href="#">nTSecurityDescriptor</a> <a href="#">mSMQSignKey</a> <a href="#">mSMQEncryptKey</a> <a href="#">distinguishedName</a> <a href="#">mSMQSites</a> <a href="#">mSMQOutRoutingServers</a> <a href="#">mSMQInRoutingServers</a> <a href="#">mSMQServiceType</a>
<a href="#">PROPID_E_NAMESTYLE</a> <a href="#">PROPID_E_CSP_NAME</a> <a href="#">PROPID_E_ID</a> <a href="#">PROPID_E_LONG_LIVE</a> <a href="#">PROPID_E_SECURITY</a> <a href="#">PROPID_E_VERSION</a>	mSMQEnterpriseSettings (MSMQ-Enterprise-Settings)	<a href="#">mSMQNameStyle</a> <a href="#">mSMQCSPName</a> <a href="#">objectGUID</a> <a href="#">mSMQLongLived</a> <a href="#">nTSecurityDescriptor</a> <a href="#">mSMQVersion</a>
<a href="#">PROPID_L_NEIGHBOR1</a> <a href="#">PROPID_L_NEIGHBOR2</a> <a href="#">PROPID_L_COST</a> <a href="#">PROPID_L_ID</a> <a href="#">PROPID_L_GATES</a> <a href="#">PROPID_L_DESCRIPTION</a> <a href="#">PROPID_L_FULL_PATH</a>	mSMQSiteLink (MSMQ-Site-Link)	<a href="#">mSMQSite1</a> <a href="#">mSMQSite2</a> <a href="#">mSMQCost</a> <a href="#">objectGUID</a> <a href="#">mSMQSiteGates</a> <a href="#">description</a> <a href="#">distinguishedName</a>
<a href="#">PROPID_U_SID</a> <a href="#">PROPID_U_SIGN_CERT</a> <a href="#">PROPID_U_DIGEST</a> <a href="#">PROPID_U_ID</a>	user (User)	<a href="#">objectSid</a> <a href="#">mSMQSignCertificates</a> <a href="#">mSMQDigests</a> <a href="#">objectGUID</a>
<a href="#">PROPID_MQU_SID</a> <a href="#">PROPID_MQU_SIGN_CERT</a>	mSMQMigratedUser (MSMQ-Migrated-User)	<a href="#">mSMQUserSid</a> <a href="#">mSMQSignCertificates</a>

Directory Service properties	Active Directory class	Active Directory properties
<a href="#">PROPID_MQU_DIGEST</a> <a href="#">PROPID_MQU_ID</a> <a href="#">PROPID_MQU_SECURITY</a>		<a href="#">mSMQDigests</a> <a href="#">objectGUID</a> <a href="#">nTSecurityDescriptor</a>
<a href="#">PROPID_S_PATHNAME</a> <a href="#">PROPID_S_SITEID</a> <a href="#">PROPID_S_SECURITY</a> <a href="#">PROPID_S_FULL_NAME</a> <a href="#">PROPID_S_NT4_STUB</a> <a href="#">PROPID_S_FOREIGN</a> <a href="#">PROPID_S_INTERVAL1</a> <a href="#">PROPID_S_INTERVAL2</a>	site (Site)	<a href="#">cn</a> <a href="#">objectGUID</a> <a href="#">nTSecurityDescriptor</a> <a href="#">distinguishedName</a> <a href="#">mSMQNT4Stub</a> <a href="#">mSMQSiteForeign</a> <a href="#">mSMQInterval1</a> <a href="#">mSMQInterval2</a>
<a href="#">PROPID_SET_NAME</a> <a href="#">PROPID_SET_OLDSERVICE</a> <a href="#">PROPID_SET_SERVICE_ROUTING</a> <a href="#">PROPID_SET_SERVICE_DSSERVER</a> <a href="#">PROPID_SET_SERVICE_DEPCLIENTS</a> <a href="#">PROPID_SET_QM_ID</a> <a href="#">PROPID_SET_FULL_PATH</a> <a href="#">PROPID_SET_NT4</a> <a href="#">PROPID_SET_MASTERID</a> <a href="#">PROPID_SET_SITENAME</a>	mSMQSettings (MSMQ-Settings)	<a href="#">cn</a> <a href="#">mSMQServices</a> <a href="#">mSMQRoutingService</a> <a href="#">mSMQDsService</a> <a href="#">mSMQDependentClientService</a> <a href="#">mSMQQMID</a> <a href="#">distinguishedName</a> <a href="#">mSMQNT4Flags</a> <a href="#">mSMQOwnerID</a> <a href="#">mSMQSiteNameEx</a>
<a href="#">PROPID_COM_FULL_PATH</a> <a href="#">PROPID_COM_SAM_ACCOUNT</a> <a href="#">PROPID_COM_ACCOUNT_CONTROL</a> <a href="#">PROPID_COM_DNS_HOSTNAME</a> <a href="#">PROPID_COM_SID</a> <a href="#">PROPID_COM_SIGN_CERT</a> <a href="#">PROPID_COM_DIGEST</a> <a href="#">PROPID_COM_ID</a>	computer (Computer)	<a href="#">distinguishedName</a> <a href="#">sAMAccountName</a> <a href="#">userAccountControl</a> <a href="#">dNSHostName</a> <a href="#">objectSid</a> <a href="#">mSMQSignCertificates</a> <a href="#">mSMQDigests</a> <a href="#">objectGUID</a>

### 2.6.1 Distribution Lists and Alias Classes

MSMQ uses two additional Active Directory schema classes—"msMQ-group" and "msMQ-Custom-Recipient"—to contain attributes of distribution lists and queue aliases. These schema classes are always accessed directly from Active Directory and consequently have no associated **PROPID**s.

Active Directory Class	Attribute	Description
msMQ-Group	<a href="#">objectGUID</a> <a href="#">member</a>	Contains the GUID identifier of the distribution list. A multi-valued attribute that contains a list of distinguished names of queue and/or group objects.
msMQ-Custom-Recipient	<a href="#">msMQ-Recipient-FormatName</a>	Contains the format name of a queue. Contains a description of the alias.

Active Directory Class	Attribute	Description
	<a href="#">description</a>	

**Note** The use of these two Active Directory classes was introduced in MSMQ 3.0.

## 2.7 Error Codes

The following table specifies MSMQ-specific [HRESULT](#) values. Not all methods of all protocols return these error codes. Common HRESULT values are specified in [\[MS-ERREF\]](#) section 2. [<5>](#)

Value	Message text
MQ_INFORMATION_PROPERTY 0x400E0001	One or more of the properties passed resulted in a warning, but the function completed.
MQ_INFORMATION_ILLEGAL_PROPERTY 0x400E0002	The property ID is invalid.
MQ_INFORMATION_PROPERTY_IGNORED 0x400E0003	The property specified was ignored for this operation.
MQ_INFORMATION_UNSUPPORTED_PROPERTY 0x400E0004	The property specified is not supported and was ignored for this operation.
MQ_INFORMATION_DUPLICATE_PROPERTY 0x400E0005	The property specified is already in the property identifier array. The duplicate was ignored for this operation.
MQ_INFORMATION_OPERATION_PENDING 0x400E0006	An asynchronous operation is currently pending.
MQ_INFORMATION_FORMATNAME_BUFFER_TOO_SMALL 0x400E0009	The format name buffer supplied was too small to hold the format name; however, the queue was created successfully.
MQ_INFORMATION_INTERNAL_USER_CERT_EXIST 0x400E000A	An internal Message Queuing certificate already exists for this user.
MQ_INFORMATION_OWNER_IGNORED 0x400E000B	The queue owner was not set during the processing of this call.
MQ_ERROR 0xC00E0001	Generic error code.
MQ_ERROR_PROPERTY 0xC00E0002	One or more of the properties passed are invalid.
MQ_ERROR_QUEUE_NOT_FOUND 0xC00E0003	The queue does not exist or you do not have sufficient permissions to perform the operation.

Value	Message text
MQ_ERROR_QUEUE_NOT_ACTIVE 0xC00E0004	The queue is not open or may not exist.
MQ_ERROR_QUEUE_EXISTS 0xC00E0005	A queue with the same path name already exists.
MQ_ERROR_INVALID_PARAMETER 0xC00E0006	An invalid parameter was passed to a function.
MQ_ERROR_INVALID_HANDLE 0xC00E0007	An invalid handle was passed to a function.
MQ_ERROR_OPERATION_CANCELED 0xC00E0008	The operation was canceled before it could be completed.
MQ_ERROR_SHARING_VIOLATION 0xC00E0009	There is a sharing violation. The queue is already open for exclusive retrieval.
MQ_ERROR_SERVICE_NOT_AVAILABLE 0xC00E000B	The Message Queuing service is not available.
MQ_ERROR_MACHINE_NOT_FOUND 0xC00E000D	The computer specified cannot be found.
MQ_ERROR_ILLEGAL_SORT 0xC00E0010	The sort operation specified is invalid; for example, there are duplicate columns.
MQ_ERROR_ILLEGAL_USER 0xC00E0011	The user specified is not a valid user.
MQ_ERROR_NO_DS 0xC00E0013	A connection with Active Directory cannot be established. Verify that there are sufficient permissions to perform this operation.
MQ_ERROR_ILLEGAL_QUEUE_PATHNAME 0xC00E0014	The queue path name specified is invalid.
MQ_ERROR_ILLEGAL_PROPERTY_VALUE 0xC00E0018	The property value specified is invalid.
MQ_ERROR_ILLEGAL_PROPERTY_VT 0xC00E0019	The VARTYPE value specified is invalid.
MQ_ERROR_BUFFER_OVERFLOW 0xC00E001A	The buffer supplied for <b>message property</b> retrieval is too small. The message was not removed from the queue, but the part of the message property that was in the buffer was copied.
MQ_ERROR_IO_TIMEOUT	The time specified to wait for the



Value	Message text
0xC00E001B	message elapsed.
MQ_ERROR_ILLEGAL_CURSOR_ACTION 0xC00E001C	The MQ_ACTION_PEEK_NEXT value specified cannot be used with the current <b>cursor</b> position.
MQ_ERROR_MESSAGE_ALREADY_RECEIVED 0xC00E001D	The message to which the cursor is currently pointing was removed from the queue by another process or by another call without the use of this cursor.
MQ_ERROR_ILLEGAL_FORMATNAME 0xC00E001E	The format name specified is invalid.
MQ_ERROR_FORMATNAME_BUFFER_TOO_SMALL 0xC00E001F	The format name buffer supplied to the API was too small to hold the format name.
MQ_ERROR_UNSUPPORTED_FORMATNAME_OPERATION 0xC00E0020	Operations of the type requested (for example, deleting a queue using a <a href="#">direct format name</a> ) are not supported for the format name specified.
MQ_ERROR_ILLEGAL_SECURITY_DESCRIPTOR 0xC00E0021	The specified security descriptor is invalid.
MQ_ERROR_SENDERID_BUFFER_TOO_SMALL 0xC00E0022	The size of the buffer for the user ID property is too small.
MQ_ERROR_SECURITY_DESCRIPTOR_TOO_SMALL 0xC00E0023	The size of the buffer passed is too small.
MQ_ERROR_CANNOT_IMPERSONATE_CLIENT 0xC00E0024	The security credentials cannot be verified because the <b>remote procedure call (RPC)</b> server cannot impersonate the client application.
MQ_ERROR_ACCESS_DENIED 0xC00E0025	Access is denied.
MQ_ERROR_PRIVILEGE_NOT_HELD 0xC00E0026	The client does not have sufficient security privileges to perform the operation.
MQ_ERROR_INSUFFICIENT_RESOURCES 0xC00E0027	There are insufficient resources to perform this operation.
MQ_ERROR_USER_BUFFER_TOO_SMALL 0xC00E0028	The request failed because the user buffer is too small to hold the information returned.
MQ_ERROR_MESSAGE_STORAGE_FAILED 0xC00E002A	A recoverable or journal message could not be stored. The message

Value	Message text
	was not sent.
MQ_ERROR_SENDER_CERT_BUFFER_TOO_SMALL 0xC00E002B	The buffer for the user certificate property is too small.
MQ_ERROR_INVALID_CERTIFICATE 0xC00E002C	The user certificate is invalid.
MQ_ERROR_CORRUPTED_INTERNAL_CERTIFICATE 0xC00E002D	The internal Message Queuing certificate is corrupted.
MQ_ERROR_INTERNAL_USER_CERT_EXIST 0xC00E002E	An internal Message Queuing certificate already exists for this user.
MQ_ERROR_NO_INTERNAL_USER_CERT 0xC00E002F	No internal Message Queuing certificate exists for the user.
MQ_ERROR_CORRUPTED_SECURITY_DATA 0xC00E0030	A cryptographic function failed.
MQ_ERROR_CORRUPTED_PERSONAL_CERT_STORE 0xC00E0031	The personal certificate store is corrupted.
MQ_ERROR_COMPUTER_DOES_NOT_SUPPORT_ENCRYPTION 0xC00E0033	The computer does not support encryption operations.
MQ_ERROR_BAD_SECURITY_CONTEXT 0xC00E0035	The security context is invalid.
MQ_ERROR_COULD_NOT_GET_USER_SID 0xC00E0036	The SID cannot be obtained from the thread token.
MQ_ERROR_COULD_NOT_GET_ACCOUNT_INFO 0xC00E0037	The account information for the user cannot be obtained.
MQ_ERROR_ILLEGAL_MQCOLUMNS 0xC00E0038	The MQCOLUMNS parameter is invalid.
MQ_ERROR_ILLEGAL_PROPID 0xC00E0039	A property identifier is invalid.
MQ_ERROR_ILLEGAL_RELATION 0xC00E003A	A relationship parameter is invalid.
MQ_ERROR_ILLEGAL_PROPERTY_SIZE 0xC00E003B	The size of the buffer for the message identifier or correlation identifier is invalid.
MQ_ERROR_ILLEGAL_RESTRICTION_PROPID 0xC00E003C	A property identifier specified in MQRESTRICTION is invalid.
MQ_ERROR_ILLEGAL_MQQUEUEPROPS 0xC00E003D	Either the pointer to the MQQUEUEPROPS structure has a

Value	Message text
	null value, or no properties are specified in it.
MQ_ERROR_PROPERTY_NOTALLOWED 0xC00E003E	The property identifier specified is invalid for the operation requested.
MQ_ERROR_INSUFFICIENT_PROPERTIES 0xC00E003F	Not all the properties required for the operation were specified for the input parameters.
MQ_ERROR_MACHINE_EXISTS 0xC00E0040	The MSMQ Configuration (msmq) object already exists in Active Directory.
MQ_ERROR_ILLEGAL_MQMPROPS 0xC00E0041	Either the pointer to the MQMPROPS structure has a null value, or no properties are specified in it.
MQ_ERROR_DS_IS_FULL 0xC00E0042	Valid for MSMQ 1.0 and MSMQ 2.0. distinguished name (DS) is full.
MQ_ERROR_DS_ERROR 0xC00E0043	There is an internal Active Directory error.
MQ_ERROR_INVALID_OWNER 0xC00E0044	The object owner is invalid.
MQ_ERROR_UNSUPPORTED_ACCESS_MODE 0xC00E0045	The access mode specified is unsupported.
MQ_ERROR_RESULT_BUFFER_TOO_SMALL 0xC00E0046	The result buffer specified is too small.
MQ_ERROR_DELETE_CN_IN_USE 0xC00E0048	Valid for MSMQ 1.0 and MSMQ 2.0. The connected network cannot be deleted; it is in use.
MQ_ERROR_NO_RESPONSE_FROM_OBJECT_SERVER 0xC00E0049	There was no response from the object owner.
MQ_ERROR_OBJECT_SERVER_NOT_AVAILABLE 0xC00E004A	The object owner is not available.
MQ_ERROR_QUEUE_NOT_AVAILABLE 0xC00E004B	An error occurred while reading from a queue located on a remote computer.
MQ_ERROR_DTC_CONNECT 0xC00E004C	A connection cannot be established with the Distributed Transaction Coordinator.
MQ_ERROR_TRANSACTION_IMPORT	The transaction specified cannot

Value	Message text
0xC00E004E	be imported.
MQ_ERROR_TRANSACTION_USAGE 0xC00E0050	An attempted action cannot be performed within a transaction.
MQ_ERROR_TRANSACTION_SEQUENCE 0xC00E0051	The transaction's operation sequence is incorrect.
MQ_ERROR_MISSING_CONNECTOR_TYPE 0xC00E0055	The connector type message property is not specified. This property is required for sending an acknowledgment message or a secure message.
MQ_ERROR_STALE_HANDLE 0xC00E0056	The Message Queuing service was restarted. Any open queue handles should be closed.
MQ_ERROR_TRANSACTION_ENLIST 0xC00E0058	The transaction specified cannot be enlisted.
MQ_ERROR_QUEUE_DELETED 0xC00E005A	The queue was deleted. Message cannot be received anymore using this queue handle. The handle should be closed.
MQ_ERROR_ILLEGAL_CONTEXT 0xC00E005B	The context parameter is invalid.
MQ_ERROR_ILLEGAL_SORT_PROPID 0xC00E005C	An invalid property identifier is specified in MQSORTSET.
MQ_ERROR_LABEL_TOO_LONG 0xC00E005D	The message label is too long. Its length should be less than or equal to MQ_MAX_MSG_LABEL_LEN.
MQ_ERROR_LABEL_BUFFER_TOO_SMALL 0xC00E005E	The label buffer supplied to the API is too small.
MQ_ERROR_MQIS_SERVER_EMPTY 0xC00E005F	Valid for MSMQ 1.0 and MSMQ 2.0. The list of MQIS servers (in the registry) is empty.
MQ_ERROR_MQIS_READONLY_MODE 0xC00E0060	Valid for MSMQ 1.0 and MSMQ 2.0: The MQIS database is in read-only mode.
MQ_ERROR_SYMM_KEY_BUFFER_TOO_SMALL 0xC00E0061	The buffer passed for the symmetric key is too small.
MQ_ERROR_SIGNATURE_BUFFER_TOO_SMALL 0xC00E0062	The buffer passed for the signature property is too small.

Value	Message text
MQ_ERROR_PROV_NAME_BUFFER_TOO_SMALL 0xC00E0063	The buffer passed for the provider name property is too small.
MQ_ERROR_ILLEGAL_OPERATION 0xC00E0064	The operation is invalid for a foreign Message Queuing system.
MQ_ERROR_WRITE_NOT_ALLOWED 0xC00E0065	Another MQIS server is being installed. Write operations to the database are not allowed at this stage.
MQ_ERROR_WKS_CANT_SERVE_CLIENT 0xC00E0066	<b>Independent clients</b> cannot support dependent clients. A Message Queuing server is required.
MQ_ERROR_DEPEND_WKS_LICENSE_OVERFLOW 0xC00E0067	The number of dependent clients served by the Message Queuing server has reached its limit.
MQ_CORRUPTED_QUEUE_WAS_DELETED 0xC00E0068	The corresponding file for the designated queue in the Lqs folder was deleted because it was corrupted.
MQ_ERROR_REMOTE_MACHINE_NOT_AVAILABLE 0xC00E0069	The remote computer is not available.
MQ_ERROR_UNSUPPORTED_OPERATION 0xC00E006A	This operation is not supported for Message Queuing installed in workgroup mode.
MQ_ERROR_ENCRYPTION_PROVIDER_NOT_SUPPORTED 0xC00E006B	The requested cryptographic service provider is not supported by Message Queuing.
MQ_ERROR_CANNOT_SET_CRYPTO_SEC_DESCR 0xC00E006C	The security descriptors for the cryptographic keys cannot be set.
MQ_ERROR_CERTIFICATE_NOT_PROVIDED 0xC00E006D	A user attempted to send an authenticated message without a certificate.
MQ_ERROR_Q_DNS_PROPERTY_NOT_SUPPORTED 0xC00E006E	The column PROPID_Q_PATHNAME_DNS is not supported for the API.
MQ_ERROR_CANNOT_CREATE_CERT_STORE 0xC00E006F	A certificate store cannot be created for the internal certificate.
MQ_ERROR_CANNOT_OPEN_CERT_STORE 0xC00E0070	The certificate store for the internal certificate cannot be opened.
MQ_ERROR_ILLEGAL_ENTERPRISE_OPERATION 0xC00E0071	This operation is invalid for an MsmqServices object.

Value	Message text
MQ_ERROR_CANNOT_GRANT_ADD_GUID 0xC00E0072	The Add GUID permission cannot be granted to the current user.
MQ_ERROR_CANNOT_LOAD MSMQOCM 0xC00E0073	Valid for MSMQ 1.0 and MSMQ 2.0. The dynamic-link library Msmqocm.dll cannot be loaded.
MQ_ERROR_NO_ENTRY_POINT MSMQOCM 0xC00E0074	An entry point cannot be located in Msmqocm.dll.
MQ_ERROR_NO_MSMQ_SERVERS_ON_DC 0xC00E0075	Message Queuing servers cannot be found on domain controllers.
MQ_ERROR_CANNOT_JOIN_DOMAIN 0xC00E0076	The computer joined the domain, but Message Queuing will continue to run in workgroup mode because it failed to register itself in Active Directory.
MQ_ERROR_CANNOT_CREATE_ON_GC 0xC00E0077	The object was not created on the Global Catalog server specified.
MQ_ERROR_GUID_NOT_MATCHING 0xC00E0078	Valid for MSMQ 1.0 and MSMQ 2.0. Failed to create an msmqConfiguration object with a GUID that matches the computer installation. MSMQ must be uninstalled and then reinstalled.
MQ_ERROR_PUBLIC_KEY_NOT_FOUND 0xC00E0079	The public key for the designated computer cannot be found.
MQ_ERROR_PUBLIC_KEY_DOES_NOT_EXIST 0xC00E007A	The public key for the designated computer does not exist.
MQ_ERROR_ILLEGAL_MQPRIVATEPROPS 0xC00E007B	The parameters in MQPRIVATEPROPS are invalid. Either the pointer to the MQPRIVATEPROPS structure has a null value, or no properties are specified in it.
MQ_ERROR_NO_GC_IN_DOMAIN 0xC00E007C	Global Catalog servers cannot be found in the domain specified.
MQ_ERROR_NO_MSMQ_SERVERS_ON_GC 0xC00E007D	No Message Queuing servers were found on Global Catalog servers.
MQ_ERROR_CANNOT_GET_DN 0xC00E007E	Valid for MSMQ 1.0 and MSMQ 2.0. Failed to retrieve the distinguished name of the local computer.

Value	Message text
MQ_ERROR_CANNOT_HASH_DATA_EX 0xC00E007F	Data for an authenticated message cannot be hashed.
MQ_ERROR_CANNOT_SIGN_DATA_EX 0xC00E0080	Data cannot be signed before sending an authenticated message.
MQ_ERROR_CANNOT_CREATE_HASH_EX 0xC00E0081	A hash object cannot be created for an authenticated message.
MQ_ERROR_FAIL_VERIFY_SIGNATURE_EX 0xC00E0082	The signature of the message received is not valid.
MQ_ERROR_CANNOT_DELETE_PSC_OBJECTS 0xC00E0083	The delete operation against the designated object failed because the object is owned by a Primary Site Controller (PSC). The operation cannot be performed.
MQ_ERROR_NO_MQUSER_OU 0xC00E0084	There is no MSMQ Users organizational unit object in Active Directory for the domain. Please create one manually.
MQ_ERROR_CANNOT_LOAD_MQAD 0xC00E0085	The dynamic-link library Mqad.dll cannot be loaded.
MQ_ERROR_CANNOT_LOAD_MQDSSRV 0xC00E0086	Obsolete: not used in any version of MSMQ.
MQ_ERROR_PROPERTIES_CONFLICT 0xC00E0087	Two or more of the properties passed cannot coexist. For example, when sending a message, both PROPID_M_RESP_QUEUE and PROPID_M_RESP_FORMAT_NAME cannot be set.
MQ_ERROR_MESSAGE_NOT_FOUND 0xC00E0088	The message does not exist or was removed from the queue.
MQ_ERROR_CANT_RESOLVE_SITES 0xC00E0089	The sites in which the computer resides cannot be resolved. Verify that the subnets in the network are configured correctly in Active Directory and that each site is configured with the appropriate subnet.
MQ_ERROR_NOT_SUPPORTED_BY_DEPENDENT_CLIENTS 0xC00E008A	This operation is not supported by dependent clients.
MQ_ERROR_OPERATION_NOT_SUPPORTED_BY_REMOTE_COMPUTER 0xC00E008B	This operation is not supported by the remote Message Queuing service.

Value	Message text
MQ_ERROR_NOT_A_CORRECT_OBJECT_CLASS 0xC00E008C	The object for which properties were requested from Active Directory does not belong to the class requested.
MQ_ERROR_MULTI_SORT_KEYS 0xC00E008D	The value of cCol in MQSORTSET cannot be greater than 1. Active Directory supports only a single sort key.
MQ_ERROR_GC_NEEDED 0xC00E008E	An MSMQ Configuration (msmq) object with the GUID supplied cannot be created.
MQ_ERROR_DS_BIND_ROOT_FOREST 0xC00E008F	Binding to the forest root failed. This error usually indicates a problem in the Domain Name System (DNS) configuration.
MQ_ERROR_DS_LOCAL_USER 0xC00E0090	A local user is authenticated as an anonymous user and cannot access Active Directory. The local user must log on as a domain user to access Active Directory.
MQ_ERROR_Q_ADS_PROPERTY_NOT_SUPPORTED 0xC00E0091	The column PROPID_Q_ADS_PATH is not supported for the API.
MQ_ERROR_BAD_XML_FORMAT 0xC00E0092	The given property is not a valid XML document.
MQ_ERROR_UNSUPPORTED_CLASS 0xC00E0093	The Active Directory object specified is not an instance of a supported class.
MQ_ERROR_UNINITIALIZED_OBJECT 0xC00E0094	The MSMQManagement object must be initialized before it is used.
MQ_ERROR_CANNOT_CREATE_PSC_OBJECTS 0xC00E0095	The create object operation cannot be performed because the object must be owned by a Primary Site Controller (PSC).
MQ_ERROR_CANNOT_UPDATE_PSC_OBJECTS 0xC00E0096	The update operation cannot be performed because the designated object is owned by a Primary Site Controller (PSC).



### 3 Appendix A: PROPVARIANT IDL

```
import "ms-dtyp.idl";

// forward declaration
typedef struct tag_inner_PROPVARIANT PROPVARIANT;

// basic type aliases
typedef unsigned long    PROPID;
typedef short            VARIANT_BOOL;

typedef struct tagBLOB {
    ULONG cbSize;
    [size_is(cbSize)]
    BYTE *pBlobData;
} BLOB;

typedef struct tagCAUB
{
    unsigned long    cElems;
    [size_is( cElems )]
    unsigned char *  pElems;
} CAUB;

typedef struct tagCAUI
{
    unsigned long    cElems;
    [size_is( cElems )]
    unsigned short * pElems;
} CAUI;

typedef struct tagCAL
{
    unsigned long    cElems;
    [size_is( cElems )]
    long * pElems;
} CAL;

typedef struct tagCAUL
{
    unsigned long    cElems;
    [size_is( cElems )]
    unsigned long *  pElems;
} CAUL;

typedef struct tagCAUH
{
    unsigned long    cElems;
    [size_is( cElems )]
    ULARGE_INTEGER * pElems;
} CAUH;

typedef struct tagCACLSID
{
    unsigned long    cElems;
```

```

        [size_is( cElems )]
        GUID * pElems;
} CACLSID;

typedef struct tagCALPWSTR
{
    unsigned long cElems;
    [size_is( cElems )]
    LPWSTR * pElems;
} CALPWSTR;

typedef struct tagCAPROPVARIANT
{
    unsigned long cElems;
    [size_is( cElems )]
    PROPVARIANT * pElems;
} CAPROPVARIANT;

enum VARENUM
{
    VT_EMPTY            = 0,
    VT_NULL             = 1,
    VT_I2               = 2,
    VT_I4               = 3,
    VT_BOOL             = 11,
    VT_VARIANT          = 12,
    VT_I1               = 16,
    VT_UI1              = 17,
    VT_UI2              = 18,
    VT_UI4              = 19,
    VT_I8               = 20,
    VT_UI8              = 21,
    VT_LPWSTR           = 31,
    VT_BLOB              = 65,
    VT_CLSID            = 72,

    VT_VECTOR           = 0x1000,
} ;

typedef unsigned short VARTYPE;
typedef BYTE          PROPVAR_PAD1;
typedef BYTE          PROPVAR_PAD2;
typedef ULONG         PROPVAR_PAD3;

struct tag_inner_PROPVARIANT
{
    VARTYPE vt;
    PROPVAR_PAD1 wReserved1;
    PROPVAR_PAD2 wReserved2;
    PROPVAR_PAD3 wReserved3;
    [switch_is((unsigned short) vt)] union
    {
        [case (VT_UI1)]          UCHAR          bVal;
        [case (VT_I2)]          SHORT           iVal;
        [case (VT_UI2)]         USHORT          uiVal;
        [case (VT_I4)]          LONG            lVal;
        [case (VT_UI4)]         ULONG           ulVal;
        [case (VT_BOOL)]        VARIANT_BOOL    boolVal;
    }

```

```

[case (VT_CLSID)]          GUID *      puuid;
[case (VT_BLOB)]          BLOB        blob;
[case (VT_LPWSTR)]        LPWSTR      pwszVal;
[case (VT_VECTOR|VT_UI1)] CAUB        caub;
[case (VT_VECTOR|VT_UI2)] CAUI        caui;
[case (VT_VECTOR|VT_I4)]  CAL        cal;
[case (VT_VECTOR|VT_UI4)] CAUL        caul;
[case (VT_VECTOR|VT_UI8)] CAUH        cauh;
[case (VT_VECTOR|VT_CLSID)] CACLSID    cauid;
[case (VT_VECTOR|VT_LPWSTR)] CALPWSTR  calpwstr;
[case (VT_VECTOR|VT_VARIANT)] CAPROPVARIANT capropvar;
} _varUnion;
};

```

## 4 Appendix B: Windows Behavior

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

- Windows Vista and Windows Server 2008
- Windows Server 2003
- Windows XP
- Windows 2000
- Windows NT

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 2.3.1.5:](#) When the property is not set, the storage size of a queue is limited only by the available disk space on the local computer or the computer quota. For Windows XP Professional, there is no default computer quota. For the Windows Server 2003 family, the default computer quota is 8 GB. For Windows 2000, the default computer quota is 2 GB.

[<2> Section 2.3.1.7:](#) When the property is not set, the storage size of a queue is limited only by the available disk space on the local computer or the computer quota. For Windows XP Professional, there is no default computer quota. For the Windows Server 2003 family, the default computer quota is 8 GB. For Windows 2000, the default computer quota is 2 GB.

[<3> Section 2.3.2.11:](#) Windows Server 2003, Windows Vista, and Windows Server 2008 accept an empty string.

[<4> Section 2.3.2.11:](#) Windows NT, Windows 95, Windows 98, and Windows 2000 require the following ABNF format:

```
MachineType = WindowsLiteral Space OSType Space UInt "."
              UInt Space "(" BuildLiteral Space UInt "," Space Platform ")"
              Space "-" Space MSMQLiteral Space UInt "." UInt Space
              "(" BuildLiteral Space UInt ")"
WindowsLiteral = %x57 %x69 %x6e %x64 %x6f %x77 %x73
BuildLiteral = %x42 %x75 %x69 %x6c %x64
MSMQLiteral = %x4d %x53 %x4d %x51
OSType = *(%x20-7E)
Platform = *(%x20-7E)
UInt = *(%x30-39)
Space = %x20
```

[<5> Section 2.7:](#) For more information about MSMQ HRESULTs, see [\[MSDN-MQEIC\]](#).

## 5 Index

### A

[Active Directory properties](#)  
[Alias classes](#)  
[Applicability](#)

### B

[BLOB structure](#)

### C

[CACLSID structure](#)  
[CAL structure](#)  
[CALPWSTR structure](#)  
[CAPROPVARIANT structure](#)  
[CAUB structure](#)  
[CAUH structure](#)  
[CAUI structure](#)  
[CAUL structure](#)  
[Computer property identifiers](#)  
[Connected network property identifiers](#)  
[Connector format names](#)  
[COUNTEDARRAY](#)

### D

[Definitions](#)  
[Direct format names](#)  
[Distribution list format names](#)  
[Distribution lists](#)  
[DL ID structure](#)

### E

[Enterprise object property identifiers](#)  
[Error codes](#)

### F

[Fields - vendor-extensible](#)  
[Full IDL](#)

### G

[Glossary](#)

### I

[IDL](#)  
[Informative references](#)  
[Introduction](#)  
[IP Address packet](#)  
[IPX Address packet](#)

### L

[Localization](#)

### M

[Machine format names](#)  
[Machine property identifiers](#)  
[Management machine property identifiers](#)  
[Management queue property identifiers](#)  
[MODSPUBLICKEY packet](#)  
[MODSPUBLICKEYS packet](#)  
[MQUser property identifiers](#)  
[mSMQDigestsMig](#)  
[mSMQMigrated](#)  
[msMQ-Recipient-FormatName](#)  
[mSMQSignCertificatesMig](#)  
[mSMQSiteGatesMig](#)  
[Multicast format names](#)  
[MULTICAST ID structure](#)  
[Multiple-element format names](#)

### N

[Normative references](#)

### O

[OBJECTID structure](#)

### P

[Path names](#)  
[Private format names](#)  
[PROPID\\_CN\\_SECURITY](#)  
[PROPID to AD Schema mapping](#)  
[PROPID\\_MQU\\_ID](#)  
[PROPID\\_COM\\_ACCOUNT\\_CONTROL](#)  
[PROPID\\_COM\\_DIGEST](#)  
[PROPID\\_COM\\_DNS\\_HOSTNAME](#)  
[PROPID\\_COM\\_FULL\\_PATH](#)  
[PROPID\\_COM\\_ID](#)  
[PROPID\\_COM\\_SAM\\_ACCOUNT](#)  
[PROPID\\_COM\\_SID](#)  
[PROPID\\_COM\\_SIGN\\_CERT](#)  
[PROPID\\_E\\_ID](#)  
[PROPID\\_E\\_CSP\\_NAME](#)  
[PROPID\\_E\\_LONG\\_LIVE](#)  
[PROPID\\_E\\_NAMESTYLE](#)  
[PROPID\\_E\\_SECURITY](#)  
[PROPID\\_E\\_VERSION](#)  
[PROPID\\_L\\_ACTUAL\\_COST](#)  
[PROPID\\_L\\_COST](#)  
[PROPID\\_L\\_DESCRIPTION](#)  
[PROPID\\_L\\_FULL\\_PATH](#)  
[PROPID\\_L\\_GATES](#)  
[PROPID\\_L\\_GATES\\_DN](#)  
[PROPID\\_L\\_ID](#)  
[PROPID\\_L\\_NEIGHBOR1](#)  
[PROPID\\_L\\_NEIGHBOR1\\_DN](#)  
[PROPID\\_L\\_NEIGHBOR2](#)  
[PROPID\\_L\\_NEIGHBOR2\\_DN](#)

[PROPID\\_MGMT\\_MSMQ\\_ACTIVEQUEUES](#)  
[PROPID\\_MGMT\\_MSMQ\\_BYTES\\_IN\\_ALL\\_QUEUES](#)  
[PROPID\\_MGMT\\_MSMQ\\_CONNECTED](#)  
[PROPID\\_MGMT\\_MSMQ\\_DSSERVER](#)  
[PROPID\\_MGMT\\_MSMQ\\_PRIVATEQ](#)  
[PROPID\\_MGMT\\_MSMQ\\_TYPE](#)  
[PROPID\\_MGMT\\_QUEUE\\_SUBQUEUE\\_COUNT](#)  
[PROPID\\_MGMT\\_QUEUE\\_SUBQUEUE\\_NAMES](#)  
[PROPID\\_MGMT\\_QUEUE\\_BYTES\\_IN\\_JOURNAL](#)  
[PROPID\\_MGMT\\_QUEUE\\_BYTES\\_IN\\_QUEUE](#)  
[PROPID\\_MGMT\\_QUEUE\\_CONNECTION\\_HISTORY](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_FIRST\\_NON\\_ACK](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_LAST\\_ACK](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_LAST\\_ACK\\_COUNT](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_LAST\\_ACK\\_TIME](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_LAST\\_NON\\_ACK](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_NEXT\\_SEQ](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_NO\\_ACK\\_COUNT](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_NO\\_READ\\_COUNT](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_RESEND\\_COUNT](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_RESEND\\_INTERVAL](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_RESEND\\_TIME](#)  
[PROPID\\_MGMT\\_QUEUE\\_EOD\\_SOURCE\\_INFO](#)  
[PROPID\\_MGMT\\_QUEUE\\_FOREIGN](#)  
[PROPID\\_MGMT\\_QUEUE\\_FORMATNAME](#)  
[PROPID\\_MGMT\\_QUEUE\\_JOURNAL\\_MESSAGE\\_COUNT](#)  
[PROPID\\_MGMT\\_QUEUE\\_LOCATION](#)  
[PROPID\\_MGMT\\_QUEUE\\_MESSAGE\\_COUNT](#)  
[PROPID\\_MGMT\\_QUEUE\\_NEXTHOPS](#)  
[PROPID\\_MGMT\\_QUEUE\\_PATHNAME](#)  
[PROPID\\_MGMT\\_QUEUE\\_STATE](#)  
[PROPID\\_MGMT\\_QUEUE\\_TYPE](#)  
[PROPID\\_MGMT\\_QUEUE\\_XACT](#)  
[PROPID\\_MQU\\_DIGEST](#)  
[PROPID\\_MQU\\_SECURITY](#)  
[PROPID\\_MQU\\_SID](#)  
[PROPID\\_MQU\\_SIGN\\_CERT](#)  
[PROPID\\_Q\\_ADS\\_PATH](#)  
[PROPID\\_Q\\_AUTHENTICATE](#)  
[PROPID\\_Q\\_BASEPRIORITY](#)  
[PROPID\\_Q\\_CREATE\\_TIME](#)  
[PROPID\\_Q\\_FULL\\_PATH](#)  
[PROPID\\_Q\\_INSTANCE](#)  
[PROPID\\_Q\\_JOURNAL](#)  
[PROPID\\_Q\\_JOURNAL\\_QUOTA](#)  
[PROPID\\_Q\\_LABEL](#)  
[PROPID\\_Q\\_MASTERID](#)  
[PROPID\\_Q\\_MODIFY\\_TIME](#)  
[PROPID\\_Q\\_MULTICAST\\_ADDRESS](#)  
[PROPID\\_Q\\_NAME\\_SUFFIX](#)  
[PROPID\\_Q\\_OBJ\\_SECURITY](#)  
[PROPID\\_Q\\_PATHNAME](#)  
[PROPID\\_Q\\_PATHNAME\\_DNS](#)  
[PROPID\\_Q\\_PRIV\\_LEVEL](#)  
[PROPID\\_Q\\_QMID](#)  
[PROPID\\_Q\\_QUOTA](#)  
[PROPID\\_Q\\_SECURITY](#)  
[PROPID\\_Q\\_SECURITY\\_INFORMATION](#)  
[PROPID\\_Q\\_TRANSACTION](#)  
[PROPID\\_Q\\_TYPE](#)  
[PROPID\\_QM\\_CNS](#)  
[PROPID\\_QM\\_CREATE\\_TIME](#)

[PROPID\\_QM\\_FOREIGN](#)  
[PROPID\\_QM\\_FULL\\_PATH](#)  
[PROPID\\_QM\\_INFRS\\_DN](#)  
[PROPID\\_QM\\_JOURNAL\\_QUOTA](#)  
[PROPID\\_QM\\_MODIFY\\_TIME](#)  
[PROPID\\_QM\\_OS](#)  
[PROPID\\_QM\\_OUTFRS\\_DN](#)  
[PROPID\\_QM\\_QUOTA](#)  
[PROPID\\_QM\\_SERVICE\\_ROUTING](#)  
[PROPID\\_QM\\_SITE\\_IDS](#)  
[PROPID\\_QM\\_ENCRYPT\\_PK](#)  
[PROPID\\_QM\\_ENCRYPT\\_PKS](#)  
[PROPID\\_QM\\_ENCRYPTION\\_PK](#)  
[PROPID\\_QM\\_ENCRYPTION\\_PK\\_BASE](#)  
[PROPID\\_QM\\_ENCRYPTION\\_PK\\_ENHANCED](#)  
[PROPID\\_QM\\_GROUP\\_IN\\_CLUSTER](#)  
[PROPID\\_QM\\_INFRS](#)  
[PROPID\\_QM\\_MACHINE\\_ID](#)  
[PROPID\\_QM\\_MACHINE\\_TYPE](#)  
[PROPID\\_QM\\_OBJ\\_SECURITY](#)  
[PROPID\\_QM\\_OUTFRS](#)  
[PROPID\\_QM\\_OWNER\\_SID](#)  
[PROPID\\_QM\\_PATHNAME](#)  
[PROPID\\_QM\\_PATHNAME\\_DNS](#)  
[PROPID\\_QM\\_SECURITY](#)  
[PROPID\\_QM\\_SECURITY\\_INFORMATION](#)  
[PROPID\\_QM\\_SERVICE](#)  
[PROPID\\_QM\\_SERVICE\\_DEPCLIENTS](#)  
[PROPID\\_QM\\_SERVICE\\_DSSERVER](#)  
[PROPID\\_QM\\_SIGN\\_PK](#)  
[PROPID\\_QM\\_SIGN\\_PKS](#)  
[PROPID\\_QM\\_SITE\\_ID](#)  
[PROPID\\_QM\\_UPGRADE\\_DACL](#)  
[PROPID\\_S\\_SECURITY\\_INFORMATION](#)  
[PROPID\\_S\\_FOREIGN](#)  
[PROPID\\_S\\_FULL\\_NAME](#)  
[PROPID\\_S\\_GATES](#)  
[PROPID\\_S\\_INTERVAL1](#)  
[PROPID\\_S\\_INTERVAL2](#)  
[PROPID\\_S\\_NT4\\_STUB](#)  
[PROPID\\_S\\_PATHNAME](#)  
[PROPID\\_S\\_PSC](#)  
[PROPID\\_S\\_PSC\\_SIGNPK](#)  
[PROPID\\_S\\_SECURITY](#)  
[PROPID\\_S\\_SITEID](#)  
[PROPID\\_SET\\_FULL\\_PATH](#)  
[PROPID\\_SET\\_MASTERID](#)  
[PROPID\\_SET\\_NAME](#)  
[PROPID\\_SET\\_NT4](#)  
[PROPID\\_SET\\_OLDSERVICE](#)  
[PROPID\\_SET\\_QM\\_ID](#)  
[PROPID\\_SET\\_SERVICE](#)  
[PROPID\\_SET\\_SERVICE\\_DEPCLIENTS](#)  
[PROPID\\_SET\\_SERVICE\\_DSSERVER](#)  
[PROPID\\_SET\\_SERVICE\\_ROUTING](#)  
[PROPID\\_SET\\_SITENAME](#)  
[PROPID\\_U\\_DIGEST](#)  
[PROPID\\_U\\_ID](#)  
[PROPID\\_U\\_SID](#)  
[PROPID\\_U\\_SIGN\\_CERT](#)  
[PROP VARIANT](#)  
[PTA\\_ADDRESS](#)

[Public format names](#)

## Q

[Queue names](#)

[Queue property identifiers](#)

[QUEUE\\_FORMAT structure](#)

[QUEUE\\_FORMAT\\_TYPE enumeration](#)

[QUEUE\\_SUFFIX\\_TYPE enumeration](#)

## R

References

[informative](#)

[normative](#)

[overview](#)

[Relationship to other protocols](#)

## S

[SEQUENCE\\_INFO packet](#)

[SEQUENCE\\_INFO structure](#)

[Settings property identifiers](#)

[Site property identifiers](#)

[Sitelink property identifiers](#)

Structures

[discussed](#)

[overview](#)

## T

[TA\\_ADDRESS structure](#)

[tag\\_inner\\_PROPVARIANT structure](#)

## U

[ULARGE\\_INTEGER structure](#)

[Unused Active Directory properties](#)

[User object property identifiers](#)

## V

[VARENUM enumeration](#)

[Vendor-extensible fields](#)

[Versioning](#)

## W

[Windows behavior](#)