Microsoft Message Queue Server Overview

Microsoft® Message Queue Server (MSMQ) is a fast store-and-forward service for Windows NT Server, Enterprise Edition (Windows NT Server/E), that enables applications running at different times to communicate across heterogeneous networks and systems that may be temporarily offline. Applications send messages to MSMQ, and MSMQ uses *queues of messages* to ensure that the messages eventually reach their destination. MSMQ provides guaranteed message delivery, efficient routing, security, and priority-based messaging.

With MSMQ, *end users* communicate across networks and systems that are occasionally offline, independent of the current state of the communicating applications and systems. With MSMQ *developers* can focus on business programming and not on networking issues because MSMQ provides guaranteed network communication. *System administrators* can use MSMQ to efficiently manage large, complex networks of message queues using the MSMQ Explorer. Through MSMQ, *MIS decision makers* get more reliable communication in auditable, network-aware distributed applications; more efficient use of network resources; reduced software development and maintenance time; and increased productivity from end users, developers, and system administrators.

Software products with these features are often categorized as store-and-forward software, message queue software, or message-oriented middleware (MOM).

Introducing MSMQ

MSMQ version 1.0 supports the following features:

- **Connectionless messaging**. With store-and-forward message queuing, applications aren't affected by network fluctuations and do not have to establish sessions. Because MSMQ uses a sessionless model at the application level, the sender and receiver don't need to support the same protocol. MSMQ supports Internet Protocol (IP) and Internet Packet eXchange (IPX).
- **Network traffic prioritization**. Message prioritization allows urgent or important traffic to preempt less-important traffic so you can guarantee adequate response time for critical applications at the expense of less important applications.
- **Guaranteed delivery**. Messages can be logged to a disk-based queue to provide guaranteed delivery.
- **Transactions**. Using the MSMQ transaction flag you can implement transaction-based applications, ensure messages are delivered in order, ensure messages are delivered no more than once, and confirm messages reached or were retrieved from the destination queue.
- **Dynamic queues**. Queue information resides in a dynamic/replicated database so administrators can change queue properties without affecting messaging applications. Using MSMQ Explorer, administrators can make these changes from any computer running MSMQ Explorer.
- **Routing**. MSMQ supports smart routing, based on the physical topology of the network, session concentration, and transport connectivity. Session concentration allows efficient usage of slow links.
- **Security**. MSMQ supports privacy and security through access control, auditing, encryption, and authentication. Access control is implemented using Windows NT security and digital signatures. Auditing is implemented through the Windows NT event logging service. Encryption and authentication (using digital signatures) are supported using public and private keys.
- **Disparate system integration**. MSMQ-based applications can be implemented across a wide variety of hardware platforms using MSMQ connectivity products provided by Level 8 Systems. For more information, see the following section, "Supported Platforms."

Dynamic queues, integrated security, manageable scalability, and smart routing differentiate MSMQ from other middleware implementations available today.

For more information on writing MSMQ-based applications that take advantage of these features, see the Microsoft Message Queue Server Software Development Kit (MSMQ SDK) documentation.

Supported Platforms

MSMQ supports the Windows NT and Windows 95 platforms. Level 8 Systems provides a product that supports the MSMQ API on many platforms including IBM MVS and CICS; Sun Solaris, HP-UNIX, and AIX UNIX platforms; as well as OS/2; VMS; and AS/400 platforms. The Level 8 Systems product also supports the mapping of native IBM MQSeries API (MQI) calls and CICS Transient Data API calls to the MSMQ API calls.

Using the MSMQ SDK you can build your own server applications to integrate MSMQ with other message queuing applications.

These features make MSMQ ideal for the implementation of semi-independent client-server systems (such as order/entry, accounting, and inventory applications), batch processing, queue-based client-server systems (first come, first served resource access), and migration from legacy systems.

For more information on Level 8 Systems products and product availability, contact Level 8 Systems, or see the Level 8 Systems web page at http://www.level8.com/.

MSMQ Programming Environment

You can use the MSMQ application programming interface (API) to develop MSMQ-based applications in C or C++. MSMQ also includes ActiveX controls, which you can use to quickly and easily write MSMQ-based applications using Microsoft Visual Java (VJ), Visual Basic (VB), or any other ActiveX container application (for example, Microsoft Access or Borland Delphi).

With Microsoft® Active Server and Microsoft® Internet Information Server (IIS) you can integrate your MSMQ-based application with Web pages and forms. With the Messaging API (MAPI) transport provider and the Exchange connector you can integrate your MSMQ-based application with Exchange forms and MAPI clients. With the MSMQ RPC transport you can use MSMQ as a reliable transport for your RPC-based applications.

For information on developing MSMQ-based applications that use these MSMQ features, see the MSMQ SDK documentation.

Finding Information About MSMQ

This manual, the *Microsoft Message Queue Server Administrator's Guide (MSMQ Administrator's Guide)*, covers the concepts required to install and administer an MSMQ enterprise. Additional information is available online, in the MSMQ SDK, and on the Microsoft MSMQ World Wide Web site: http://www.microsoft.com/msmq.

Online Documentation

The following information is available online:

- MSMQ Readme.doc
- MSMQ Explorer Help
- MSMQ Glossary
- *MSMQ Administrator's Guide* Online
- MSMQTest.txt

MSMQ Readme.doc

The Readme.doc file located in the MSMQ folder on your Windows NT Server/E Components CD contains general information about MSMQ including software-specific information, a description of product features, and limitations that were added or identified after the documentation was created.

MSMQ Explorer Online Help

MSMQ Explorer online Help is the primary source for assistance with MSMQ Explorer. The online Help includes procedures to guide you through administrative tasks. Context-sensitive Help is available on all MSMQ Explorer property pages.

MSMQ Glossary

A complete MSMQ glossary is available online in both the MSMQ Explorer and the online version of the *MSMQ Administrators Guide*.

MSMQ Administrator’s Guide Online

This book is also available online on all MSMQ servers. The online version of the *MSMQ Administrator’s Guide* can be accessed by clicking **Start**, pointing to **Programs**, pointing to **Microsoft Message Queue**, and then clicking **Administrator’s Guide**.

MSMQTest.txt

MSMQTest, a command line application that demonstrates basic MSMQ functionality, is installed on all MSMQ clients and servers. MSMQTest usage is documented in the MSMQTest.txt file, installed along with MSMQTest.

MSMQ SDK Documentation

The MSMQ SDK documentation is installed with the MSMQ SDK on MSMQ independent clients and servers. The MSMQ SDK includes background information on developing MSMQ-based applications and a reference section that describes the functions, properties, structures, and ActiveX components supplied by, and used by, MSMQ. For more information on sample code and program files, see the MSMQ SDK..

The Microsoft World Wide Web Site

For information on upgrades and other news pertaining to MSMQ, see the Microsoft Message Queue Server World Wide Web site:

http://www.microsoft.com/msmq

Understanding MSMQ

This chapter introduces the terms and concepts you must understand to install and administer MSMQ. It begins with a terminology overview, covers the conceptual topics of interest to administrators, and concludes with a section covering some common message queuing business scenarios.

MSMQ Terminology Overview

MSMQ applications communicate using *messages*. A message is simply a unit of information or data being sent between computers. The message can contain text or binary data. *Transactional* messages can be used to pair the sending or receiving of any message with an action in another operation. Using transactional messages ensures that the unit of work is carried out as an atomic operation—that is, the operation succeeds or fails as a whole. Transactional messages can also be used to ensure that a message is delivered only once and to ensure that all messages sent from one computer to another are delivered in order. Positive and negative acknowledgements can be used to confirm messages reached or were retrieved from the destination queue.

MSMQ supports two delivery methods: *express* and *recoverable*. Choosing between express and recoverable delivery is a matter of trading performance and resource use for reliability and failure recovery. In general, express messages use fewer resources and are faster than recoverable messages. However, express messages cannot be recovered if the computer storing the memory-mapped message files fails. Recoverable messages use more resources and are slower than express messages, but can be recovered no matter which computer fails.

MSMQ uses public and private *queues* to store and forward messages. All MSMQ queues, regardless of their function, can be manipulated with the same MSMQ functions. This includes the special *journal*, *dead letter*, *transactional dead letter*, *administration*, *system*, and *report* queues. Each of the queues is simply a standard MSMQ queue used for a specific purpose. For more information on the MSMQ API, see the Microsoft Message Queue Server Software Development Kit (MSMQ SDK).

Queue quotas and *computer quotas* specify the cumulative limit for messages in a queue or in all queues on a computer. The queue and computer quotas are based on size, and can be set independently. When a queue quota is reached, messages can no longer be sent to the queue until one or more messages are removed from the queue. When a computer quota is reached, messages can no longer be sent to any queues on the computer until one or more messages are removed from one of the queues.

MSMQ routes and delivers messages based on a combination of *queue priority* and *message priority*. Messages are routed and delivered by queue priority first, and message priority second.

MSMQ supports *dependent clients*, *independent clients*, and *servers*. Both independent clients and servers run the MSMQ service and can communicate asynchronously. MSMQ dependent clients require synchronous access to an MSMQ server.

Some MSMQ servers hold copies of the *MSMQ information store (MQIS)* database. The MQIS is a distributed database that holds enterprise topology, enterprise settings, computer information, and queue information. MSMQ-based applications can query the MQIS to find queues and get queue properties.

All computers operate within one MSMQ *enterprise*. The enterprise is divided into *sites*, where communication between any two computers is fast and inexpensive. Sites are connected through *site links*. *Site link costs* define the cost of sending messages between sites. Computers running in MSMQ communicate over *connected networks (CNs)*. A CN is a collection of computers where any two computers can communicate directly. MSMQ servers designated as *in routing servers (InRSs)*, *out routing servers (OutRSs)*, and *site gates* can be used to control the flow of messages and provide *session concentration*. MSMQ servers take all these factors into account when *routing* messages within your MSMQ enterprise.

Topology and Connectivity

Before installing or configuring MSMQ, you must understand the following terms:

- MSMQ Enterprise
- MSMQ Sites
- MSMQ Connected Networks

MSMQ Enterprise

In MSMQ, all computers that run MSMQ belong to one enterprise and access information from the same distributed database, called the MSMQ information store(MQIS). To simplify administration and for compatibility with future versions of Windows NT Server, you should not install multiple enterprises within your organization. Issues relating to security and isolating the use of MSMQ between groups within your organization can be addressed using MSMQ security features.

However, if you choose to have more than one enterprise within a company, or wish to exchange MSMQ messages with another company (for example, over the Internet) you can still do so. For information on sending messages between enterprises, see the MSMQ SDK.

MSMQ Sites

A site is a physical collection of computers where communication between any two computers is fast and inexpensive. Site boundaries usually parallel the physical location of the computers (for example, all computers within a building). However, all computers in a site do not necessarily have to be running the same protocol, and computers in the same site may not be able to directly communicate with each other.

Sites are connected to other sites through communication links called *site links*. Inter-site routing is the process of sending messages between sites on these links. MSMQ calculates inter-site routing based on relative numbers that administrators assign to site links. These numbers, called *site link costs*, represent the cost of communication of that link.

Although establishing site boundaries in MSMQ is fairly simple, additional factors should be taken into consideration for compatibility with the site object in future versions of Windows NT Server. For more information, see “Topology” in Chapter 6, “Deploying MSMQ.”

Site Link Costs

MSMQ calculates inter-site routing based on the *cost* of each site link. Site link costs are defined using relative numbers between zero and 999,999. It is up to the administrator to define the relative cost of routing between sites. An administrator typically balances cost with delay (the speed of one link versus another).

You set site link costs when you install new sites. If you have only two sites, choose any value above zero. If you have three or more sites, and the cost of routing between sites is more or less equal, use the same value for each site link. However, if you have three or more sites, and the cost of routing between sites is not equal, use site link costs to define the difference in the routing costs. For example, suppose you have two sites called A and B in one city connected by a high-speed link, and one site called C, which is overseas and connected to site B by a low-speed link. Define the site link cost between A and B as 1, and between B and C as 2.

A site link cost of zero indicates that the two sites are not connected.

For more information on defining site links, see “MSMQ Routing” later in this chapter, “Installing a PSC” in Chapter 2, Installing MSMQ, and MSMQ Explorer Help.

MSMQ Connected Networks

Within MSMQ, a connected network (CN) is a collection of computers where any two computers can communicate directly. The computers within a CN must support the same protocol and must be able to establish a session. A computer can belong to multiple CNs, and CNs can span sites. However, all computers in a physical local area network (all computers monitoring the same broadcasts) that use the same protocol (IP or IPX) must belong to the same CN.

When you define a CN, you are simply defining a label. When you install an MSMQ server, you associate each network address on the computer with the appropriate CNs. These CNs form logical groupings of computers that can communicate directly. MSMQ dependent clients use their supporting server CNs, and MSMQ independent clients determine their CNs automatically.

Tip When you define CNs for your enterprise, use meaningful labels so that administrators can easily choose a CN from a list when overriding the default CN settings. For more information, see “Naming Conventions” in Chapter 6, “Deploying MSMQ.”

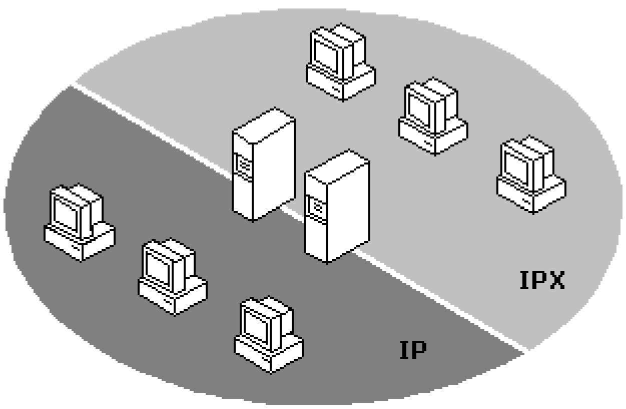


Figure 1.1

In most cases, the CN layout in an enterprise mirrors the protocol use. For example, in Figure 1.1, half the client computers run IP, the other half run IPX, and the servers run both IP and IPX. The MSMQ clients running IP belong to an IP CN and the MSMQ clients running IPX belong to the IPX CN. The servers belong to both CNs. However, CNs are not always defined to mirror protocol use. For example, you need two CNs when you have two separate IPX networks (subnets) on the same local area network.

Figure 1.2 shows an enterprise with two sites. The enterprise has two CNs, with one of the CNs spanning both sites. The servers belong to both the IP CN and the IPX CN to support the routing of MSMQ messages between CNs.

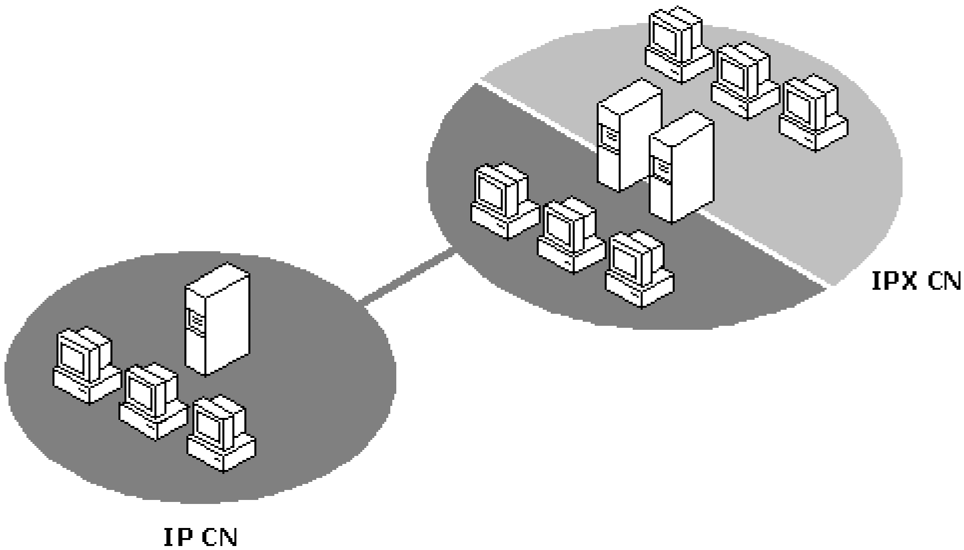


Figure 1.2

The preceding figure shows two sites. one is covered by only one CN, while the other is covered by a CN shared with the first site and a CN that is used only in the second site. In this example, at least one of the two servers in the rightmost site must belong to both CNs to allow computers in the IPX CN to communicate with computers in the IP CN.

MSMQ Servers

MSMQ uses four server types to control message queuing:

- Primary enterprise controller (PEC)
- Primary site controller (PSC)
- Backup site controller (BSC)
- MSMQ routing server

Each of these servers must be installed on a computer running Windows NT Server, Enterprise Edition (NTSE). The specific functions of these servers are explained in the following sections. Your MSMQ enterprise must contain one PEC. The installation of other server types is optional, depending on your network topology, the size of your MSMQ enterprise, and the types of MSMQ-based applications you develop. Any one of these servers can also function as an MSMQ connector server. You can also configure Windows NT RAS servers to support remote MSMQ independent clients.

MSMQ Explorer uses a variety of icons to display the different types of computers. To see a table that defines the computer, queue, and message icons, click the Help button on the MSMQ toolbar and click anywhere in the MSMQ enterprise.

For more information on determining the appropriate number of MSMQ servers for your MSMQ network and their location and configuration, see Chapter 6, “Deploying MSMQ.” For more information on installing the various MSMQ servers, see Chapter 2, “Installing MSMQ.”

MSMQ Server Licensing Considerations

MSMQ does not limit the number of client/server sessions. However, MSMQ does count the number of Windows NT client access licenses (CALs) and restricts concurrent MSMQ client sessions accordingly. MSMQ supports both Per Server and Per Seat licensing.

For more information on CALs, see *Microsoft Windows NT Server Concepts and Planning*, Chapter 12, “Licensing and License Manager.”

Primary Enterprise Controller

Administrators install one *primary enterprise controller (*PEC*)* on an MSMQ network. The PEC functions as a PSC for one site. The PEC holds information about the enterprise configuration and the certification keys (used in authenticating messages) in a database, and also functions as an MSMQ routing server. You must install a PEC before you can install any PSCs.

Primary Site Controller

You install one *primary site controller (*PSC) for each additional site in your MSMQ network – The PEC functions as the site controller for the initial site you create. The PSC holds information about the computers and queues in the site in a database, and also functions as an MSMQ routing server. Although you do not have to install a PSC in each site (as defined in the preceding section, “MSMQ Sites”), it is highly recommended that you do.

Using MSMQ in a Site without a PSC

When you use MSMQ in a site without a PSC (or any other MSMQ server), you lose many of the benefits of MSMQ , including the efficient and persistent routing of messages. When the site is disconnected from the PEC site, independent clients within the site cannot locate queues or create queues, and each client uses additional resources as it attempts to resend undeliverable messages. MSMQ dependent clients can only be used within the site when the site controller is online.

Backup Site Controller

A site does not require a backup site controller (BSC). However, one or more BSCs should be installed at each site to provide load balancing and failure recovery, should the PSC or PEC fail. The BSC holds a read-only replica of the PSC or PEC database and also functions as an MSMQ routing server. You must install a PEC or PSC before you can install any BSCs.

MSMQ Routing Server

MSMQ routing servers support dynamic routing and intermediate store-and-forward message queuing. They allow computers using different protocols to communicate and can be used to provide session concentration. Unlike BSCs, MSMQ routing servers do not hold a read-only replica of the PSC or PEC database. You must install a PEC before you can install any MSMQ routing servers.

Because every PEC, PSC, and BSC functions as an MSMQ routing server, the number of additional MSMQ routing servers you install depends on your MSMQ connectivity requirements (number of dependent clients, independent clients, sites, CNs, session concentration needs, and message volume). At a minimum, you should strategically install enough MSMQ routing servers to allow messages to reach target queues through different servers.

MSMQ Connector Server

Any MSMQ server can also function as an MSMQ connector server. MSMQ connector servers allow MSMQ-based applications to communicate with computers (called *foreign computers*) that either use other messaging systems or support the MSMQ functions on hardware not supported by Windows 95 or Windows NT. MSMQ connector servers use *foreign CNs* and *foreign queues* to communicate with foreign computers, as shown in Figure 1.3. The Level 8 Systems MSMQ message queuing product is an example of an MSMQ connector server.

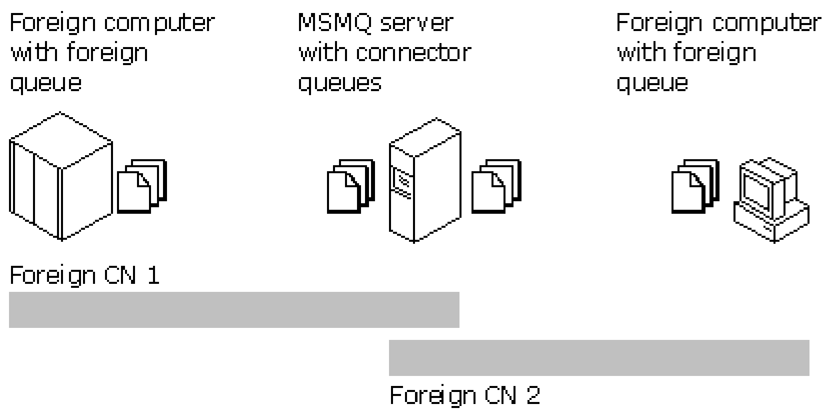


Figure 1.3

Note MSMQ connector servers and the foreign computers that they support must be in the same site.

To develop an MSMQ connector server, you must obtain the MSMQ Connector Software Development Kit (SDK).

Windows NT RAS Servers and the MSMQ RAS Connectivity Service

A Windows NT RAS server with the MSMQ RAS connectivity service installed allows remote MSMQ independent clients to connect to your MSMQ network. To reduce long distance charges and ensure independent clients always connect to the same site, you should configure a Windows NT RAS server with the MSMQ RAS connectivity service in each site. While it is possible to configure MSMQ independent clients to connect to your MSMQ network through a RAS server in any site, this configuration does not provide the same benefits.

MSMQ servers (PEC, PSC, BSC, and routing servers) running RAS and the MSMQ RAS connectivity service must use only one IP CN. If the MSMQ server is configured to use more than one IP CN, automatic site and CN recognition may fail. As a result, MSMQ workstations that connect to your MSMQ enterprise through RAS may have poor messaging performance, or may not be able to send or receive messages.

Note Windows NT RAS servers with the MSMQ RAS connectivity service installed should not support the NetBEUI protocol over RAS. The MSMQ RAS connectivity service does not support full communications with MSMQ independent clients over a RAS link when RAS support for NetBEUI is enabled.

RAS support for NetBEUI is disabled by default. To view your RAS protocol settings, run Control Panel, double-click **Network**, click the **Services** tab, click **Remote Access Service**, click **Properties**, click the appropriate modem, and then click **Network**.

MSMQ Independent Clients

MSMQ independent client software can be installed on computers running Windows 95, Windows NT Workstation version 4.0 or later, and Windows NT Server version 4.0 or later. MSMQ independent clients can create and modify queues locally and send and receive messages, just as MSMQ servers can. MSMQ independent clients can create queues and store messages on the local computer, without synchronous access to an MSMQ server. The primary differences between MSMQ independent clients and MSMQ servers are that independent clients do not have the intermediate store-and-forward capability of MSMQ servers, nor do they store information from the distributed MSMQ database.

In addition to the basic MSMQ files, you can install the MSMQ Software Development Kit (SDK) on MSMQ independent clients.

You can also install the MSMQ Explorer on MSMQ independent clients running under Windows NT Workstation or Server so that you can administer your MSMQ enterprise remotely from computers running Windows NT Workstation.

Client Session Limitations

MSMQ clients (both independent and dependent) are limited to a maximum of ten concurrent sessions with other MSMQ clients.

Disconnected Messaging

MSMQ independent clients can send messages to public queues while disconnected from the network. The “disconnect” can be a brief interruption in a network server or the mobile use of a laptop or portable computer.

MSMQ supports disconnected messaging automatically, without additional application design or network configuration. However, you cannot install MSMQ independent clients while the PSC is disconnected because MSMQ Setup must have access to the MQIS on the PSC.

Connecting Through Multiple Sites

MSMQ independent clients can move to other CNs and sites without manual reconfiguration. However, MSMQ independent clients that connect to a new site cannot receive messages that are sent to them while they are disconnected from the network unless they specify the new site before disconnecting from the network.

When an MSMQ independent client moves to a different site, its InRS and OutRS settings are ignored (when routing messages) until the workstation returns to its original site. You can reconfigure InRS and OutRS settings for the independent client while it is out of its original site, but the InRS and Out RS settings are still ignored. When the independent client is returned to its original site, InRS and OutRS settings are again used when routing messages.

MSMQ does not support mobile servers. To move a server to another CN, simply use MSMQ Explorer to change the server's CN assignments. To move a server to another site you must uninstall the server and then reinstall it in the new site.

How Automatic Site Recognition Works

On MSMQ independent clients, the MSMQ service sends out a broadcast when it starts, and monitors all replies. If a site controller other than the independent client's current site controller replies, and if the new site controller can communicate with the site controller in the independent client's original site, the independent client connects to the new site.

If you have more than one site controller in a broadcast segment (for example, in an MSMQ lab) more than one site controller can reply to the independent client's broadcast. If one of the site controllers that replies is the controller for the independent client's current site, the independent client does not connect to another site.

For information on how to use the MSMQ option in Control Panel to preset your new site, see "Preparing to Travel to a New Site" in Chapter 3, "Managing Your MSMQ Enterprise."

Connecting Through a RAS Server

MSMQ independent clients can connect to your MSMQ network through a Windows NT RAS server configured with the MSMQ RAS connectivity service. As mentioned previously, The MSMQ RAS connectivity service works in conjunction with Windows NT RAS server to support MSMQ independent clients over RAS links.

MSMQ independent clients can connect to the MSMQ enterprise through a RAS server (included with Windows NT Server), even if the RAS server belongs to a site or CNs that differ from the independent client's default site or CNs.

MSMQ independent clients connected to your MSMQ network through a RAS server have full MSMQ independent client functionality.

Note You cannot dial in to two different sites sequentially over a RAS line without first restarting the computer, or, under Windows NT, stopping and then restarting the MSMQ service.

MSMQ Dependent Clients

MSMQ dependent clients function much like MSMQ independent clients; however they cannot function without synchronous access to a PEC, PSC, BSC, or MSMQ routing server (referred to as the dependent client's *supporting server*). MSMQ dependent clients rely on their assigned server to perform all standard MSMQ functions on their behalf (such as creating queues, sending messages, and receiving messages).

MSMQ dependent clients can be installed on computers running Windows 95, and Intel-compatible computers running Windows NT Workstation or Windows NT Server. (MSMQ dependent clients cannot be installed on Alpha computers running Windows NT). MSMQ servers can support up to 15 dependent clients.

The following diagram shows the logical configuration in which MSMQ dependent clients function:

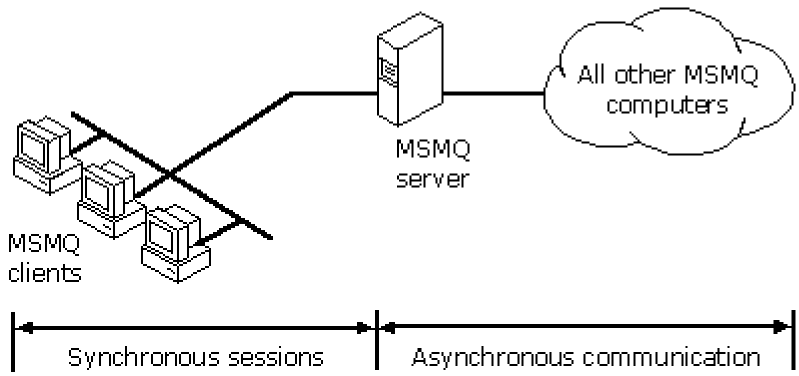


Figure 1.4

This configuration provides the following advantages:

- MSMQ dependent clients on computers running Windows 95 can send and receive transactional messages.
- Server disk space is used to store messages, reducing dependent client hardware requirements. Should more resources be required (memory or hard disk space) fewer computers must be upgraded.
- With fewer points of administration you have fewer computers to back up and fewer journal and dead letter queues to monitor.

This configuration also results in the following limitations:

- Because the MSMQ service runs on the supporting server, encrypted messages sent to or received by MSMQ dependent clients travel between the dependent client and supporting server in an unencrypted format.
- Because MSMQ dependent clients run the MS DTC proxy, they cannot transact MSMQ functions with transactional resources on the dependent client computer–they can transact MSMQ functions only with transactional resources on the supporting server.
- Because MSMQ dependent clients do not run the MSMQ service, they do not appear in MSMQ Explorer. However, when you view the properties of a server, the Dependent Clients tab displays which dependent clients the server supports. You can also use MSMQ Explorer to view the creation of queues and messaging activity for dependent clients on the dependent client's supporting server.

Note An MSMQ dependent client and its assigned server do not have to be in the same site. However, because all of the dependent client's communication goes through the server, if the communication to the server is slow or expensive, all dependent client communication will also be slow or expensive.

For information on the message time-to-reach-queue property, see the MSMQ SDK.

Client Session Limitations

MSMQ clients (both independent and dependent) are limited to a maximum of ten concurrent sessions with other MSMQ clients.

Routing

MSMQ establishes a direct connection (a session) using the underlying protocol if possible. When a direct connection is not possible or not allowed MSMQ uses its own routing system. MSMQ routing occurs when one or more of the following conditions exist:

- a session cannot be established between the sender and the receiver (for example, the source and target computers do not share a common CN or the target computer is offline).
- In Routing Servers (InRSs) or Out Routing Servers (OutRSs) are defined for the sender or receiver.
- messages must travel between two sites, and one or both sites have a site gate defined.

MSMQ servers make two assumptions about your MSMQ network: *Intra-site routing* is fast and inexpensive, while *inter-site routing* is slow and expensive.

Intra-site Routing

Intra-site routing is the process of routing messages within a site. MSMQ measures intra-site routing in *hops*, which is the number of MSMQ servers (PEC, PSCs, BSCs, or MSMQ routing servers) a message must pass through before reaching its destination. MSMQ always chooses the shortest available path when routing a message (unless routing restrictions have been applied by an administrator).

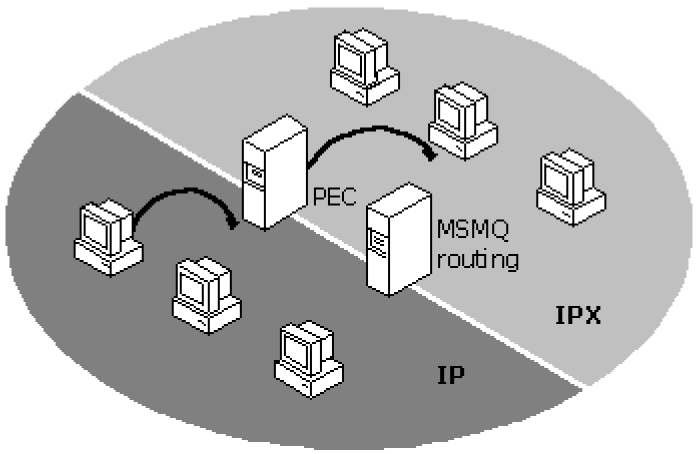


Figure 1.5

In Figure 1.5, any message sent from a computer in the IP CN to a computer in the IPX CN must make two hops-one hop to get to one of the MSMQ servers (assuming they both connect to both CNs) and one hop to get to the target computer. Because each hop is within a site, the cost of each hop is minimal.

The MSMQ intra-site routing system ensures that messages sent between two computers are delivered even if the two computers are never online at the same time.

For information on configuring intra-site and inter-site routing restrictions, see “Session Concentration” later in this chapter.

Inter-site Routing

Inter-site routing is the process of routing messages between sites. MSMQ calculates inter-site routing based on a cost defined by the administrator.

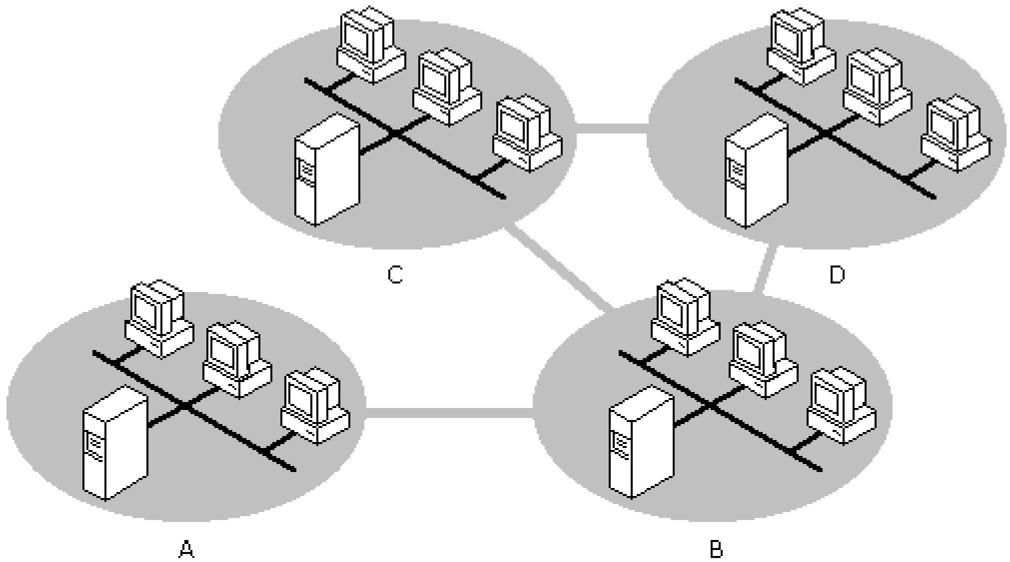


Figure 1.6

The route a message takes to get from site A to site C depends on the cost of the site links. If the cost associated with the A-B site link is 3, and the cost associated with the other three site links (B-C, B-D, and C-D) is 1, messages routed from site A to C will always travel from site A to site B and then to site C. However, if the cost associated with the A-B and B-C site links is 3, and the cost associated to the C-D and B-D site links is 1, messages routed from site A to C will always travel from site A to site B, to site D, and then to site C. If the B-C link, B-D link, or the C-D link is down, messages will be routed using any available link.

For more information on defining the cost of inter-site routing while installing a PSC, see "Installing a PSC" in Chapter 2, "Installing MSMQ." For more information on changing the cost of inter-site routing between existing sites, see MSMQ Explorer Help.

Routing Between Enterprises

MSMQ-based applications can send messages to and receive messages from other enterprises. This allows MSMQ-based applications to communicate over the Internet.

Note MSMQ communicates over TCP port 1801, registered with the Internet Assigned Numbers Authority (IANA). MSMQ-based applications can communicate over properly configured firewalls that allow communication over that port.

For more information on writing MSMQ-based applications that send messages between enterprises, see the MSMQ SDK.

Full Enterprise Connectivity

An MSMQ site is considered to be fully connected if it is possible to route messages, either directly or using MSMQ servers (PEC, PSCs, BSCs, or MSMQ routing servers), between every computer in the site, without having to go to another site. For example, a site does not have full connectivity if it has two CNs, and the only computer that connects the two CNs is an MSMQ independent client.

An MSMQ enterprise is considered to be fully connected if sites are fully connected.

Session Concentration

Session concentration can be used to:

- Reduce sessions within a site
- Reduce sessions between sites

Unnecessary sessions can increase connection and bandwidth costs. MSMQ session concentration reduces these costs by funneling connections through MSMQ servers.

MSMQ supports two types of session concentration: intra-site and inter-site. Intra-site session concentration typically reduces network bandwidth use within a site. Inter-site session concentration typically reduces the number of sessions between sites.

By loading specific servers with more independent clients, you can use session concentration to manually load or tune your MSMQ network. For example, all MSMQ independent clients in one department can be configured to send and receive all messages through a specific server or set of servers.

Session Concentration Within a Site

Intra-site session concentration is done by configuring MSMQ independent clients to use between one and three dedicated MSMQ servers, called In Routing Servers (InRSs) and Out Routing Servers (OutRSs). If as independent client is configured to use an OutRS, every outgoing MSMQ message sent by the independent client is routed to the OutRS. Likewise, if an independent client is configured to use an InRS, every message sent to the independent client is routed through the InRS

Only MSMQ independent clients can be configured to have either InRSs or OutRSs−MSMQ dependent clients and MSMQ servers cannot be configured to use InRSs or OutRSs. By default, MSMQ independent clients are not configured to use InRSs or OutRSs.

The PEC, PSCs, BSCs, and MSMQ routing servers can all be used as InRSs and OutRSs. The same MSMQ server can be used as an independent client's InRS and OutRS. However, the InRSs and OutRSs must be in the independent client's original site, and must have at least one CN in common. When a user travels to another site with an independent client, InRS and OutRS settings are disabled until the independent client is returned to its original site.

The cost of session concentration is one additional hop for each message and subsequent additional load on the server. The enterprise administrator must determine the best tradeoff between session concentration versus hops and load balancing based on the configuration and network load.

As mentioned previously, Intra-site session concentration typically reduces network bandwidth usage. For example, if you use a star topology within a site, and each computer usually communicates directly with every other computer, you can greatly reduce your bandwidth usage by using an MSMQ-based application and configuring each MSMQ independent client to use the PSC as its InRS and OutRS.

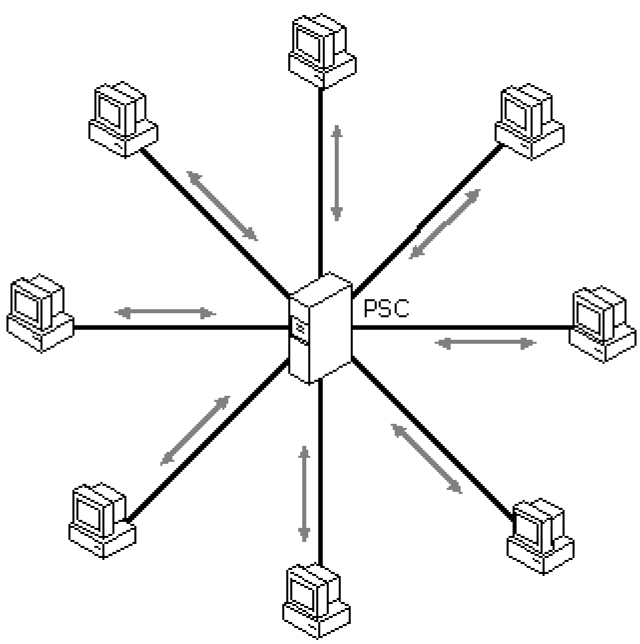


Figure 1.7

In Figure 1.7 the PSC is being used as both an InRS and an OutRS for each MSMQ independent client. This reduces the total number of possible sessions from 36 to 8.

Because computers configured with InRSs and OutRSs are dependent on the MSMQ server, it is preferable to assign more than one InRS or OutRS to independent clients to provide failure recovery.

For information on how to specify an independent client's InRS or OutRS, see MSMQ Explorer Help.

Session Concentration Between Sites

Inter-site session concentration is done by establishing site gates. If a site is configured to use a site gate, every MSMQ message sent between computers in different sites must be routed through the site gate. In the following figure, the routing topology beyond the site is transparent to the computers within the site, with the exception of the site gate. Thus, the message route is simplified. By default, sites do not use site gates.

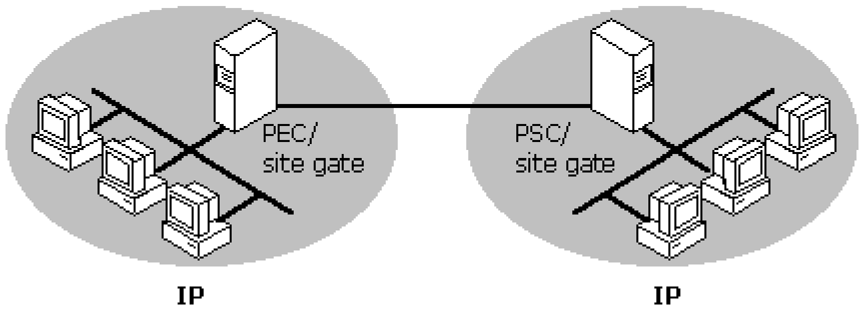


Figure 1.8

In Figure 1.8, site-to-site sessions are reduced. Because each independent client must route messages to its site gate, only the PEC and PSC can establish a session that spans the site.

The following are the requirements for site gates:

- The computer must belong to a site to be a site gate for the site
- The computer must be able to connect to each neighboring site

As long as the computer meets these requirements, the computer can be a PEC, PSC, BSC, or MSMQ routing server.

Because sites configured with site gates are dependent on the site gate for all inter-site transmission of MSMQ messages, you should assign more than one site gate to a site to provide load balancing and failure recovery. There is no limit to the number of servers you can assign as site gates.

For information on how to specify a site’s site gate, see MSMQ Explorer Help.

Queue and Message Priority

MSMQ routes and delivers messages based on a combination of queue priority and message priority. Messages are routed and delivered by queue priority first, and message priority second.

Queue priority (called the *base priority*) for public queues can be set by any MSMQ-based application with write permissions for the queue. The priority can be set at any time. The queue priority can be set from -32768 to 32767. The default priority for public queues is 0. Private queues do not support queue priority.

MSMQ-based applications can send messages with a priority of 0 through 7, 7 being the highest priority.

For more information on changing the base priority of a queue from your MSMQ-based application, see the MSMQ SDK. For more information on using MSMQ Explorer to change the base priority of a queue, see MSMQ Explorer Help.

The MSMQ Information Store

The MSMQ information store (MQIS) is a distributed database. MSMQ stores the following information in the MQIS:

- Enterprise topology (such as sites, CNs, InRS/OutRS assignments)
- Enterprise settings (such as enterprise name, PEC name, default replication intervals)
- Computer information
- Queue information

SQL Server Requirement for the MQIS

MSMQ controller servers (the PEC, PSCs, and BSCs) use a Microsoft SQL Server version 6.5 database to store the MQIS. You can purchase SQL Server 6.5 for each MSMQ controller server, or you can install a limited version of SQL Server 6.5 when you install each controller server.

For information on MSMQ SQL Server requirements, see “Microsoft SQL Server Requirements” in Chapter 2, “Installing MSMQ.”

No two computers in your MSMQ network can have the same computer name (also called “friendly” computer names), even if the computers are on separate, non-connected networks.

Everyone has full read access to the MQIS so that all MSMQ users can search for queues within your enterprise. However, access to those queues can still be controlled using a combination of Windows NT security and signed messages. For more information on access control, see “Enabling Access Control and Authentication” in Chapter 5, “Securing Your MSMQ Enterprise.”

Note Although the properties of each MSMQ queue reside in the MQIS, the contents of the queues (the messages themselves) reside in memory-mapped files on MSMQ independent clients and servers.

For more information about how MSMQ stores messages, see “Message Delivery” later in this chapter.

MQIS Replication and Ownership

Although the MQIS database is replicated between sites, different servers own and control different sets of data. The MQIS database on the PEC contains the master copy of enterprise, site, site link, CN, and user settings, as well as a master copy of its site's computers and queues. The MQIS database on PSCs contains a master copy of their site's computers and queues.

Each PSC MQIS (including the PEC MQIS) also contains a copy of the MQIS information owned by other sites. Each BSC contains a replicated MQIS database from the PSC (or PEC) in its site. The PSCs and the PEC have write access to the MQIS information they own, and read-only access to all replicated information that they receive from the PEC or other PSCs. Each BSC has read-only access to its replicated MQIS databases.

Replication Mechanics

Data is replicated from the owner directly to the other sites using MSMQ messages and private queues. A PEC replicates the enterprise and its site-specific data directly to other PSCs. Each PSC replicates its site-specific settings directly to the PEC. This is called *inter-site replication*. Each PSC(and the PEC) replicates the information it owns, and the information it receives from other PSCs, to the BSC at its site. This is called *intra-site replication*. PSCs (including the PEC) never replicate information to BSCs in other sites.

For example, in Figure 1.9, the MQIS database on the PEC at site A contains the master database for enterprise, site, site link, and CN settings; the master database for site A computer and queue settings; and replicated data owned by sites B and C. The MQIS database on the PSC at Site B contains the master database for site B computer and queue settings; replicated computer and queue settings from site C and from site A; and replicated site, site link, CN, and enterprise settings from the PEC. The MQIS database on the PSC at Site C contains the master database for site C computer and queue settings; replicated computer and queue settings from site B and from site A; and replicated site, site link, CN, and enterprise settings from the PEC.

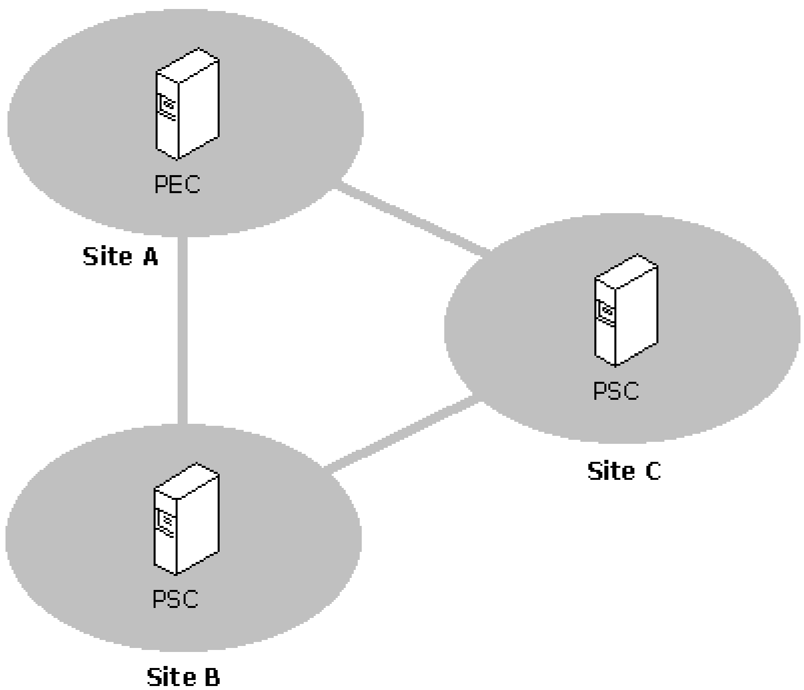


Figure 1.9

Because intra-site replication is quicker and cheaper than inter-site replication, it takes place more often. By default, intra-site replication occurs every 2 seconds, while inter-site replication occurs every 10 seconds. However, the database is immediately replicated whenever the site topology changes (for example, when a site is added).

For information on how to change MSMQ replication intervals, see MSMQ Explorer Help.

Queues

All MSMQ queues, regardless of their function, can be manipulated with the same MSMQ functions. This includes the private and public queues used to send and receive messages, as well as the special journal, dead letter, transactional dead letter, administration, and report queues used to add functionality to your MSMQ applications. Each of the queues described in the following sections is simply a standard MSMQ queue used for a specific purpose.

MSMQ Explorer uses a variety of icons to display the different types of queues. To see a table that defines the computer, queue, and message icons, click **Help Topics** on the MSMQ Explorer **Help** menu, double-click **Overview**, and then double-click **MSMQ Explorer Icons**.

Public Queues and Private Queues

Public queues are those published in the MQIS. All public queues are replicated throughout the enterprise, and can therefore be located by any computer within the enterprise.

Private queues are not published in the MQIS, and therefore don't add to the MQIS replication load. Private queues can be accessed only by applications that have access to the full pathname or formatname of the queue. In the default view, MSMQ Explorer does not display private queues.

For more information on viewing private queues, see MSMQ Explorer Help.

Journal Queues

The process of logging or storing a copy of a message in a queue is called *journaling*. Journal messages are stored in a queue called *Journal*.

Two types of journaling are available: source and target. *Source journaling* is the process of storing a copy of an outgoing message. It is configured on a message basis, and is therefore a property set by the sending application.

When source journaling is enabled for a non-transactional message, a copy of the message is placed in the *source journal queue* of the sending computer when the message arrives at the target queue, or when the message reaches an MSMQ server that is one hop closer to the target computer. When source journaling is enabled for a transactional message, a copy of the message is placed in the *source journal queue* of the sending computer when the message arrives at the target queue. A copy of the message is never stored in a source journal queue on a server that forwarded the message.

Source journaling messages for MSMQ dependent clients are placed in the source journal queue on the supporting MSMQ server. In MSMQ Explorer, the source journal queue is displayed underneath the computer.

Target journaling is the process of storing a copy of incoming messages. It is configured on a queue basis. When target journaling is enabled, a copy of each incoming message is placed in the *target journal queue* when the message is removed (read) from the target queue. A target journal queue (simply called Journal) is created for each queue when the queue is created. In MSMQ Explorer, target journal queues are displayed under each public queue.

For information on enabling target journaling, see MSMQ Explorer Help. For information on enabling source journaling, see the MSMQ SDK.

Dead Letter Queues

Messages that have expired or are undeliverable are stored in dead letter queues. A non-transactional message that cannot reach the destination application is stored in a dead letter queue on the computer where the message expired (or failed): this can be the source computer, the destination computer, or an MSMQ server that is forwarding the message.

Dead letter queues are created for each independent client and server on your MSMQ network when MSMQ is installed on that computer. These queues are displayed under the computer in MSMQ Explorer.

Messages are undeliverable when:

- The destination queue is unknown
- The message has exceeded its maximum number of hops
- The message time-to-be-delivered (TTBD) has expired
- The message time-to-reach-queue (TTRQ) has expired
- The queue’s quota has been reached
- A non-transactional message is sent to a transactional queue

Messages are not stored in dead letter queues when the message sender is not allowed to send messages to the destination queue. Messages in dead letter and journal queues are always recoverable.

For more information on message properties (such as TTBD and TTRQ) see the MSMQ SDK. For more information on recoverable messages, see "Message Delivery" later in this chapter. For more information on queue and computer quotas, see "Queue and Computer Quotas" later in this chapter.

Transaction Dead Letter Queues

A transactional message that cannot reach the destination application is stored in the transaction dead letter queue on the source computer. Transaction dead letter queues are created for each independent client and server on your MSMQ network when MSMQ is installed on that computer. these queues are displayed as **Xact Dead Letter** under the computer in MSMQ Explorer.

Transactional messages cannot reach the destination application when:

- One or more units of the transaction cannot be completed
- A transactional message is sent to a non-transactional queue

For more information on transactional messages, see “Transactional Messaging” later in this chapter.

Administration Queues

Administration queues contain acknowledgment messages generated by MSMQ for messages you send. These messages indicate that the messages you sent either arrived (a positive acknowledgment) or that an error occurred before the message could be retrieved (a negative acknowledgment). Positive acknowledgments indicate whether or not the message was retrieved by the target application. Negative acknowledgments indicate why the message was not retrieved.

The application programmer decides whether to use acknowledgements. If they are used, the application programmer must specify an administration queue in the source application and the type of acknowledgment messages to send to the queue.

Note Negative acknowledgments (NACKs) for encrypted messages are sent without the message body. Because only the destination computer can decrypt the message body, the message body is not useful on other computers.

For more information on administration queues, see the MSMQ SDK.

Report Queues

Report queues contain report messages that indicate the message's route through your enterprise. Report queues can be used when:

- Sending test messages
- Tracking message routes for a specific application

For more information on sending test messages and enabling message route tracking, see "Tracking and Monitoring Messages" in Chapter 4, "Monitoring Your MSMQ Enterprise."

System Queues

MSMQ uses up to six system queues. All six system queues are implemented as private queues. Each queue and its use is described in the following table.

Table 1.1

System Queue	Use
admin_queue$	Storing administrative messages (for example, test messages).
explorer_response_queue	Storing administrative response messages (except for MQPing response messages).
mqping response	Storing returned MQPing requests.
mqis_queue$	Storing MQIS replication messages.
notify_queue$	Storing notification messages. These messages inform a computer about a change in its properties or a change to one of its queues (including creation and deletion of queues).
order_queue$	Tracking transactional messages that require in-order-delivery.

The MSMQ system queues have a base priority of 7, and are not affected by a computer's message quota.

System queues cannot be deleted. Those users with Administrative privileges on their local computer have full control of the system queues on their computer, with exception to the Delete permission. Everyone has Get Properties, Get Security, and Send permission for system queues.

Message Delivery

MSMQ messages can contain text or binary information in the message body. The contents of the message body are specified by the sending application—MSMQ handles the message delivery but does not alter the message contents. MSMQ messages can be no larger than 4 megabytes.

To choose the appropriate delivery method for an MSMQ-based application, and to calculate required disk space on MSMQ independent clients and servers, you must understand Message storage details, The two delivery types available, and Journaling and Dead letter queues.

To install and administer an MSMQ network, you must understand:

- The two message delivery types available
- Message storage details
- Messaging limitations
- Queue and computer quotas
- Transactional messaging

MSMQ Explorer uses a variety of icons to display the different types of messages. To see a table that defines the computer, queue, and message icons, click **Help Topics** on the MSMQ Explorer **Help** menu, double-click **Overview**, and then double-click **MSMQ Explorer Icons**.

Express and Recoverable Delivery

MSMQ supports two delivery methods: express and recoverable. Choosing between express and recoverable delivery is a matter of trading performance and resource use for reliability and failure recovery.

In general, express messages use fewer resources and have faster throughput than recoverable messages. However, express messages cannot be recovered if the computer storing the memory-mapped message files fails. Recoverable messages use more resources and have slower throughput than express messages, but can be recovered no matter which computer fails.

For example, if an MSMQ-based application is sending express messages to a queue through an MSMQ routing server, and the network link between the MSMQ routing server and the destination computer fails, the messages will continue to be stored in memory on the MSMQ routing server. If the MSMQ routing server is shut down before the network link is restored, the express messages are lost. However, if recoverable messages are used, the messages are not lost and are delivered after the MSMQ routing server and network link are brought back online.

The sending application, not MSMQ, specifies the delivery method.

Message Storage Details

While queue information is stored in the MQIS, messages are stored in memory-mapped files (.mq files) on the sending, receiving, or intermediate MSMQ routing server's hard disk. MSMQ allocates disk space for both express and recoverable messages in 4 MB files, as needed. For each recoverable message, the write operation is flushed to disk, ensuring the data is recoverable. For express messages, memory-mapped files are written to disk only when the computer has insufficient resources to store the file in memory.

The memory-mapped files are created on the computer sending the message. As messages are forwarded by intermediary store-and-forward servers, they are also stored on those servers in memory-mapped files. If source or target journaling is being used, the memory-mapped files are kept on the sending or receiving computers, respectively. MSMQ servers store all messages for the MSMQ dependent clients they support.

The memory-mapped files can be stored on any file system supported by the underlying operating systems. However, because the Windows NT File System (NTFS) provides more reliable file recovery after a power failure or system crash, it is the recommended file system for MSMQ servers.

Messaging Limitations

Messaging throughput is limited by the hardware being used. Processing power, available memory, and network capacity have the greatest affect on messaging throughput. The number of messages each MSMQ independent client and server can store at one time is limited first by the available disk space on the local computer, and second, by the code and data space of the operating system on which MSMQ is running.

When running on Windows 95, MSMQ independent clients can store close to 1 GB of messages. Specifically, MSMQ independent clients can store 1 GB of messages minus the memory consumed by the overhead of storing the message and the memory of other applications using the shared system memory.

When running on Windows NT version 4.0, MSMQ independent clients and servers can store just under 2 GB of messages (specifically, 2 GB minus 20 MB).

On all platforms, each MSMQ message can have no more than 4 MB of data.

For MSMQ servers supporting multiple MSMQ dependent clients, this limit applies to the server, not each individual client. For example, an MSMQ server supporting three dependent clients can hold no more than a cumulative total of 1.98 GB of messages for the three dependent clients.

Queue and Computer Quotas

Queue and computer quotas specify the cumulative limit for messages in a public queue or computer. The queue and computer quotas are based on size, and can be set independently. When a queue quota is reached, messages can no longer be sent to the queue or computer until one or more messages are removed from the queue. Different error messages are returned depending on the queue location. Queue quotas can be set on both public queues and journal queues.

MSMQ enforces the computer limit no matter how many queues are opened, or the cumulative queue quotas. For example, if you specify a 10 MB limit for each of the six public queues on a computer, and a 50 MB total limit for a computer, MSMQ will enforce the 50 MB computer limit before each queue quota is reached. However, each queue quota still prevents any one queue from storing more than 10 MB of messages.

Queue and computer quotas can be set only for MSMQ independent clients and servers. Queue and computer quotas set on MSMQ servers that are supporting MSMQ dependent clients apply to the server and all its dependent clients as a whole. For example, if the computer quota is 50 MB, and one dependent client creates 50 MB of messages that are not deliverable, the other dependent clients supported by the server, and the server itself, cannot send messages until the total message store drops below the quota.

Queue and computer quotas can be enabled or changed at any time programmatically, or by using MSMQ Explorer. You must have write permission for a queue or computer to change its quota.

Note Computer quotas do not apply to system queues.

For more information on changing queue and computer quotas programmatically, and for information on error messages returned when a queue or computer quota has been reached, see the MSMQ SDK. For more information on changing queue and computer quotas using MSMQ Explorer, see MSMQ Explorer Help.

Transactional Messaging

MSMQ can be used as a resource manager under the control of Microsoft Distributed Transaction Coordinator (MS DTC). Using MS DTC and the MSMQ transaction flag you can implement transaction-based applications, ensuring that message operations either succeed or fail in conjunction with other OLE-compliant applications and other MSMQ operations. For example, an MSMQ-based application might send a message and update a database in the same transaction. MS DTC ensures that both actions succeed, or neither are executed. These standard two-phase transactions are called *coordinated transactions* in the MSMQ SDK.

MSMQ also provides a small transaction coordinator that supports only one resource monitor: MSMQ. These single-phase transactions are called *transactions* in the MSMQ SDK. Because they don't have the overhead of two-phase commit transactions, these transactions are much faster than transactions coordinated by MS DTC.

Note Computers running Windows 95 and the MSMQ independent client software cannot send or receive transactional messages. However, computers running Windows 95 and the MSMQ dependent client software can send and receive transactional messages using the MS DTC proxy on the MSMQ dependent client. The MS DTC proxy is always installed on MSMQ dependent clients.

Using the MSMQ transaction flag you can:

- Transact the sending or receiving of any message with an action in any other OLE-compliant resource (for example, update an SQL database and send a message) and transact the sending or receipt of multiple messages.
- Ensure that a message is delivered only once (also called exactly-once delivery).
- Ensure all messages sent from one computer to another are delivered in order (also called in order delivery).
- Use positive or negative acknowledgements (ACKs and NACKs) to confirm messages reached or were retrieved from the destination queue (also called end-to-end confirmation).

Transactional messages are standard MSMQ messages with an additional flag set. However, transactional messages can be sent only to transactional queues. Non-transactional messages cannot be sent to transactional queues, and transactional messages cannot be sent to non-transactional queues.

Transactional messages and acknowledgements are stored in various queues:

- Regardless of the type of acknowledgment requested by the application, transactional messages are stored in the source Journal queue whenever the source computer receives a system message about the message delivery result.
- If the message is to be journaled, the transactional message stays in the Journal queue, and the transaction status is updated on the message itself. Otherwise the transactional message is automatically removed only when a positive status is received.
- If the MSMQ service on the source computer receives an in-doubt or negative status, the message stays in the Journal queue.
- Receipt Acknowledgement messages are stored in the Administration queue.
- If the message has expired, or for any other reason cannot reach the destination queue, the message is stored in the dead transaction queue on the source computer.

For more information on using transactions with your MSMQ-based applications, see the MSMQ SDK.

Programming Options

You can integrate your MSMQ-based application with the MSMQ Exchange connector, the MSMQ MAPI transport provider, and the MSMQ RPC transport. Using the ActiveX™ controls provided by MSMQ and Microsoft Internet Information Server (IIS), you can integrate your MSMQ-based application with Web pages and forms.

You can develop MSMQ-based applications on MSMQ servers and MSMQ independent clients. You cannot develop MSMQ-based applications on MSMQ dependent clients.

MSMQ Exchange Connector

Exchange users can send messages to and receive messages from MSMQ queues using the MSMQ Exchange connector. The Exchange software does not have to be running on other MSMQ dependent clients, independent clients, and servers that send messages to the Exchange clients, nor does the MSMQ software need to running on the Exchange client computers.

For information on installing and using the MSMQ Exchange connector, see Appendix B, “Installing and Configuring the MSMQ Exchange Connector.”

MSMQ MAPI Transport Provider

Any messaging application programming interface (MAPI) client can send and receive MAPI messages using the MSMQ MAPI transport provider.

The MSMQ MAPI transport provider differs from the MSMQ Exchange connector in the following ways:

- The MSMQ MAPI transport must be installed on each MAPI client to allow the client to send and receive messages over MSMQ. In contrast, the MSMQ Exchange connector needs to be installed on only one MSMQ server within a site.
- The MSMQ MAPI transport provider enables any MAPI client to send and receive MSMQ messages, whereas the MSMQ Exchange connector enables only Microsoft Exchange clients to send and receive MSMQ messages.
- Every MAPI client has its own MSMQ queue for incoming messages created by the MSMQ MAPI transport. In contrast, Microsoft Exchange clients receive messages from MSMQ through one queue used by the MSMQ Exchange connector.
- It is more difficult for a group of MAPI clients using the MSMQ MAPI transport to share addresses of MSMQ queues. Each MAPI client must store the queue addresses in its own Personal Address Book (PAB). Conversely, Microsoft Exchange clients can obtain MSMQ queue addresses from the shared Exchange Address Book.

MSMQ RPC Transport

With the MSMQ RPC transport you can use MSMQ as a reliable transport for your RPC-based applications.

To use MSMQ as an RPC transport you must install Windows NT Service Pack 3 on each Windows NT computer that will send or receive RPC packets over MSMQ. on each Windows 95 computer that will send or receive RPC packets over MSMQ, you must install DCOM 95.

MSMQ routing servers forwarding MSMQ messages that encapsulate RPC packets need not be running Service Pack 3. However, Microsoft recommends that you install Service Pack 3 on all MSMQ routing servers if you are going to use the MSMQ RPC transport.

MSMQ uses authenticated RPC. Authenticated RPC connections must be validated by a Windows NT Server domain controller. If computers access a domain controller over a slow link, RPC connections will start slowly. If the domain controller is offline, computers cannot read messages (sent using the RPC transport) from queues located on other computers.

Internet Connectivity Using ActiveX Contols

Using ActiveX controls provided by MSMQ in conjunction with active server pages, you can integrate your MSMQ-based applications with Web pages and forms.

For information on using ActiveX controls provided by MSMQ in conjunction with Microsoft Active Server, see the MSMQ SDK.

Business Scenarios

The following business scenarios show different business situations that can benefit from MSMQ-based applications, and the different programming options available when you use MSMQ.

Note The term *client* in this section refers to a computer that is running MSMQ. The differences between MSMQ dependent clients, MSMQ independent clients, and MSMQ servers affect implementation details, not business scenarios.

Floor Automation

You can use MSMQ to control the automated manufacturing of any product that is produced with a numerically controlled industrial machine that is controlled through a personal computer. In the following example, MSMQ is used to collect orders and manage the production flow by controlling the personal computers that control the industrial machines.

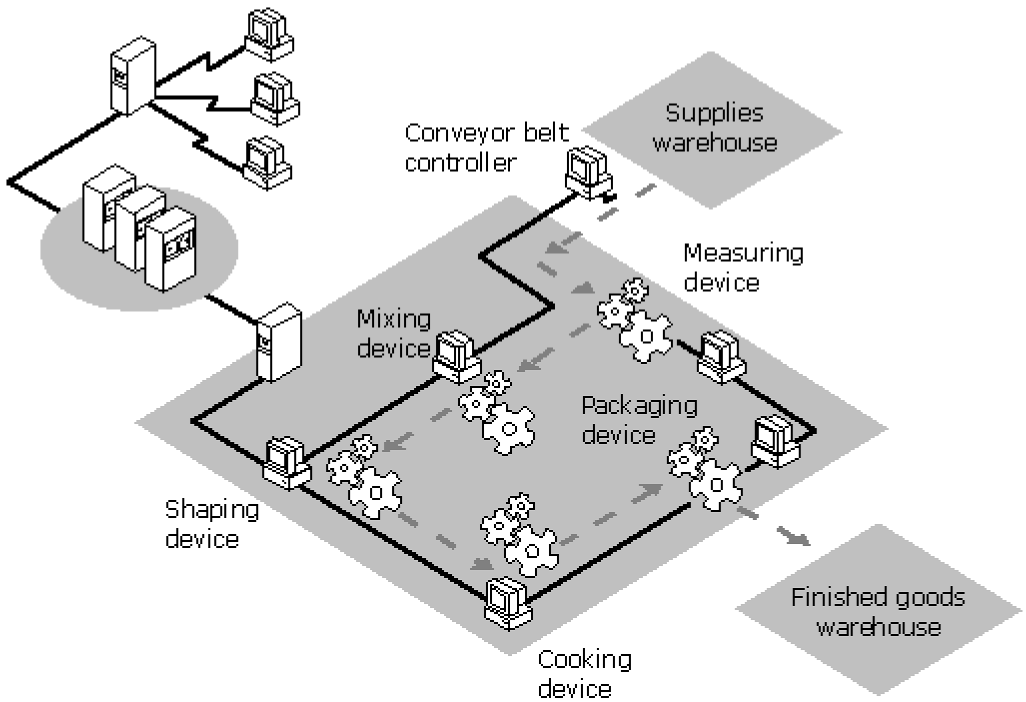


Figure 1.10

The following MSMQ features are critical to the success of floor automation:

- Guaranteed delivery – Ensures that each message is delivered, and helps ensure a reliable messaging system.
- Message prioritization – Efficient routing of messages. For example, quality assurance (QA) messages need to be delivered eventually, but shouldn't prevent the timely delivery of time-critical messages.
- Connectionless messaging – Interaction of various clients, the personal computers controlling the industrial machines, and a mainframe computer as a single system.
- Multiple platform support – Integration of personal computers and legacy systems. For example, an accounting package may reside on an MVS system.
- Performance – Control of real-time production processes; for example, controlling cooking temperature.
- Asynchronous delivery – Faster devices can continue working without waiting for slower devices to respond.

Ticket Sales

The following ticket sales scenario uses these MSMQ features:

- Internet connection – The MSMQ-based application can use the ActiveX controls provided by MSMQ, in conjunction with Microsoft Active Server and IIS, to accept orders over the Internet through HTML forms, providing security and multiple-platform support.
- Connectionless messaging – Account updating and ticket shipping can be performed asynchronously.
- Multiple platform support – Integration of personal computers and legacy systems. For example, the Venue/Show server may be a UNIX computer.

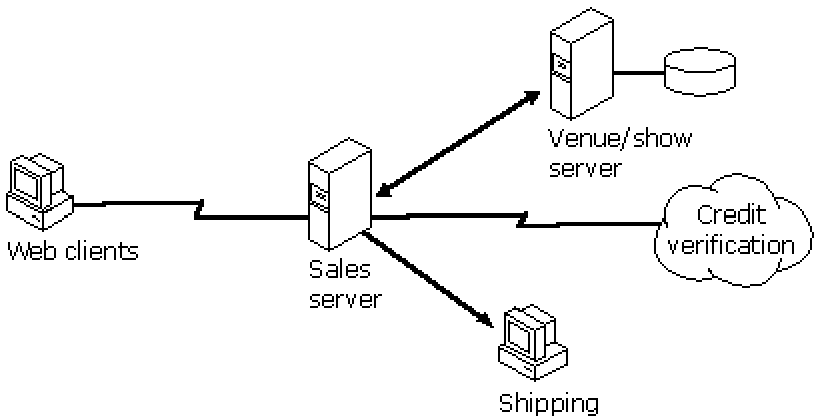


Figure 1.11

In Figure 1.11, multiple Web clients access the ticket sales server automatically. Web clients view HTML pages that provide ticket availability, pricing, and seating diagrams.

Publish and Subscribe

MSMQ can be used to simplify the development of traditional publish and subscribe applications. These applications are unique because although clients do subscribe to receive a particular type of data, they do not have to continuously request the data. Similarly, although a server application must continuously send updated data of a specified type to the client, the server does not have to continuously try to resend the data if the client if offline or unavailable. Traditional publish and subscribe applications include newswire services and financial or stock market data subscription services.

The following figure shows a publish and subscribe scenario to keep sales staff up-to-date on the status of available goods. In this scenario, an MSMQ application could also be used to take orders, and the two systems could be integrated. However, to simplify the details of the publish and subscribe scenario, this scenario covers only the dissemination of information.

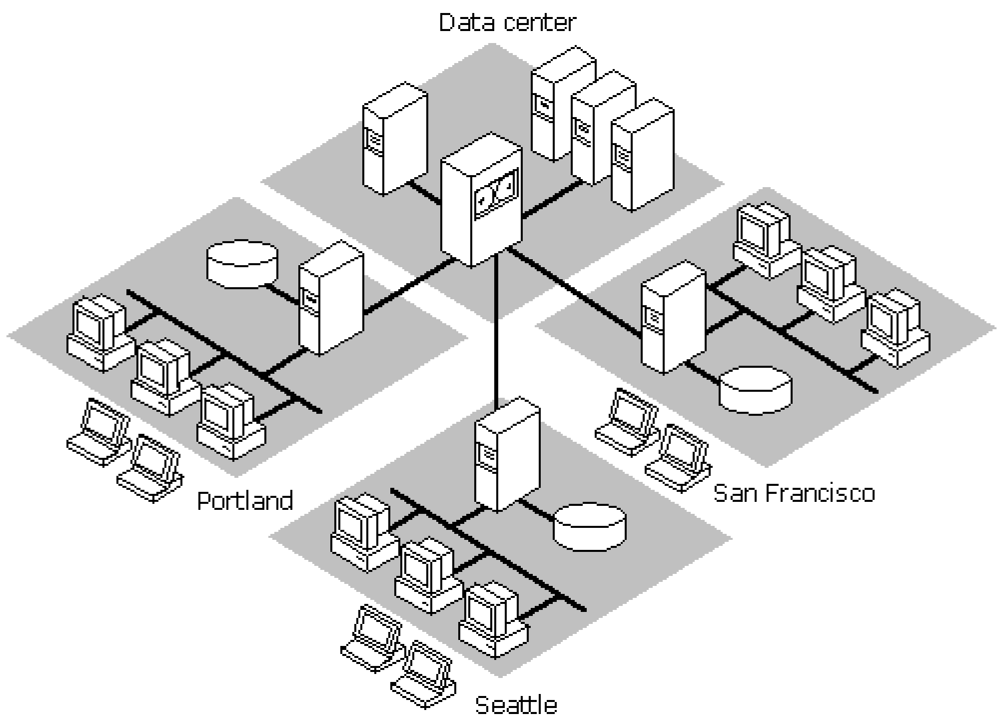


Figure 1.12

In Figure 1.12, the mobile independent clients are retrieving data while working in the field. The regional servers retrieve from the main server only those data types that the MSMQ independent clients subscribe to. Each MSMQ independent client then receives only the data it has subscribed to receive. This reduces both inter-site communication and mobile user's cellular phone use.

The following MSMQ features are critical to the success of traditional publish and subscribe applications:

- Connectionless messaging and message prioritization−Places the burden of message delivery on MSMQ. The application programmer then does not have to deal with information that expires, and is no longer useful, or information that cannot be delivered in a timely fashion.
- Security−Ensures the correct information is delivered to the correct customer, and ensures that services are not being used without payment.

Banking/ATM

Financial transactions generally require security and transaction-based programming. Additionally, automated teller machine (ATM) transactions require connectionless messaging. In Figure 1.13, each ATM has its own local database. Even if the local, regional, or headquarters database is temporarily offline, the ATM can still process transactions.

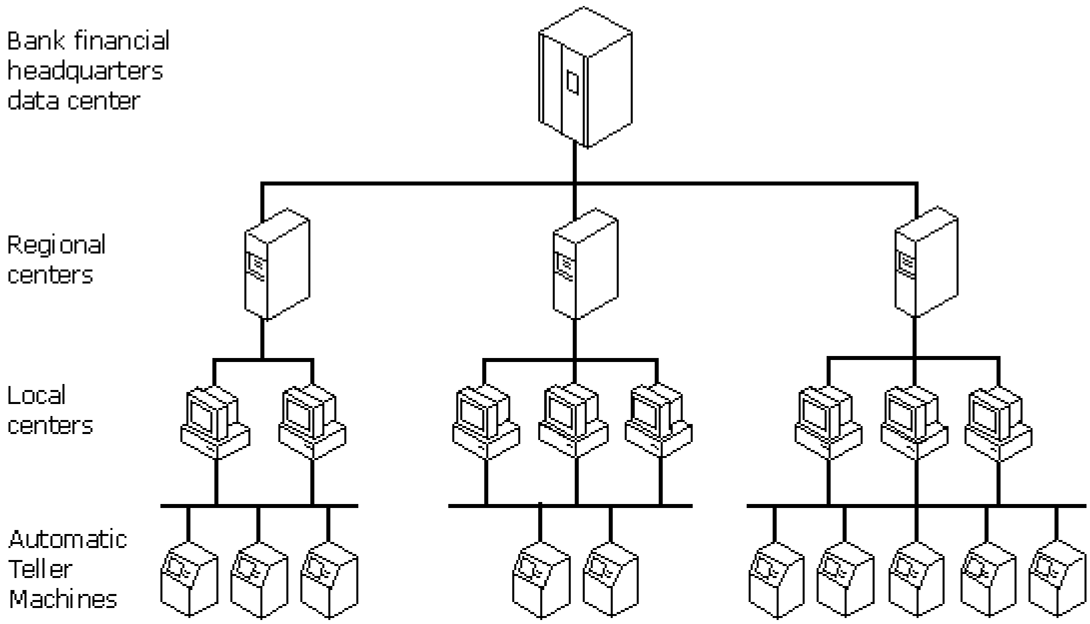


Figure 1.13

The following banking scenario uses these MSMQ features:

- Connectionless messaging – Communication between ATMs and regional centers.
- Transactional messaging – The sending and receiving of messages occurs synchronously with the updating of the financial databases, and messages are delivered only once.
- Security – Ensures the integrity of the system. Would-be thieves cannot impersonate an ATM, and transactions cannot be monitored to steal account numbers and personal identification numbers (PINs).

For more information about implementing an MSMQ-based application that uses transactional messaging, see the MSMQ SDK.

Stock Brokerage

The following stock transaction scenario uses these MSMQ features:

- Security– Message authentication, encryption, and auditing ensure a secure environment.
- Multiple platform support–Because they often must work across disparate systems, messaging allows the different platforms to communicate.
- Transactions (Exactly-once delivery)–Guaranteed message delivery ensures accurate accounting.
- Connectionless messaging–Account updating and risk assessment can be performed asynchronously.

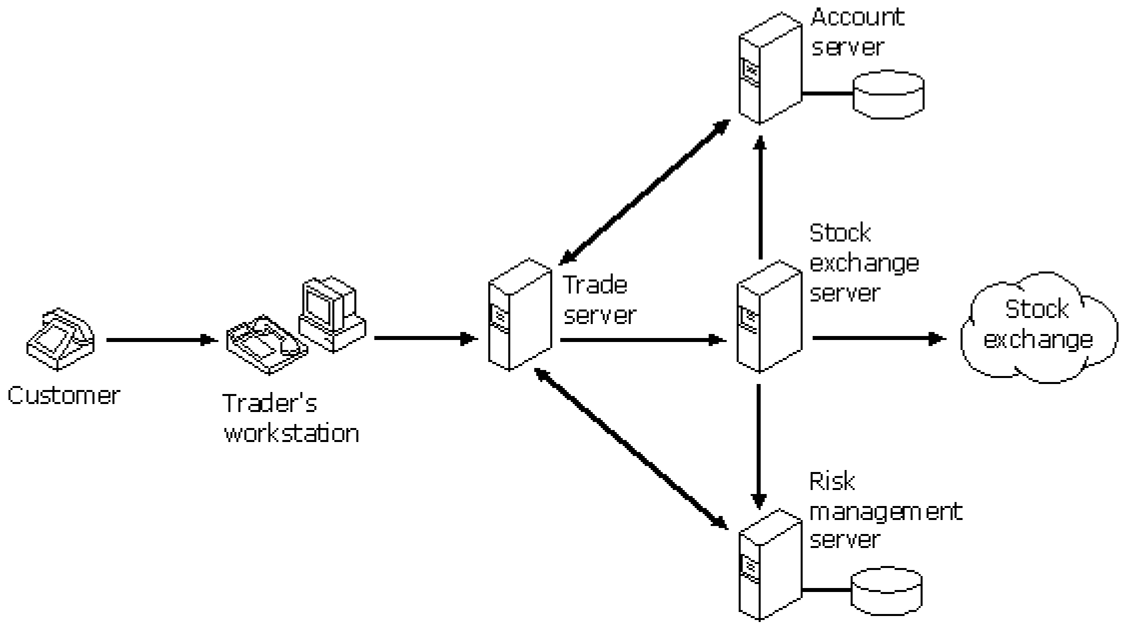


Figure 1.14

In the example shown in Figure 1.14, customers telephone traders who make trade requests on behalf of the customer. The trade server performs the following tasks in the following order:

- Verifies the transaction can be completed by querying the account server
- Verifies the transaction with the risk management server
- Places the trade request with the stock exchange server

The stock exchange server makes the stock transaction and then asynchronously updates the customer account and risk management server.

You might use the MSMQ RPC transport to access the databases, Visual Basic and the ActiveX controls provided by MSMQ to write to client applications, and C to write the core brokerage application.

Intranet Applications

With your MSMQ infrastructure in place, you can implement MSMQ-based applications to integrate internal processes. For example, you can create an MSMQ-based application to allow users to request helpdesk support from various e-mail clients, Exchange forms, and browsers, and then automatically publish resolved issues in Exchange public folders.

This MSMQ-based application might use:

- The MSMQ Exchange connector
- The MSMQ MAPI transport provider
- The ActiveX controls provided by MSMQ, in conjunction with Microsoft Active Server and Internet Information Server (IIS)

Installing MSMQ

This chapter explains the requirements and procedures for:

- Installing MSMQ dependent clients, independent clients, and servers
- Reinstalling or removing MSMQ
- Performing unattended installations
- Installing connectors and transports
- Enabling access control and authentication

For information on deploying MSMQ and MSMQ-based applications across your enterprise network, see Chapter 6, “Deploying MSMQ.”

Overview

Before installing MSMQ you must first determine your enterprise design (number of sites, number of MSMQ routing servers, and whether or not you are going to install BSCs for each site), and the hardware requirements for each server. You must also consider your enterprise's integration with Windows NT security. For small organizations with simple network topology and configuration, these choices are fairly simple. For a large organization with complex network topology and connectivity issues, these choices are more difficult. For more information on these issues, see Chapter 5, "Securing Your MSMQ Enterprise," and Chapter 6, "Deploying MSMQ."

After you plan your MSMQ enterprise network and determine individual server hardware requirements you are ready to:

- Install a PEC
- Install a PSC at each additional site
- Define site links and costs
- Install a BSC at each site, if required
- Install MSMQ routing servers, if required
- Install MSMQ dependent clients and independent clients
- Install the connectors and transports, if required

When you install a PEC you specify the enterprise name and site name. When you install a PSC, you specify the site name, define site links, and define CNs.

When you install an MSMQ server, you can have the MSMQ installation files copied to a folder called Install (typically C:\program files\msmq\install).

Note The PEC or site PSC must be online when you install other MSMQ dependent clients, independent clients, or servers.

Installing MSMQ from Installation Shares

When you install an MSMQ server you can create an installation share by copying the MSMQ installation files to a folder called Install (typically C:\program files\msmq\install). Setup shares the folder as MSMQInst. The following folders are available within the MSMQInst folder for installing other MSMQ dependent clients, independent clients, and servers.

Table 2.1

Folder	Use to Install
Client.dep	MSMQ dependent clients
Client.ind	MSMQ independent clients
Server	A PEC, PSC, BSC, or MSMQ routing server
RAS	MSMQ RAS connectivity service

If you install MSMQ dependent clients from an installation share, you are still prompted for the name of the supporting server, but a default server name is provided. When you install MSMQ independent clients from an installation share, you are not prompted to enter the computer name of a controller server (PEC, PSC, or BSC). When you install other MSMQ servers from an installation share, you are not prompted for the computer name of a controller server within the site is provided by default, although you can change it.

The computer name of the controller server provided by Setup is taken from the ControllerServer= entry in the Msmqinst.ini file on the installation share. When Setup creates an installation share, it also creates a default Msmqinst.ini file in the msmq\msmqinst\server folder. On controller servers (the PEC, PSCs, and BSC), ControllerServer= is set to the controller server's computer name. On MSMQ routing servers, ControllerServer= is set to the computer name of the controller server from which the MSMQ routing server was installed. For example, if you create an installation share when you install an MSMQ routing server from an installation share on a BSC, the ControllerServer= entry in the Msmqinst.ini file on the MSMQ routing server is set to the computer name of the BSC. If you create additional installation shares on MSMQ routing servers installed from MSMQ routing servers, the ControllerServer= entry is set to the computer name of the BSC from which the first MSMQ routing server was installed.

In most cases, you can install any server from any installation share and accept the default controller server computer name provided by Setup. The only exception occurs when you install a PSC. If you install a PSC from any installation share other than the PEC installation share, you must provide the computer name of the PEC, instead of accepting the default controller server computer name provided by Setup.

The default computer name provided by Setup for a dependent client's supporting server is taken from the SupportingServer= entry in the Msmqinst.ini file on the installation share. When Setup creates a default Msmqinst.ini file on an installation share, it sets SupportingServer= to the server's computer name (even if the server is an MSMQ routing server). So, on BSCs, PSCs, and the PEC, the ControllerServer= and SupportingServer= entries are the same. However, they differ on MSMQ routing servers.

Unattended Setup uses many other entries in the file, all of which can be controlled by an administrator. these features are documented later in this chapter in the "Unattended Setup" section.

Note By default, the person who installed the MSMQ server (the owner) has full control over the installation share, and everyone else has Read and Execute permissions. For more information on setting permissions, see Chapter 5, "Securing Your MSMQ Enterprise."

Installing MSMQ Servers

This section covers the installation of the following MSMQ servers:

- PEC
- PSCs
- BSCs
- MSMQ routing servers

MSMQ servers run only on Windows NT Server, Enterprise Edition version 4.0. The PEC, each PSC, and each BSC require Microsoft SQL Server version 6.5. You can purchase SQL Server 6.5 for each MSMQ controller server, or you can install a limited version of SQL Server 6.5 when you install each controller server.

Each MSMQ controller server must meet the connectivity requirements of the enterprise. PSCs (and the PEC) must share a CN with every other PSC. BSCs must share a CN with the PSCs or the PEC they back up. Every MSMQ independent client and MSMQ routing server must share a CN with at least one PSC/PEC or BSC within its own site.

For best performance, do not install MSMQ servers on a primary domain controller (PDC) or a backup domain controller (BDC) because PDCs and BDCs maintain and replicate the network accounts database and perform network login authentications, which are resource-intensive tasks. However, if you have a very small network in which account information rarely changes, and in which users do not log on and off frequently, you can use the same server as a domain controller and an MSMQ server.

Microsoft SQL Server Requirements

MSMQ controller servers (the PEC, PSCs, and BSCs) use a SQL Server version 6.5 database to store the MQIS. You can purchase SQL Server 6.5 for each MSMQ controller server, or you can install a limited version of SQL Server 6.5 when you install each controller server.

The MSMQ limited version of SQL Server version 6.5 is limited to a cumulative total of 100 MB in database files. The SQL master database is 30 MB, which leaves 70 MB for a maximum MQIS database of 60 MB (with 10 MB left for the logging of MSMQ database updates). To determine if the number of queue and computer objects you can store in a 60 MB MQIS database is sufficient for your enterprise, see “Choosing Between the Full and Limited versions of SQL Server” in Chapter 6, “Deploying MSMQ.” If your enterprise outgrows the limited version of SQL Server 6.5, you can purchase SQL Server version 6.5, upgrade the SQL installation on each MSMQ controller server, and increase the size of the MQIS database.

When you install an MSMQ controller server from the Windows NT Server, Enterprise Edition 4.0 Components CD, MSMQ Setup determines if an existing installation of SQL Server version 6.5 is available. If it is, MSMQ Setup uses that SQL Server installation. If SQL Server is not installed, MSMQ Setup installs the limited version of SQL Server version 6.5.

Note You cannot install the limited version of SQL Server 6.5 when you install an MSMQ controller server from an MSMQ installation share.

If you use an existing SQL Server installation, it must be configured properly for use with MSMQ. For more information on installing and configuring the retail version of SQL Server version 6.5, see the following section.

MSMQ Controller Server Computer Names and Valid SQL Identifiers

The MSMQ controller server’s computer name must be a valid SQL identifier so SQL installs properly. The first character of the computer name must be a letter or an underscore. Characters following the first character can include letters, numbers, or the symbols -, #, $, or _. No embedded spaces are allowed. Contrary to the SQL documentation, computer names with hyphens *are* supported in SQL Server version 6.5. However, if you intend to use SQL Server with other applications, further configuration steps are required. For more information, see the MSMQ release notes (\MSMQ\readme.doc on your Windows NT Server, Enterprise Edition CD).

If your computer name contains a hyphen, SQL Server Setup displays a warning message stating that the computer name is not valid. Click **Resume** to continue installing SQL Server.

If characters that SQL does not recognize are used in the computer name, rename the computer before installing MSMQ or SQL Server. For instructions on renaming computers, see Windows NT online Help. Also, if the renamed computer is a member of a Windows NT Server domain, contact your network administrator to have the new computer name added to the domain.

Installing and Configuring Microsoft SQL Server

If the 60 MB limit of the limited version of SQL Server 6.5 is not sufficient for your enterprise, you must manually install and configure the retail release of SQL Server version 6.5. You must not install SQL Server version 6.5 Service Pack 2. It is not compatible with MSMQ. You can, however, use either SQL Server version 6.5 Service Pack 1 or SQL Server version 6.5 Service Pack 3.

Note Before you start Microsoft SQL Server Setup, you must log on to the computer under an account that has administrative privileges on the computer.

To avoid ODBC configuration issues, close all ODBC applications before installing SQL Server.

By default, each SQL database has only one client (the MSMQ service running locally on that computer). Because only the local MSMQ server on the PEC, PSC, or BSC accesses the SQL database directly, you need only one client license for each SQL Server installation. However, if you use the MS Message Queue Control Panel applet to increase the number of connections between the MSMQ server and the MSMQ information store (the SQL database), you need to acquire additional client licenses. Each connection counts as one SQL Server connection, requiring one SQL client license. However, the MQIS database uses all the connections assigned to it. Therefore, when you set the MQIS connection using the MS Message Queue icon in Control Panel, you should use less than the maximum available client licenses. Otherwise, programs that require additional connections to MQIS or other SQL databases will not be able to operate while MSMQ is running.

To install the retail release of Microsoft SQL Server version 6.5

1. From the folder containing the software compatible with your hardware platform's processor architecture, run Setup.exe and complete each dialog box that appears.
2. From the **Options** dialog box, select **Install SQL Server and Utilities**, and then click **Continue**.
3. Follow the on-screen instructions. From the **Installation Options** dialog box, select **Auto Start SQL Server at boot time**. From the **SQL Executive Log On Account** dialog box, select **Install to log on as Local System account**.

For help on other dialog box options, click the Help button or press F1 to access the setup program's online Help. For more information about each option presented during installation, see your SQL Server documentation.

After you have installed SQL Server version 6.5, you must configure it for use with MSMQ

To configure the retail release of SQL Server version 6.5 for use with MSMQ

1. Click **Start**, point to **Programs**, point to **Microsoft SQL Server 6.5**, and click **SQL Setup**.
2. Click **Continue**, and then click **Continue** again.
3. Click **Set Security Options** and then click **Continue**.
4. Click **Windows NT Integrated** and then click **OK**.
5. Click the **Exit to Windows NT** button.
6. Click **Start**, point to **Programs**, point to **Microsoft SQL Server 6.5**, and click **SQL Service Manager**.
7. If the database is stopped, double-click **Start/Continue**. if the database is running, double-click **Stop**. When the database has stopped, double-click **Start/Continue**.

MSMQ Servers Setup Procedures

Before installing an MSMQ server, ensure you have the proper rights. For more information, check with your MSMQ enterprise administrator, or see Chapter 5, “Securing Your MSMQ Enterprise.” Also, if you are going to secure your MSMQ enterprise, log on to a Windows NT Server domain. If you are logged on locally when you install MSMQ, all users will have full access to your MSMQ server.

Caution If you logged on as a local user (instead of being validated by a domain controller) when you install MSMQ, your entire MSMQ enterprise may be insecure.

Installing a PEC

Before installing a PEC:

- Determine the enterprise name
- Determine the site name
- Determine all CN names
- Determine the size and location of the MQIS database

For suggestions on naming conventions, see “Naming Conventions” in Chapter 6, “Deploying MSMQ.”

For information on creating CNs and site links, see “Basic MSMQ Concepts” in Chapter 1, “Understanding MSMQ.”

For information on determining the size of the MQIS database, see “Determining the Size of the Information Store” in Chapter 6, “Deploying MSMQ.”

To install a PEC

1. Run Setup from the MSMQ\Server folder on the Windows NT Server, Enterprise Edition 4.0 Components CD.
2. Click **Continue**.
3. Confirm the default installation folder, Or click **Change Folder** to select a new one.
4. Click **Server**, **Installation Server**, or **Custom**.

   When you click **Server**, Setup installs the MSMQ server software and the MSMQ administration tools.

   When you click **Installation Server**, Setup installs the MSMQ server software, the MSMQ administration tools, and an MSMQ installation share. The installation share contains installation files for computers running Windows 95 or Windows NT on Intel-compatible computers.

   When you click **Custom**, you select specific MSMQ components. You can specify alternate folders for the MSMQ SDK, administration tools, and installation share. You can also specify which platforms will be supported by selecting **Installation Share** and clicking **Change Option**. With Custom installation, support for computers running Windows 95 and Windows NT (x86 and Alpha platforms) is selected by default.
5. Click **Primary Enterprise Controller** and then click **OK**.
6. If either the MSDTC or the MSSQLServer service is running the **Stop Running Services?** dialog box appears. Click **Yes** to continue installing MSMQ. if it is unacceptable to stop these services, click **No** and install MSMQ later.

   Setup restarts these services after reinstalling MSMQ.
7. specify the **Data Device** and **Log Device** location and size.

   You can change the size of data and log device at any time using SQL Server Enterprise Manager. However, after the MSMQ server is installed you cannot move the data or log devices. For information on calculating the appropriate data and log device size for your site, see “Determining the Size of the Information Store” in Chapter 6, “Deploying MSMQ.”
8. specify the **Enterprise** name. The enterprise name can be 126 characters long, and can contain any characters.
9. In the **Connected Networks** dialog box, click **Add**.
10. specify the **Connected Network Name** and the **Protocol** you want to use for each CN needed for your enterprise. When you’re done, click **Continue**.
11. specify the **Site Name**. The site name can be 31 characters long, and can contain any characters.
12. If necessary, pair your network adapter card addresses to the correct CNs. When you’re done, click **Continue**.

Setup prompts you for this information only if:

- You have two network adapter cards on the same protocol (for example, you have two network adapter cards using IP) and at least one CN exists for that protocol.
- Two or more CNs match the protocol being used by at least one of your installed network adapter cards (for example, two IPX CNs exist and you have one network adapter card using IPX).

Note MSMQ servers that connect to other MSMQ servers through a RAS link should not support the NetBEUI protocol over RAS.

RAS support for NetBEUI is disabled by default. To view your RAS protocol settings, run Control Panel, double-click **Network**, click the **Services** tab, click **Remote Access Service**, click **Properties**, click the appropriate modem, and then click **Network**.

Installing a PSC

Before installing a PSC:

- Determine the site name
- Determine the site link costs for the links to other sites

To install a PSC you need to know:

- The size of the MQIS database on the PEC
- The names of the CNs that the PSC will join
- The computer name of the PEC

If you install the PSC from an MSMQ installation share on the PEC, you do not need the computer name of the PEC. For more information on installing MSMQ from an installation share, see “Installing MSMQ from Installation Shares” earlier in this chapter.

The PEC must be online when you install a PSC.

In addition to MSMQ access permissions required to install PSC, you must also belong to the Administrator group on the PEC computer.

Note When you install a PSC, the MQIS database is copied from the PEC. Queues cannot be created on computers in the PEC’s site while the database is being copied. To avoid this situation when MQIS activity on the PEC MQIS is high, or the link between the PSC and PEC is slow, install the PSC from an installation share on a BSC in the PEC site.

For more information on MSMQ access permissions, see “Access Control” in Chapter 5, “Securing MSMQ.”

To install a PSC

1. Run Setup from the MSMQ\Server folder on the Windows NT Server, Enterprise Edition 4.0 Components CD, or from a shared MSMQ installation folder (typically shared as MSMQInst).

   Note You cannot install the limited version of SQL Server 6.5 when you install an MSMQ controller server from an installation share.

2. Click **Continue**.
3. Confirm the default installation folder, or click **Change Folder** to select a new one.
4. Click **Server**, **Installation Server**, or **Custom**.

   When you click **Server**, Setup installs the MSMQ software and the MSMQ administration tools.

   When you click **Installation Server**, Setup installs the MSMQ software and creates an installation share. The installation share contains installation files for computers running Windows 95 and Windows NT on Intel-compatible computers.

   When you click **Custom**, you select specific MSMQ components. You can specify alternate folders for the MSMQ SDK, administration tools, and installation share. You can also specify which platforms will be supported by selecting **Installation Share** and clicking **Change Option**. With Custom installation, support for computers running Windows 95 and Windows NT (x86 and Alpha platforms) is selected by default.
5. Click **Primary Site Controller** and then click **OK**.
6. If either the MSDTC or the MSSQLServer service is running, the **Stop Running Services?** dialog box appears. Click **Yes** to continue installing MSMQ. if it is unacceptable to stop these services, click **No** and install MSMQ later.

   Setup restarts these services after reinstalling MSMQ.

7. specify the **Data Device** and **Log Device** location and size.

   The data and log devices on a PSC should be the same size as the data and log devices on the PEC.

8. Specify the computer name of the PEC. The installation server computer name is provided by default if you are installing MSMQ from a shared installation folder.

9. Specify the **Site Name**. The site name can be 31 characters long, and can contain any characters.

10. Specify the site link costs to the other sites in the enterprise and then click **Continue**.

    Site links and site link costs are used by MSMQ to determine the available routing paths, and the cost of routing between sites. For more information on determining site link costs, see “Basic MSMQ Concepts,” and “MSMQ Routing,” in Chapter 1, “Understanding MSMQ.”

11. If necessary, pair your network adapter card addresses to the correct CNs. When you’re done, click **Continue**.

    Setup prompts you for this information only if:

    - You have two network adapter cards on the same protocol (for example, you have two network adapter cardsusing IP) and at least one CN exists for that protocol.
    - Two or more CNs match the protocol being used by at least one of your installed network adapter cards(for example, two IPX CNs exist and you have one network adapter cardusing IPX).

Note MSMQ servers that connect to other MSMQ servers through a RAS link should not support the NetBEUI protocol over RAS.

RAS support for NetBEUI is disabled by default. To view your RAS protocol settings, run Control Panel, double-click **Network**, click the **Services** tab, click **Remote Access Service**, click **Properties**, click the appropriate modem, and then click **Network**.

Installing a BSC

To install a BSC you need to know:

- The size of the MQIS database on the PSC or PEC the BSC will be supporting
- The names of the CNs that the BSC will join
- The computer name of the site controller (either the site PSC or PEC) that the BSC will support

If you install the BSC from an MSMQ installation share on the site PSC or PEC, you do not need the computer name of the site controller server (PSC or PEC). For more information on installing MSMQ from an installation share, see "Installing MSMQ from Installation Shares" earlier in this chapter.

The site controller server (PSC or PEC) must be online when you install a BSC.

In addition to MSMQ access permissions required to install BSC, you must also belong to the Administrator group on the PEC computer.

For more information on MSMQ access permissions, see "Access Control" in chapter 5, "Securing MSMQ."

To install a BSC

1. Run Setup from the MSMQ\Server folder on the Windows NT Server, Enterprise Edition 4.0 Components CD, or from a shared MSMQ installation folder (typically shared as MSMQInst).

   Note You cannot install the limited version of SQL Server 6.5 when you install an MSMQ controller server from an installation share.

2. Click **Continue**.
3. Confirm the default installation folder, Or click **Change Folder** to select a new one.
4. Click **Server**, **Installation Server**, or **Custom**.

   When you click **Server**, Setup installs the MSMQ software and the MSMQ administration tools.

   When you click **Installation Server**, Setup installs the MSMQ software and creates an installation share. The installation share contains installation files for computers running Windows 95 and Windows NT on Intel-compatible computers.

   When you click **Custom**, you select specific MSMQ components. You can specify alternate folders for the MSMQ SDK, administration tools, and installation share. You can also specify which platforms will be supported by selecting **Installation Share** and clicking **Change Option**. With Custom installation, support for computers running Windows 95 and Windows NT (x86 and Alpha platforms) is selected by default.
5. Click **Backup Site Controller** and then click **OK**.
6. If either the MSDTC or the MSSQLServer service is running, the **Stop Running Services?** dialog box appears. Click **Yes** to continue installing MSMQ. if it is unacceptable to stop these services, click **No** and install MSMQlater.

   Setup restarts these services after reinstalling MSMQ.
7. specify the **Data Device** and **Log Device** location and size.

   The data and log devices on a BSC should be the same size as the data and log devices on the PEC or PSC that the BSC is supporting.
8. Specify the computer name of the site controller (PSC or PEC). The installation server computer name is provided by default if you are installing MSMQ from a shared installation folder.
9. If necessary, pair your network adapter card addresses to the correct CNs. When you're done, click **Continue**.

   Setup prompts you for this information only if:

   - You have two network adapter cards on the same protocol (for example, you have two network adapter cards using IP) and at least one CN exists for that protocol.

- Two or more CNs match the protocol being used by at least one of your installed network adapter cards(for example, two IPX CNs exist and you have one network adapter card using IPX).

Note MSMQ servers that connect to other MSMQ servers through a RAS link should not support the NetBEUI protocol over RAS.

RAS support for NetBEUI is disabled by default. To view your RAS protocol settings, run Control Panel, double-click **Network**, click the **Services** tab, click **Remote Access Service**, click **Properties**, click the appropriate modem, and then click **Network**.

Installing an MSMQ Routing Server

To install an MSMQ routing server you need to know:

- The names of the CNs that the MSMQ routing server will join
- The computer name of the MSMQ routing server's site controller (either the site PSC or PEC)

If you install the MSMQ routing server from a BSC, PSC, or PEC installation share, you will not need the computer name of the site controller server (PSC or PEC). For more information on installing MSMQ from an installation share, see "Installing MSMQ from Installation Shares" earlier in this chapter.

The site controller server (PSC or PEC) must be online when you install an MSMQ routing server.

Unlike the PEC and each PSC and BSC, MSMQ routing servers do not require Microsoft SQL Server version 6.5.

To install an MSMQ routing server

1. Run Setup from the MSMQ\Server folder on the Windows NT Server, Enterprise Edition 4.0 Components CD, or from a shared MSMQ installation folder (typically shared as MSMQInst).
2. Click **Continue**.
3. Confirm the default installation folder, Or click **Change Folder** to select a new one.
4. Click **Server**, **Installation Server**, or **Custom**.

   When you click **Server**, Setup installs the MSMQ software and the MSMQ administration tools.

   When you click **Installation Server**, Setup installs the MSMQ software and creates an installation share. The installation share contains installation files for computers running Windows 95 and Windows NT on Intel-compatible computers.

   When you click **Custom**, you select specific MSMQ components. You can specify alternate folders for the MSMQ SDK, administration tools, and installation share. You can also specify which platforms will be supported by selecting **Installation Share** and clicking **Change Option**. With Custom installation, support for computers running Windows 95 and Windows NT (*x*86 and Alpha platforms) is selected by default.
5. Click **Routing Server** and then click **OK**.
6. If the MSDTC service is running, the **Stop Running Services?** dialog box appears. Click **Yes** to continue installing MSMQ. if it is unacceptable to stop the MSDTC service, click **No** and install MSMQ later.

   Setup restarts these services after reinstalling MSMQ.
7. Specify the computer name of the site controller (PSC or PEC). The installation server computer name is provided by default if you are installing MSMQ from a shared installation folder.
8. If necessary, pair your network adapter card addresses to the correct CNs. When you're done, click **Continue**.

   Setup prompts you for this information only if:

   - You have two network adapter cardson the same protocol (for example, you have two network adapter cardsusing IP) and at least one CN exists for that protocol.
   - Two ore more CNs match the protocol being used by at least one of your installed network adapter cards(for example, two IPX CNs exist and you have one network adapter cardusing IPX).

   Note MSMQ servers that connect to other MSMQ servers through a RAS link should not support the NetBEUI protocol over RAS.

   RAS support for NetBEUI is disabled by default. To view your RAS protocol settings, run Control Panel, double-click **Network**, click the **Services** tab, click **Remote Access Service**, click **Properties**, click the appropriate modem, and then click **Network**.

Installing MSMQ Client Software

This section covers:

- MSMQ dependent client and independent client system requirements
- MSMQ dependent client setup procedure
- MSMQ independent client setup procedure

Because Setup grants the Create Computer permission to everyone at each site, anyone can install the MSMQ dependent client and independent client software.

MSMQ Client System Requirements

The following table describes the system requirements for installing Microsoft MSMQ dependent client and independent client software.

Table 2.3

Category	**Requirement**
Operating system	Windows 95, Windows NT Workstation version 4.0 or later, or Windows NT Server version 4.0 or later. Computers running Windows 95 must have IE 3.01 or later and DCOM95 installed. Computers running Windows 95 must have the Windows 95 remote registry service installed to function as dependent clients. Computers running Windows NT Workstation or Windows NT Server must have Service Pack 2 or later installed. You must install Service Pack 3 or later on computers running Windows NT Workstation or Windows NT Server to use MSMQ as an RPC transport. For more information, see "RPC Transport" in Chapter 1, "Understanding MSMQ."
Hardware	One or more hard disks, with 1.5 MB minimum free disk space for MSMQ dependent clients or 10 MB minimum free disk space for MSMQ independent clients. Six additional megabytes are required to install the MSMQ SDK. One or more network adapter cards are required.

Note MSMQ requires Microsoft Internet Explorer (IE) 3.01 or later on Windows 95 clients because MSMQ uses IE 3.0 files and settings to support encryption and authentication. IE 3.0 must be installed even if you do not plan to use encryption and authentication.

DCOM95 allows you to use MSMQ as a reliable transport for RPC-based applications.

For ease of installation, consider placing the IE, DCOM95, and Remote Registry service files on a distribution share on your network for easy client access and installation.

You can conserve disk space on MSMQ independent clients by not installing the MSMQ SDK on computers that will not be used to develop MSMQ-based applications.

The following components are installed on MSMQ independent clients if the operating system is Windows NT:

- MS DTC
- MSMQ Explorer (if you install the administration tools)
- The MSMQ parsers for Network Monitor (if you install the MSMQ SDK)

MSMQ Dependent Client Setup Procedure

To install an MSMQ dependent client you need to know:

- The computer name of the MSMQ dependent client's supporting server

If you are installing an MSMQ dependent client from an installation share, you are still prompted for the name of the supporting server, but a default server name is provided. Setup provides the computer name specified by the SupportingServer= entry in the Mqisinst.ini file. For information on the Msmqinst.ini file default settings, see "Installing MSMQ from Installation Shares" earlier in this chapter.

Both the supporting server and the site controller server (PSC or PEC) must be online when you install an MSMQ dependent client.

Note MSMQ dependent clients and their assigned server do not have to be in the same site. However, because all of the dependent client's communication goes through the server, if the communication to the server is slow or expensive, all dependent client communication will also be slow or expensive.

Warning MSMQ dependent clients cannot be installed on a computer that also has Microsoft SQL Server, Microsoft Transaction Server, MS DTC, or any other application that installed MS DTC. If you want to install an MSMQ dependent client on a computer that has MS DTC installed, you must first remove the current MS DTC installation. For specific instructions, see "Installing an MSMQ Client on a Compuer with MS DTC Server Installed" in the MSMQ release notes (\Msmq\readme.doc on the Windows NT Server, Enterprise Edition 4.0 Components CD).

To install the MSMQ dependent client software

1. If you are installing MSMQ on a computer running Windows 95, install Internet Explorer 3.0 or later if you have not already done so. You can install Internet Explorer 3.01 from the \Clients\Win95\Win95upd\Ie3 folder on the Windows NT Server, Enterprise Edition 4.0 Base CD.
2. If you are installing MSMQ on a computer running Windows 95, install DCOM95 from the \Clients\Win95\Win95upd\Dcom folder on the Windows NT Server, Enterprise Edition 4.0 Base CD.
3. Run MSMQ dependent client Setup from the MSMQ\Client.dep folder on the Windows NT Server, Enterprise Edition 4.0 Components CD, or from a shared MSMQ installation folder (typically shared as MSMQInst).
4. Click **Continue**.
5. Confirm the default installation folder, Or click **Change Folder** to select a new one.
6. Click **Client**.
7. If you are not installing MSMQ from an installation share, you are prompted for the computer name of the dependent client's supporting computer. Specify the computer name of a PEC, PSC, BSC, or MSMQ routing server. If you are installing the dependent client from an MSMQ server, Setup uses the computer name of the server holding the installation share.
8. If you are not installing MSMQ on a computer running Windows 95, restart the computer to allow Setup to complete.

MSMQ Dependent Client Configuration Procedure

To configure an MSMQ dependent client installed on a computer running Windows 95, you must:

- Configure Windows 95 to use user level security.
- Install the Remote Registry service.
- Configure MS DTC

To configure an MSMQ dependent client installed on a computer running Windows NT, you must configure MS DTC.

To configure Windows 95 to use user level security

1. Run Control Panel and double-click **Network**.
2. On the **Access Control** page, click **User Level Security**.
3. Specify the name of the computer's domain.
4. Click **Yes** to restart your computer.

To install the Remote Registry Service

1. Run Control Panel and double-click **Network**.
2. On the **Configuration** tab, click **Add**.
3. Click **Service** and then click **Add**.
4. Click **Have Disk**.
5. Specify the \clients\win95\win95upd\remotereg folder on the Windows NT Server, Enterprise Edition 4.0 Base CD, and then click **OK**.
6. Click **Microsoft Remote Registry** and click **OK**.
7. Click **OK**.
8. If prompted, provide the Windows 95 CD.
9. Click **Yes** to shutdown and restart your computer.

For more information about the Remote Registry Service, see the Windows 95 Resource Kit (Win95rk.hlp).

To configure MS DTC

1. Run Control Panel and double-click **MS DTC**.
2. In **Current Selection**, type the computer name of the dependent client's MSMQ supporting server.
3. In **Available Network Protocols**, select the protocol being used by MSMQ on the dependent client.

MSMQ Independent Client Setup Procedure

To install an MSMQ independent client you need to know:

- The computer name of the independent client's site controller (either the site PSC or PEC) or the computer name of any BSC within the site

If you are installing an MSMQ independent client from an installation share, you are not prompted for the name of the site controller. For more information on installing MSMQ from an installation share, see "Installing MSMQ from Installation Shares" earlier in this chapter.

Each MSMQ independent client must meet the MSMQ connectivity requirements: Every independent client must share a CN with the site controller server (PSC or PEC) or at least one BSC within its own site.

The site controller server (PSC or PEC) must be online when you install the MSMQ independent client software.

Note If you logged on as a local user (instead of being validated by a domain controller) when you install MSMQ, your independent client installation will be insecure.

To install the MSMQ independent client software

1. If you are installing MSMQ on a computer running Windows 95, install Internet Explorer 3.0 or later if you have not already done so. You can download Internet Explorer from http://www.microsoft.com/ie/. It is also located on your Windows NT Server, Enterprise Edition 4.0 Base CD.
2. If you are installing MSMQ on a computer running Windows 95, install DCOM95 from the \Clients\Win95\Win95upd\Dcom95.exe folder on the Windows NT Server, Enterprise Edition 4.0 Base CD.

   You can also download DCOM95 from http://www.microsoft.com/.
3. Run MSMQ independent client Setup from the MSMQ\Client.ind folder on the Windows NT Server, Enterprise Edition 4.0 Components CD, or from a shared MSMQ installation folder (typically shared as MSMQInst).
4. Click **Continue**.
5. Confirm the default installation folder, Or click **Change Folder** to select a new one.
6. Click **Independent Client**, **Development Independent Client**, or **Custom**.
   - click **Independent Client** to install only the MSMQ software.
   - click **Development Independent Client** to install the MSMQ software and the MSMQ SDK.
   - on a computer running Windows NT, click **Custom** to select **Administration Tools** to install MSMQ Explorer. On a computer running Windows 95, click **Custom** to install only the MSMQ SDK.
7. If the MSDTC service is running, the **Stop Running Services?** dialog box appears. click **Yes** to continue installing MSMQ. if it is unacceptable to stop the MSDTC service, click **No** and install MSMQ later.

   Setup restarts these services after reinstalling MSMQ.
8. If you are not installing MSMQ from an installation share, you are prompted for the **Server Name**. Specify the computer name of the site controller server (PSC or PEC) or any BSC within the site.
9. If you are not installing MSMQ on a computer running Windows 95, restart the computer to allow Setup to complete.

Installing the MSMQ RAS Connectivity Service

To install the MSMQ RAS connectivity service you need to know:

- The names of the CNs that the MSMQ RAS connectivity service will join
- The computer name of the server's site controller (PSC or PEC).

If you install the MSMQ RAS connectivity service from a BSC, PSC, or PEC installation share, you do not need the computer name of the site controller server (PSC or PEC).

The site controller server (PSC or PEC) must be online when you install the MSMQ RAS connectivity service.

Note Windows NT RAS servers with the MSMQ RAS connectivity service installed should not support the NetBEUI protocol over RAS. The MSMQ RAS connectivity service does not support full communications with MSMQ independent clients over a RAS link when RAS support for NetBEUI is enabled.

RAS support for NetBEUI is disabled by default. To view your RAS protocol settings, run Control Panel, double-click **Network**, click the **Services** tab, click **Remote Access Service**, click **Properties**, click the appropriate modem, and then click **Network**.

To install the MSMQ RAS connectivity service

1. Run Setup from the MSMQ\Ras folder on the Windows NT Server, Enterprise Edition 4.0 Components CD, or from a shared MSMQ installation folder (typically shared as MSMQInst).
2. Click **Continue**.
    4. Confirm the default installation folder, Or click **Change Folder** to select a new one.
5. Click **RAS Connectivity Service**.

   If RAS is not installed a warning dialog box appears explaining that you must install RAS for MSMQ RAS connectivity service to function. Click **OK** and install RAS after you finish installing the MSMQ RAS connectivity service.

    6. Specify the computer name of the site controller (PSC or PEC). The installation server computer name is provided by default if you are installing MSMQ from a shared installation folder.
7. If necessary, pair your network adapter card addresses to the correct CNs. When you're done, click **Continue**.

   Setup prompts you for this information only if:

   - You have two network adapter cards on the same protocol (for example, you have two network adapter cards using IP) and at least one CN exists for that protocol.
   - Two or more CNs match the protocol being used by at least one of your installed network adapter cards (for example, two IPX CNs exist and you have one network adapter card using IPX).

   Note Window NT RAS servers running the MSMQ connectivity service should use only one IP CN. If the RAS server is configured to use more that one IP CN, automatic site and CN recognition may fail. As a result, MSMQ independent clients that connect to your MSMQ enterprise through the RAS server may have poor messaging performance, or may not be able to send or receive messages at all.

Reinstalling MSMQ

You can reinstall MSMQ dependent clients, independent clients, and servers by running Setup from the **Start** menu. You can either add or remove a component (for example, the MSMQ SDK or administration tools), or simply reinstall all files. Reinstalling MSMQ does not affect your MSMQ settings or your Windows registry (unless the change is related to the addition or removal of an MSMQ component). You must close all MSMQ-based applications before reinstalling MSMQ. To reinstall MS DTC, Setup must stop the MSDTC and MSSQLServer services. Setup restarts these services after reinstalling MSMQ.

If you want to install a different MSMQ computer type (dependent client, independent client, or server) than you currently have installed, it is best to first remove your existing MSMQ installation.

When you reinstall an MSMQ controller server, Setup preserves the existing database and does not remove the local private queues from the computer's registry.

Note If you installed the limited version of SQL Server 6.5, MSMQ Setup does not reinstall it when you reinstall MSMQ. To reinstall the limited version of SQL Server 6.5, run SQL Server Setup.

To reinstall all MSMQ files

1. Click **Start**, point to **Programs**, point to **Microsoft Message Queue**, and then click **Setup**.
2. Click **Reinstall**.
3. If either the MSDTC or the MSSQLServer service is running, the **Stop Running Services?** dialog box appears. click **Yes** to continue installing MSMQ. if it is unacceptable to stop these services, click **No** and install MSMQ later.

   Setup restarts these services after reinstalling MSMQ.

To add or remove MSMQ components

1. Click **Start**, point to **Programs**, point to **Microsoft Message Queue**, and then click **Setup**.
2. Click **Add/Remove**.
3. Select the check boxes of those components you want to add. clear the check boxes of those components you want to remove. to add or remove specific platform support provided by the installation share on MSMQ servers, select the **Installation Share** check box and then click **Change Option**.
4. Click **Continue**.
5. If either the MSDTC or the MSSQLServer service is running, the **Stop Running Services?** dialog box appears. click **Yes** to continue installing MSMQ. if it is unacceptable to stop these services, click **No** and install MSMQ later.

   Setup restarts these services after reinstalling MSMQ.

Removing MSMQ

You can remove all installed MSMQ components by running Setup. When you remove a PEC, all enterprise settings, including all the computers and queues within the enterprise, are deleted. When you remove a PSC, all site settings, including all the computers and queues within the site, are deleted. If the computer you remove is being used as an InRS or OutRS, it is deleted from InRS and OutRS lists of all independent clients using that server. When you remove MSMQ from any computer, Setup also removes all of the computer's queues in the MQIS and all local private queues from the local computer.

If a computer has failed and you cannot remove it using Setup, use MSMQ Explorer to delete the computer.

You must close all MSMQ-based applications before removing MSMQ.

Note If you installed the limited version of SQL Server 6.5, MSMQ Setup does not remove it when you remove MSMQ. To remove the limited version of SQL Server 6.5, run SQL Server Setup.

To remove MSMQ

1. Click **Start**, point to **Programs**, point to **Microsoft Message Queue**, and then click **Setup**.
2. Click **Remove All**.
3. Click **Yes** To confirm you want to remove MSMQ.
4. click **Yes** To confirm you want to permanently delete the storage files created by MSMQ.
5. If you are removing a PSC or PEC, click **Yes** to confirm that you want to remove the site or enterprise.

Unattended Setup

You can use MSMQ unattended setup to install all types of dependent clients, independent clients, and servers (including a PEC). MSMQ unattended setup is controlled using a file called Msmqinst.ini. The Msmqinst.ini file can be stored in either the source MSMQ installation folder on the server or in the Windows folder (typically c:\Win95 or c:\Winnt) on the destination computer. If both files exist when you run unattended setup, Setup uses the Msmqinst.ini file in the Windows folder. When MSMQ servers are installed with an installation share, a default Msmqinst.ini file is created in the installation share folder.

The Msmqinst.ini file specifies default options for all computers in the [Common Parameters] section. Optionally, other sections named for specific computers specify settings that override the common settings. For example, to install MSMQ on twenty computers using unattended setup, install a PEC with an installation share, modify the Msmqinst.ini file on the PEC's installation share, and then run unattended setup on each computer.

Because the site controller's computer name is required to install every independent client and server, it is easiest to use separate Msmqinst.ini files for each site. You can use a single Msmqinst.ini file for multiple sites, but you have to create entries for every computer that is not in the default site.

Unattended Setup Modes

Unattended setup has two modes: minimal interaction mode, and no interaction mode. The minimal interaction mode is specified using the /q switch (*q* is short for *quiet*). The no interaction mode is specified using the /qt switch (*qt* stands for *totally quiet*).

In minimal interaction (quiet) mode, the main setup window is displayed on the desktop, as well as the copy progress indicator and a dialog box that indicates whether Setup succeeded or failed. You must click **OK** to clear this dialog box. If Setup cannot resolve all issues based on the Msmqinst.ini files, Setup displays a message box to alert the user. The user can then correct the problem.

In contrast, Setup does not display any text or dialog boxes on the desktop in no interaction (totally quiet) mode. If a problem occurs, Setup writes an error in the Msmqinst.log file in the Windows folder (typically c:\Win95 or c:\Winnt) and quits.

Unattended Setup Parameters

When you run unattended setup, you can specify the following command line parameters:

- */q* or */qt*
- */b#*

For example, the following command can be used to install MSMQ:

```
setup /qt /b1
```

The */q* or */qt* parameter specifies quiet mode (*/q*) or totally quiet mode (*/qt*). For more information on unattended setup modes, see the previous section, “Unattended Setup Modes.” If you do not specify */q* or */qt*, Setup does not run in unattended mode.

The */b#* switch corresponds to the button order in the Setup Installation Type dialog box. The number of available options (*/b1*, */b2*, or */b3*) depends on which type of dependent client, independent client, or server you are installing and, in the case of independent clients, which platform you are installing on. However, in all cases, */b* is the default−It is used if the */b#* parameter is not specified.

To run an unattended reinstall, use the following command:

```
setup /q[t] /r
```

To run an unattended uninstall, use the following command:

```
setup /q[t] /u
```

Note Unattended uninstall automatically removes your MSMQ data files.

Understanding the /b# Parameter

For the */b#* parameter, the number of available options (*/b1*, */b2*, or */b3*) and their function depends on which type of dependent client, independent client, or server you are installing and, in the case of independent client, which platform you are installing on. However, in all cases, */b1* is the default−It is used if the */b#* parameter is not specified.

When using unattended setup to install the MSMQ dependent client software, the */b1* parameter maps to the only setup button: **Client**.

When using unattended setup to install MSMQ independent client software, the */b#* parameter maps to the **Independent Client** and **Development Independent Client** buttons.

- Use */b1* (**Independent Client)** to install the MSMQ independent client software without the MSMQ SDK.
- Use */b2* (**Development Independent Client)** to install the MSMQ independent client software and the MSMQ SDK.

Note You can also specify */b3* when installing the MSMQ independent client software on computers running Windows NT Workstation or Windows NT Server. The */b3* parameter maps to the **Custom** button. This parameter has the same effect as specifying */b2*.

When using unattended setup to install a PEC, PSC, BSC, or MSMQ routing server, the */b#* parameter specifies **Server**, **Installation Server**, or **Custom**.

- Use */b1* (**Server)** to install the MSMQ server software and administration tools.
- Use */b2* (**Installation Server)** to install the MSMQ server software, administration tools, the MSMQ SDK, and an MSMQ installation folder for computers running Windows 95 and Windows NT (Intel-compatible computers only).
- Use */b* (**Custom)** to install the server software, administration tools, MSMQ SDK, and an MSMQ installation folder for Windows 95 computers and all supported Windows NT platforms (x86, Alpha, and PowerPC).

When using unattended setup to install the MSMQ RAS connectivity service, the */b1* parameter maps to the only button: **RAS Connectivity Service**.

Msmqinst.ini File Entries

The [Common Parameters] section in the Msmqinst.ini file specifies default options for all computers. Optionally, other [computer_name] sections for specific dependent client, independent client, and server computers specify settings that override the settings in the [Common Parameters] section for those computers and set settings for specific server types (for example the PSCs and BSCs).

Most of the Msmqinst.ini entries can appear in the [Common Parameters] section, a specific computer section, or both. Exceptions are noted in the following sections.

ControllerServer=

Setup searches for this entry when you install MSMQ independent clients and servers. This entry specifies the computer name of the site controller (PSC or PEC) for the independent client or server being installed. MSMQ dependent client setup does not use this entry because it gets the dependent client's controller server name from the supporting server (specified by SupportingServer=).

DataDevice=

Setup searches for this entry when you install a PEC, PSC, or BSC. This entry specifies both the location and size of the MQIS data device using the following format: <location>,<size>. For example:

```
DataDevice=c:\msmq,50
```

The location specifies the drive and folder for the data device. The size is specified in megabytes, and must be an integer−Decimal numbers are not allowed. For example, DataDevice=c:\msmq,50.5 is not valid

EnterpriseName=

Setup searches for this entry when you install a PEC. This entry specifies the enterprise name. The enterprise name can be up to 126 characters long, and can contain any characters.

SupportingServer=

Setup searches for this entry when you install an MSMQ dependent client. This entry specifies the computer name of the dependent client's supporting server. MSMQ dependent client setup doesn't automatically use this computer name, it is provided as the default. When Setup creates the default Msmqinst.ini file on an installation share, the SupportingServer= entry in the [Common Parameters] section is set to the computer name of the server. You can specify a different supporting server computer name for specific dependent clients by creating [computer name] sections for each dependent client and adding additional SupportingServer= entries.

IPAddresses=

Setup searches for this entry when you install MSMQ independent clients and servers. If you specify only a CN name, Setup associates the CN with all available IP addresses on the computer on which MSMQ is being installed. If you specify both a CN name and IP address, Setup associates only that specific IP address with the CN name. For example:

```
IPAddresses=157.55.86.198,Seattle IP
```

You can specify multiple IP Address/CN pairs by separating them with semicolons (;). For example:

```
IPAddresses=157.55.87.198,New York IP; 157.55.86.198,Seattle IP
```

Because Setup processes IPAddresses= entries from left to right, specific IP address/CN pairs that appear to the right of a CN name that appears without a CN address override previous settings. For example, on a computer with three IP addresses, the following setting sets two of the addresses to **Seattle IP**, and a third, specific address, to **New York IP**:

```
IPAddresses=Seattle IP;157.55.86.198,New York IP
```

IPCNs=

Setup searches for this entry when you install the PEC. This entry creates an IP connected network for each name specified. Specify multiple CNs by separating the CN names with semicolons (;). For example:

```
IPCN=Seattle IP;New York IP
```

IPXAddresses=

Setup searches for this entry when you install MSMQ independent clients and servers. If you specify only a CN name, Setup associates the CN with all available IPX addresses on the computer on which MSMQ is being installed. If you specify both a CN name and IPX address, Setup associates only that specific IPX address with the CN name. For example:

```
IPXAddresses=00002402.00AA00BDCF04,Seattle IPX
```

Specify multiple IPX Address/CN pairs by separating them with semicolons (;). For example:

```
IPXAddresses=00002402.00AA00BDCF04,New York IPX;
00002402.00AA00BDCF04,Seattle IPX
```

Because Setup processes IPXAddresses= entries from left to right, specific IPX address/CN pairs that appear to the right of a CN name that appears without a CN address override previous settings. For example, on a computer with three IPX addresses, the following setting sets two of the addresses to **Seattle IPX**, and a third, specific address, to **New York IPX**:

```
IPXAddresses=Seattle IPX;00002402.00AA00BDCF04,New York IPX
```

IPXCNs=

Setup searches for this entry when you install the PEC. This entry creates an IPX connected network for each name specified. Specify multiple CNs by separating the CN names with semicolons (;). For example:

```
IPXCN=Seattle IPX;New York IPX
```

LogDevice=

Setup searches for this entry when you install a PEC, PSC, or BSC. This entry specifies both the location and size of the MQIS log device using the following format: <*location*>,<*size*>. For example:

```
LogDevice=c:\msmq,8
```

The size is specified in megabytes, and must be an integer–decimal numbers are not allowed. For example, LogDevice=c:\msmq,8.5 is not valid.

ServerType=

Setup searches for this entry when you install a PEC, PSC, BSC, or MSMQ routing server. This entry specifies which type of server to install. There are four valid settings: PEC, PSC, BSC, and RS.

SiteLinks=

Setup searches for this entry when you install a PSC. This entry specifies the link cost for each site using the following format: <*sitename*>,<*linkcost*>. Separate multiple sites with a semicolon (;). For example, to set the site link costs for two sites called NewYork and Seattle to 10 and 5, respectively, use the following entry:

```
SiteLinks=NewYork,10;Seattle,5
```

If no site name is specified but a site link cost is specified, the new site is linked to all existing sites with the specified number as the link cost.

Using a combination of the two previous examples, you can specify a default site link cost, and then specify specific site link settings for some sites that override the default. For example, the following SiteLinks= entry sets a default site link cost of 5 for the site links to all existing sites except the New York site–that site link cost is set to 10:

```
SiteLinks=5; NewYork,10
```

SiteName=

Setup searches for this entry when you install a PEC or PSC. This entry specifies the name of the site to be created. The site name can be up to 31 characters long, and can contain any characters.

StopServices=

Setup searches for this entry when you install an MSMQ independent client or server. This entry specifies whether Setup can stop and restart the DTC and SQL Server services if either service is running. The two valid settings are Allow and Disallow. If StopServices=Disallow, Setup displays an error message stating that it cannot continue. The error message lists the currently running services that must be stopped for Setup to install MSMQ. Setup then exits. Setup assumes StopServices=True if it cannot find the entry

Semicolons (;)

Setup ignores Msmqinst.ini file entries that begin with a semicolon (;). Use them to comment your Msmqinst.ini files.

An Unattended Setup Example

This section contains a sample Msmqinst.ini file created for a hypothetical company called Volcano Coffee. Volcano Coffee plans to implement MSMQ using the following criteria:

- One PEC in Seattle. The enterprise will be called Volcano Coffee, the PEC computer name will be \\VolcanoPEC, and the site name will be Seattle.
- A second site in Portland will be named Portland. This PSC will be installed on a computer called \\PortlandPSC.
- One BSC at each site.
- One MSMQ routing server at each site.
- 50 MSMQ independent clients at each site.
- The MQIS data device on the PEC, the Portland PSC, and both BSCs will be 50 megabytes. The MQIS log device on those computers will be 8 megabytes. For better performance, each server will have two physical drives with the data device on drive C and the log device on drive D.
- The independent clients in the Seattle site use the IP protocol, and will therefore use an IP CN called Volcano IP. The independent clients in the Portland site use the IPX protocol, and will therefore use an IPX CN called Volcano IPX. The MSMQ servers in both sites use both IP and IPX protocols, and will therefore belong to both CNs.

Because Volcano Coffee has two sites, two Msmqinst.ini files are needed. One Msmqinst.ini file could be used, but you would need to create an entry for each computer that is not in the default site.

The following Msmqinst.ini file will be used to install MSMQ on all computers in the Volcano Coffee Seattle site.

```
[Common Parameters]
ControllerServer=volcanopec
DataDevice=c:\msmq,50
LogDevice=d:\msmq,8
IPAddresses=Volcano IP
IPXAddresses=Volcano IPX

; Setup uses the following section to install the Volcano PEC
[VolcanoPEC]
ServerType=PEC
EnterpriseName=Volcano Coffee
IPCNs= Volcano IP; Test Lab IP
IPXCNs= Volcano IPX
SiteName=Seattle

; Setup uses the following section to install the BSC for the
; Volcano PEC
[SeattleBSC]
ServerType=BSC

; Setup uses the following section to install
; an MSMQ routing server in the Seattle site
; Because the server has two network adapters,
; the IPAddresses entry is needed to pair the IP
; address with the available CNs
[SeattleRS]
ServerType=RS
IPAddresses=157.55.86.198,Seattle IP; 157.55.86.199,Test Lab IP
```

```
; Setup uses the following section when you install
; MSMQ for a lab user that does not use the default CN.
[peter]
IPAddresses= Test Lab IP
```

The following Msmqinst.ini file will be used to install MSMQ on all computers in the Volcano Coffee Portland site.

```
[Common Parameters]
ControllerServer=portlandpsc
DataDevice=c:\msmq,50
LogDevice=d:\msmq,8
IPAddresses=Volcano IP
IPXAddresses=Volcano IPX

; Setup uses the following section to install the Portland PSC
[PortlandPSC]
ControllerServer=volcanopec
ServerType=PSC
SiteName=Portland
SiteLinks=Seattle,10

; Setup uses the following section to
; install the BSC for the Portland PSC
[PortlandBSC]
ControllerServer=portlandpsc
ServerType=BSC

[PortlandRS]
ServerType=RS
```

Installing Transports

The MSMQ RPC transport, the MSMQ Exchange connector, and the MSMQ MAPI provider are not installed by Setup. These should be installed on your MSMQ enterprise as needed.

For information on installing and using the MSMQ Exchange connector, see Appendix B, “Installing and Configuring the MSMQ Exchange Connector.”

Installing the MSMQ RPC Transport

To use MSMQ as an RPC transport on computers running Windows NT, you must install Windows NT Service Pack 3 on each computer that will send or receive RPC packets over MSMQ. To use MSMQ as an RPC transport on computers running Windows 95, you must install DCOM 95 on each computer that will send or receive RPC packets over MSMQ.

MSMQ servers (PEC, PSCs, BSCs, and MSMQ routing servers) forwarding MSMQ messages that encapsulate RPC packets need not be running Service Pack 3. However, it is recommended that you install Service Pack 3 on all MSMQ servers if you are going to be using the MSMQ RPC transport.

Installing the MSMQ MAPI Transport

Any computer running MSMQ can install the MSMQ MAPI transport provider by running Setup from the Msmq\Mqmail\Mapixp\Setup folder on the Windows NT Server, Enterprise Edition 4.0 Components CD—no user interaction is required.

Removing Transports

To remove the MSMQ RPC transport from a computer running Windows NT Workstation or Windows NT Server, uninstall Service Pack 3.

For more information on removing an MSMQ Exchange connector, see Appendix B, "Installing and Configuring the MSMQ Exchange Connector."

Use the following procedure to remove the MSMQ MAPI provider:

To remove the MSMQ MAPI transport provider

- Run Setup from the \MSMQ\Mqmail\Mapixp\Setup folder on the Windows NT Server, Enterprise Edition 4.0 Components CD.

Using Control Panel to Enable Access Control and Authentication

If queues in your enterprise require access control, each user must register their certificates in the MQIS database so MSMQ can map the SIDs of users who are not logged on to their registered certificates.

To register your certificates with MSMQ

1. Run Control Panel and double-click the **MS Message Queue** icon.
2. On the **Security** tab click **Register**.
3. Click a certificate and then click **OK**.

Managing Your MSMQ Enterprise

Because you manage most of your MSMQ enterprise with MSMQ Explorer, this chapter begins with an overview of MSMQ Explorer. The chapter then covers conceptual information relating to managing MSMQ. Procedures are included for administration tasks that require the use of Control Panel, or SQL server administration tools. All administration procedures for using MSMQ Explorer are in MSMQ Explorer Help.

For information on monitoring your MSMQ enterprise, see Chapter 4, “Monitoring Your MSMQ Enterprise.”

Managing MSMQ Using MSMQ Explorer

MSMQ Explorer displays a logical view of your MSMQ enterprise network. Using MSMQ Explorer, you can view and modify the properties of the following objects:

- The Enterprise
- Sites
- Connected Networks
- Computers
- Queues

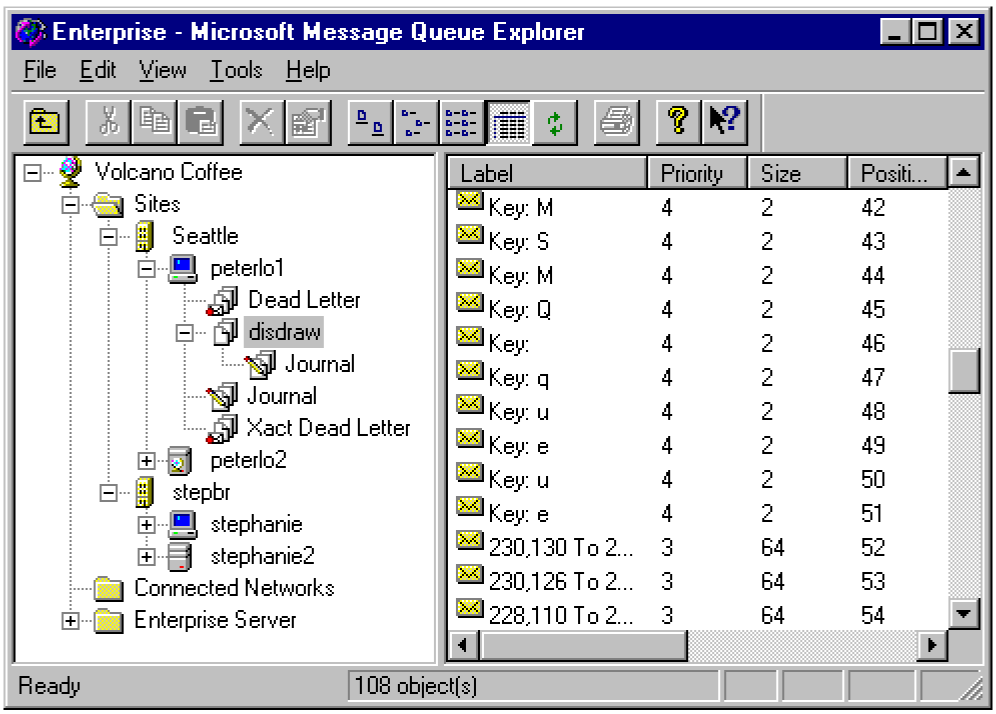


Figure 3.1

To set properties on the enterprise, a site, a CN, a computer, or a queue, right-click the object and then click **Properties**. To see the tasks that can be performed from MSMQ Explorer, click **Help Topics** on the MSMQ Explorer **Help** menu, double-click the **How to** book on the **Contents** tab, and then double-click the other books.

MSMQ Explorer displays only those computers running the MSMQ service (dependent clients, MSMQ Routing servers, BSC, PSCs, and the PEC). MSMQ remote dependent clients are not displayed. However, MSMQ connector servers (supporting foreign CNs, foreign computers, and foreign queues) are displayed. MSMQ Explorer uses a variety of icons to represent the different types of computers, queues, and messages. To see a table that defines the icons in MSMQ Explorer, click Help Topics on the MSMQ Explorer **Help** menu, double-click the **Overview** book on the **Contents** page, and then double-click **MSMQ Explorer Icons**.

Information displayed in MSMQ Explorer may not be current because of MQIS replication delays. Changes you make from within MSMQ Explorer may not be displayed until replication takes place. For example, when you add a queue to a computer in site C from a computer in site B, you don't see the change in MSMQ Explorer until the MQIS change has been replicated from site C to site B, regardless of which site contains the PEC. The MSMQ Explorer display is not updated automatically. To update the display, press F5.

Because MSMQ Explorer can be installed on MSMQ independent clients running on Windows NT, you can administer your MSMQ enterprise from both independent clients and servers, as long as you have the required permissions. MSMQ Explorer cannot be installed on computers running Windows 95, or on MSMQ dependent clients running on Windows NT.

Note To view the contents of a queue from MSMQ Explorer, you must have Peek Message and Get Properties permissions for the queue. To view the contents of a computer's journal queue, you must have Peek Journal permissions for the computer. To view the security permissions for an enterprise, site, CN, computer, or queue, you must have Get Permissions permissions for that object.

For more information on queue permissions, see "Setting Permissions for a Queue" in Chapter 5, "Securing Your MSMQ Enterprise."

Viewing Private Queues

Viewing private queues places an additional load on the network so they are only displayed when you right-click on a computer and then click **Show Private Queues**.

When you update the MSMQ Explorer display by pressing the F5 key, private queues are no longer displayed. However, requests to view the contents of additional private queues do not affect the private queues already displayed.

To view private queues for more than one computer within the same site, select the site that contains the computers, select multiple computers in the right pane of MSMQ Explorer, right-click on a computer, and click **Show Private Queues**. To select a contiguous block of computers, hold the SHIFT key while selecting the computers. To select non-contiguous computers, hold the CTRL key while selecting the computers.

Note Viewing the private queues of many computers generates significant network traffic in large enterprises. View the private queues of all computers only when necessary, and preferably when the network load is low.

Changing the MSMQ Explorer Display

MSMQ Explorer displays different columns in the right pane when you click:

- The **Sites** folder
- A specific site
- A computer
- A queue

Although not all columns are displayed by default, you can choose which columns to display, and in which order they are displayed. For specific instructions on how to change which columns are displayed, see MSMQ Explorer Help.

Viewing Site Properties

Site properties are displayed when you click the **Sites** folder. You can configure MSMQ Explorer to display the following site properties in the right pane:

- ID
- Server
- Site Pathname

All three are displayed by default.

Viewing Computer Properties

Computer properties are displayed when you click a specific site. You can configure MSMQ Explorer to display the following computer properties in the right pane:

Computer Pathname	Modified
Created	Quota
Foreign Computer	Service
ID	Type
Journal Quota	

Computer Pathname, Service, Created, Modified, and Foreign Computer are displayed by default.

Viewing Queue Properties

Queue properties are displayed when you click a specific computer. You can configure MSMQ Explorer to display the following queue properties in the right pane:

Authenticated	Journal	Privacy Level
Base Priority	Journal Quota	Quota
Created	Label	Transaction
Encryption Level	Modified	Type
ID	Pathname	

Pathname, Label, Journal, Created, and Modified are displayed by default.

Viewing Message Properties

Message properties are displayed when you click a specific queue. You can configure MSMQ Explorer to display the following message properties in the right pane:

Admin Queue	Delivery Mode	Label Length	Sender ID Length
Admin queue name len	Destination Queue	Message ID	Sent
Arrived	Destination queue name len	Position In Queue	Size
Authenticated	Encrypted	Priority	Source Computer
Body	Encryption Algorithm	Response Queue	Tracked
Class	Hash Algorithm	Response queue name len	
Correlation	Label	Sender ID	

Label, Priority, Message ID, Position In Queue, and Size are displayed by default.

Note When you click a computer, the queue information in the right pane is not sorted until you click the column heading you want to sort by.

Managing MSMQ Using Other Tools

The following administrative tasks may need to be performed by MSMQ dependent clients, independent clients, and/or MSMQ Servers. Most of these tasks can be accomplished using the **MS Message Queue** icon in Control Panel.

Note

All administration procedures for using MSMQ Explorer are in MSMQ Explorer Help.

Changing your designated MSMQ Server

MSMQ dependent clients require synchronous access to an MSMQ server. The dependent client's designated supporting server is specified when you install the dependent client. Use the **MS Message Queue** icon in Control Panel to change your supporting server.

To change your supporting MSMQ server

1. double-click the **MS Message Queue** icon In Control Panel.
2. On the **MSMQ Client** tab, specify you new supporting server name.

Renewing Encryption Keys

MSMQ supports encryption through the use of public and private keys. It is a good idea to periodically renew your encryption keys (perhaps once a year. You should also renew you encryption keys any time you suspected that they my be compromised.

To renew an encryption key

1. double-click the **MS Message Queue** icon In Control Panel.
2. On the **Security** tab, click **Renew Encryption Key**.
3. Click **OK**.
4. Click **Restart Windows Now**.

Managing Certificates in the MQIS

If queues in your enterprise require access control, each user must register their certificates in the MQIS database so MSMQ can map the SIDs of users who are not logged on to their registered certificates.

These procedures apply to MSMQ dependent clients, independent clients, and servers.

To register a certificate

1. double-click the **MS Message Queue** icon In Control Panel.
2. On the **Security** tab, click **Register**.
3. Click a certificate and then click **OK**.

To view a certificate

1. double-click the **MS Message Queue** icon in Control Panel.
2. On the **Security** tab, click **View**.
3. Click a certificate and then click **View Certificate**.

To remove a certificate

1. double-click the **MS Message Queue** icon in Control Panel.
2. On the **Security** tab, click **Remove**.
3. Click a certificate and then click **Remove**.

Renewing Your Internal Certificate

MSMQ dependent clients, independent clients, and servers can renew their internal certificates (used for authentication) using the **MS Message Queue** option in Control Panel. You should renew you internal certificate when you:

- Rename your computer
- Remove your computer from one site and install it in another
- Remove your computer from one enterprise and install it in another
- Log on as more than one domain user under Windows 95

Windows 95 clients can specify which network should be their primary logon network (using the Control Panel **Network** applet). If you choose Windows Logon as your primary logon, Windows caches passwords you specify for network resources. If you send authenticated messages and log on to a Windows NT Server domain as more than one domain user, you must renew your internal certificate each time you log on as a different user.

For more information on renaming a computer running MSMQ, see “Renaming a Computer Running MSMQ” later in this chapter.

To renew your internal certificate

1. double-click the **MS Message Queue** icon in Control Panel.
2. On the **Security** tab, click **Renew Internal Certificate**.
3. A warning appears stating that undelivered messages that you’ve already sent using the current internal certificate will be rejected as not authentic. If this is acceptable, click **Yes**. if it is not acceptable, click **No** and renew your internal certificate some other time.

Preparing to Travel to a New Site

MSMQ independent clients that connect to a new site cannot receive messages that are sent to them while they are disconnected from the network unless they specify the new site before disconnecting from the network.

To specify a new site before disconnecting from the network

1. double-click the **MS Message Queue** icon in Control Panel.
2. On the **Client** tab, in **New Site**, type the name of the site in which you are going to connect.

Securing Controller Server Communications

When you secure controller server communications, all communication from MQIS servers to MSMQ clients (both independent and dependent) and MSMQ routing servers is secured. This ensures unauthorized persons cannot install software that provide false information in the place of a real MSIQ server. By default, MSMQ does not use secure communications.

For more information on securing controller server communication, see "Securing Communication with Controller Servers" in Chapter 5, "Securing Your MSMQ Enterprise."

To secure controller server communications

1. Install IIS version 2.0 or later on each controller server.
2. Obtain one or more certificates from a certification authority. You must have one certificate for each controller server that will support secured MQIS access.
3. Use the IIS Key Manager utility to install and configure the certificates on the controller servers.
4. on each MSMQ client that will support secure MQIS access, double-click the **MS Message Queue** icon in Control Panel, and, On the **Security** tab, select **Use only secured connections when communicating with MSMQ controller server**.

To install and configure a certificate for authenticating MQIS access

1. Click **Start**, point to **Programs**, point to **Microsoft Internet Information Server (Common)**, and click **Key Manager**.
2. Click **MSMQ**, and then click **Create New Key** on the **Key** menu.
3. In **Password**, specify a password to encrypt the key.
4. Type the information required in **Distinguishing Information**. For **Common Name**, specify the name of the computer running Windows NT Server.
5. In **Request File**, specify the path to the certificate request file that Key Manager will create, or accept the default.
6. click **OK**.
7. Retype the password specified in Step 3, and click **OK**.
8. Provide your e-mail address and phone number, and then click **OK**.
9. Send the certificate request file to a certification authority.
10. When you receive your key from the certification authority, select the corresponding key in Key Manager, and then click **Install Key Certificate** on the **Key** menu.

Using the Same Certificate for IIS and MSMQ

Although you can install multiple certificates for use with IIS, you can install only one certificate for use with MSMQ (for the purpose of securing MQIS communication). If one or more certificates is already installed for use with IIS, and the common name of one of the certificates matches the name of the computer running Windows NT, you can run Key Manager and copy and paste, or you can cut and paste, one certificate from the WWW node to the MSMQ node. If you copy and past the certificate, it will be used by IIS and MSMQ. However, if the certificate's common name matches the server's domain name system (DNS) name instead of its Windows NT computer name, the certificate cannot be used with MSMQ.

To enable or disable secure controller server communications on a client

1. Run **Control Panel**, and double-click **MS Message Queue**.
2. On the **Security** tab, select or clear **Use only secured connections when communicating with MSMQ controller server**.

To view or change the IE certification authority configuration

1. On the IE **View** menu, click **Options**.
2. On the **Security** tab, click **Sites**.

To view or change the MSMQ certification authority configuration on MSMQ clients and MSMQ routing servers

1. Run **Control Panel**, and double-click **MS Message Queue**.
2. On the **Security** tab, click **Certification Authorities**.

To add a certification authority to the IE and MSMQ certification authority configuration

1. Click **Start** and then click **Run**.
2. Type the path to the .crt file provided by the certification authority and then click **OK**.

   Or, click **Browse** and specify the .crt file, click **Open**, and then click **OK**.

3. Click either **Accept and enable this site certificate** Or **Accept this certificate but do not enable it**, depending on whether you want IE to trust the certification authority.
4. Run **Control Panel**, and double-click **MS Message Queue**.
5. On the **Security** tab, click **Certification Authorities**.
6. Select or clear the check box next to the name of the certification authority, depending on whether you want MSMQ to trust the certification authority.

Maximizing MSMQ Messaging Performance

MSMQ independent clients and servers can use up to three different drives to store the message files, message logger files, and transaction logger files. The drives must be local physical drives. If you have only two drives, store the message files on one drive and the message logger and transaction logger files on another.

Note In Windows NT, two or more IDE drives accessed through the same controller cannot be accessed simultaneously. Therefore, to benefit from having the data and log devices on separate physical drives, either use SCSI drives or use two IDE controllers.

To specify multiple disk drives for storing messages

1. double-click the **MS Message Queue** icon in Control Panel.
2. On the **Storage** tab, specify different physical drives. If you do not specify an existing folder, MSMQ creates the folder, as long as the parent folder already exists. For example, MSMQ creates the D:\Sorage\Msmq\Msgfiles folder only if the D:\Sorage\Msmq folder already exists.
3. When prompted to stop services, click **Yes**.
4. When prompted to shutdown and restart Windows, click **Yes**.

For information on using Performance Monitor to determine if additional hard disk drives will improve messaging performance, see “Monitoring MSMQ Messaging Performance” in Chapter 4, Monitoring Your MSMQ Enterprise.

Changing the Security on MSMQ Files and Folders

On computers running Windows NT Workstation, Widows NT Server, or Windows NT Server, Enterprise Edition, all MSMQ message files, log files, and LQS files are secured on NTFS-formatted drives. If you move the files to a different folder using the **MS Message Queue** icon in Control Panel, these same security settings are not copied to the new folders (because you may want to use, or may already be using, the folder to store other files).

To set the security settings on other folders to match the default security settings

1. Using My Computer or Windows NT Explorer, right-click the folder used to store the MSMQ message files, logger files, or transaction logger files, and then click **Properties**.
2. click **Permissions** on the **Security** tab.

   You can set file and folder permissions only on NTFS-formatted drives. If you select a folder on a FAT formatted drive, the **Security** tab is not available.
3. Select **Replace Permissions on Existing Files**.

   This check box is only available when editing the security of a folder. Even though you are replacing permissions on existing files, the folder's security settings are used for all new files that are created in the folder after setting the security.
4. specify **Full Control** For the local administrators group and **remove** all other users and groups from the **Name** box.

For more information on how MSMQ handles file security on the \Storage and \LQS folders, see "Securing MSMQ Files and Folders" in Chapter 5, "Securing Your MSMQ Enterprise."

Renaming a Computer Running MSMQ

When you change the computer name of an MSMQ dependent client, independent client, or server, you must also make additional configuration changes.

To rename an MSMQ computer

1. Remove your computer from the MSMQ enterprise.
2. Rename your computer. If you are running Windows NT, create a new computer account before you rename the computer, or select **Create a Computer Account in the Domain** when you rename the computer.
3. If your are reinstalling a controller server, reconfigure the SQL database for the new computer name.
4. Reinstall MSMQ.
5. Each user who logs on to the computer should log on and use the **MS Message Queue** Control Panel applet to renew his or her internal certificate.

Increasing MQIS Performance

You can increase MQIS performance on a server by increasing the number of connections between the MSMQ service and the local MQIS. You can specify between 1 and 1000 connections – 1 is the default. Each MQIS connection requires one additional client license.

Note If you are using the limited version of SQL Server 6.5 that is included with MSMQ, do not use more that 5 connections, or MQIS performance will degrade.

To increase the number of connections between the MSMQ service and the local MQIS

1. double-click the **MS Message Queue** icon in Control Panel.
2. On the **Server** tab, specify the number of connections.

Note You can also increase MQIS performance by putting the MQIS data and log devices on separate physical drives. However, this can be done only when installing MSMQ servers – once the MSMQ server is installed you cannot move the data or log devices.

Increasing the Size of the MQIS

If you need to increase the size of the MQIS after you have installed MSMQ, you can do so using Microsoft SQL Enterprise Manager. Typically, you should expand both the data device and the log device. The log device should be approximately 15 percent of the size of the data device.

The data and log devices on each MSMQ controller server should be the same size. For example, if you choose to have a 75 MB data device and a 12 MB log device and your enterprise has one PEC, two PSCs, and three BSCs, all six computers should have a 75 MB data device and a 12 MB log device.

Note The MSMQ limited version of SQL Server version 6.5 is limited to a cumulative total of 100 MB in database files. The SQL master database is 30 MB, which leaves 70 MB for a maximum MSMQ database of 60 MB (with 10 MB left for the logging of MSMQ database updates).

To increase the size of the MQIS

1. Run SQL Enterprise Manager.
2. On the **Manage** menu, click **Database Devices**.
3. Double-click the **MQISData** graph bar.

   If you don’t see the **MQISData** graph bar, use the scroll bar on the right side of the window to find it.
4. Change the **Size** value and then click **Change Now**.
5. Double-click the **MQISLog** graph bar.

   If you don’t see the **MQISLog** graph bar, use the scroll bar on the right side of the window to find it.
6. Change the **Size** value and then click **Change Now**.
7. Close the **Manage Database Devices** window.
8. On the **Manage** menu, click **Databases**.
9. Double-click the **MQIS** graph bar.

   If you don’t see the **MQIS** graph bar, use the scroll bar on the right side of the window to find it.
10. Click **Expand**.
11. In **Data Device** click **MQISData**.
12. In **Log Device** click **MQISLog**.
13. Click **Expand Now**.

For information on using Microsoft SQL Enterprise Manager, see the Microsoft SQL Server documentation.

Creating a Custom Managing Application

With the ActiveX™ controls provided by MSMQ, you can use Microsoft Visual Basic version 4 or 5 to create customized management and monitoring tools for your MSMQ enterprise. These tools can be used on MSMQ servers and MSMQ independent clients running under Windows NT Workstation.

The ActiveX controls provided by MSMQ can be used to:

- Create sites, CNs, and computers
- Change CNs, InRSs, and OutRSs for computers
- Change site gate settings for the PEC and PSCs
- View enterprise settings

For more information on creating applications with the ActiveX controls provided by MSMQ, see the Microsoft Message Queue Server Software Development Kit (MSMQ SDK) documentation.

Monitoring Your MSMQ Enterprise

You monitor your MSMQ enterprise network through MSMQ Enterprise Explorer and the following Windows NT administration tools: Event Viewer; Performance Monitor; and Network Monitor. For procedures for all MSMQ administrative tasks, see MSMQ Explorer Help.

Checking Computer Status Using MQPing

MSMQ supports an **MQPing** command, not unlike the ping feature supported by the TCP/IP protocol. Use **MQPing** (for MSMQ Explorer) to determine if an MSMQ computer is currently connected to the network. When a computer running MSMQ receives an **MQPing** request, it returns the contents of the request to a private queue called ***mqping response***.

When you ping a computer, MSMQ attempts to connect to the computer for 30 seconds, and updates the computer icon accordingly. MSMQ Explorer displays one of the following symbols next to the computer's icon.

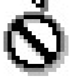
Table 4.1

Displayed

When the computer responds

When the computer does not respond within 30 seconds

›

Symbol	**Displayed**
	When the computer responds
	When the computer does not respond within 30 seconds

When you update the MSMQ Explorer display by pressing F5 , **MQPing** status symbols are no longer displayed. However, pinging additional computers does not affect the ping status symbols already displayed.

To ping more than one computer, select multiple computers in the right pane of MSMQ Explorer before right-clicking a computer and clicking **MQPing**. To select a contiguous block of computers, hold the SHIFT key while clicking the computers. To select non-contiguous computers, hold the CTRL key while clicking the computers.

Note Pinging many computers generates significant network traffic in large enterprises. Ping all computers only when necessary, and preferably when the network load is low.

Monitoring Events

MSMQ logs important information about its operation. This information is logged in different places, depending on which platform you use to run MSMQ.

Monitoring Events On Computers Running Windows 95

On computers running Windows 95, MSMQ logs important information about its operation to the Msmqlog.txt file. This file is located in the same folder used to store MSMQ messages (C:\Program Files\Msmq\Storage by default). Use Notepad, WordPad, or any other application that can read text files to view the Msmqlog.txt file.

Note If you are using an application to view Msmqlog.txt when an event occurs, the event is not logged.

Monitoring Events On Computers Running Windows NT

On computers running Windows NT, MSMQ logs important information about its operation to Windows NT event logs. The event source for all events is always MSMQ. Most MSMQ events are written to the Application log, while security events are written to the Security log. MSMQ application events can be of three types:

- Informational
- Warning
- Error

Informational events are infrequent significant events that describe successful operations. Warning events are not necessarily significant, but indicate possible future problems. Error events are generated for significant problems.

Note Some application events are not logged at each occurrence. For example, if an MSMQ routing server cannot establish a session with the target computer, an event is logged for the first failure, and another event is logged when the session is established.

MSMQ security events can be of two types:

- Success Audit
- Failure Audit

Success Audit events are generated when selected objects are accessed successfully, while Failure Audit events are generated when access to a selected object is denied.

For information on enabling MSMQ auditing, see “Auditing” in Chapter 5, “Securing Your MSMQ Enterprise”.

Viewing Events

You can view events from MSMQ Explorer or from Windows NT Event Viewer. MSMQ Explorer displays those events of type **MSMQ**. If you view the events from Event Viewer, you can sort, filter, and search for events based on different event properties.

To view events from Windows NT Explorer, right-click a computer, click **Properties**, and then click the **Events** tab.

For more information on Event Viewer, see *Windows NT Server version 4.0 Concepts and Planning*.

Monitoring MSMQ Performance

You can use Windows NT Performance Monitor and MSMQ Performance Monitor counters to monitor and tune you MSMQ enterprise.

MSMQ Performance Monitor Counters

Many performance statistics can be viewed from MSMQ Explorer for both computers and queues. To see the performance counters from MSMQ Explorer, right-click a computer or a queue, click **Properties**, and then click the **Status** tab. To view all available performance counters for MSMQ, run Windows NT Performance Monitor. The default MSMQ performance counters are grouped under the following objects:

- MSMQ IS
- MSMQ Service
- MSMQ Queue
- MSMQ Session

For definitions of each counter, run Performance Monitor, click the **Add counter** button, select one of the MSMQ objects, and then click **Explain**. As you select different counters, definitions for each counter are displayed under **Counter Definition**.

Additional performance counters might also be installed on MSMQ servers by MSMQ connectors. For example, performance counters are grouped under the MSMQ Exchange Connector object when you install the MSMQ Exchange Connector.

You can also use Perf2mib.exe (the Performance Monitor MIB Builder Tool) to enable any simple network management protocol (SNMP) manager to read MSMQ performance counters. For more information, see the *Windows NT Workstation Resource Kit* or the *Windows NT Server Resource Kit*.

For more information on Performance Monitor, see *Windows NT Server version 4.0 Concepts and Planning*. For more information the MSMQ Exchange connector, see Appendix B “Installing and Configuring the MSMQ Exchange Connector.”

The MSMQ IS Object

The MSMQ IS object is available on all MSMQ controller servers. The MSMQ IS object contains performance counters that you can use to monitor MQIS access.

For information on each counter, run Performance Monitor, click **Add to Chart** on the **Edit** menu, select the **MQIS IS** object, click the **Explain** button, and then click the various counters.

The MSMQ Service Object

The MSMQ Service object is available on any computer running the Microsoft Message Queue Service. The MSMQ Service object contains performance counters you can use to monitor session and message statistics for the selected computer.

The MSMQ Queue Object

The MSMQ Queue object contains performance counters you can use to monitor message statistics for the selected queue. There are instances for each queue on the computer. If an MSMQ-based application has a private queue open on another computer, an instance for that queue is also available. The instance name of these queues is the format name of the queue. If an MSMQ connector application is running on the computer, an instance is available for the MSMQ connector queue. The queue name is displayed as foreign=GUID.

Instance lists only the first 97 queues created on the computer, including private queues.

Note The **Computer Queues** instance represents the computer's source journal and dead letter queues. The **Bytes in Queue** and **Messages in Queue** counters monitor the computer's dead letter queue. The **Bytes in Journal Queue** and **Messages in Journal Queue** counters monitor the computer's source journal queue.

Temporary queues are easily identifiable because the computer name in the instance does not match the name of the computer on which they reside.

The MSMQ Session Object

The MSMQ Session object is available when the computer has an MSMQ session established with another computer. Multiple instances of this object are available, one for each session. **Instance** lists IP or IPX address of the computer with which the session is established.

Monitoring MSMQ Messaging Performance

If available processing resources (CPU time) are not limiting messaging performance, MSMQ independent clients and servers may be able to improve messaging performance by storing the various files used to implement messaging across multiple physical disks.

MSMQ can use two different drives to send and receive standard messages, or three for transactional messages. If you are sending transactional messages, you may also be able to put the data and log files for the application you are transacting with on two or more different drives. However, if your CPU usage is at or near 100 percent, adding additional disk drives will not improve messaging performance.

To determine if additional hard disk drives will improve messaging performance, use Performance Monitor to track the **% Processor Time** and **Avg. Disk Queue Length** counters. If sustained CPU use is above 75% while sending messages, adding additional processing capacity may improve messaging performance. If the average disk queue length for any drive is greater than 0.6 while sending messages, additional drives may improve messaging performance.

For more information on putting the MSMQ message queuing files on separate physical drives, see “Maximizing MSMQ Messaging Performance” in Chapter 2, Installing MSMQ.

Tracking and Monitoring Messages

You can monitor message routes and message content by:

- Sending a test message
- Enabling message route tracking
- Using network monitor

Sending a Test Message

Using MSMQ Explorer, you can send test messages from any MSMQ independent client or server to a test queue on any other independent client or server within the same enterprise.. You can use MSMQ Explorer to create test queues or change the type ID of existing queues so you can send test messages to the queues.

Note Test queues have a type ID of {55EE8F33-CCE9-11CF-B108-0020AFD61CE9}. If you change the type ID of a transactional queue to {55EE8F33-CCE9-11CF-B108-0020AFD61CE9}, and send test messages to the transactional queue, the messages are not delivered. Only transactional messages can be sent to transactional queues, and MSMQ Explorer test messages are not transactional. Similarly, if you change the type ID of a queue requiring authentication to {55EE8F33-CCE9-11CF-B108-0020AFD61CE9}, and send test messages to the queue, the messages are not delivered (unless the target queue is located on the sending computer).

For information on how to send a test message using MSMQ Explorer, see MSMQ Explorer Help.

Enabling Message Route Tracking

You can use message route tracking to track the path messages take on your MSMQ network. As the messages pass through an MSMQ routing server, a message is sent to the report queue indicating which server the message passed through.

Enabling message route tracking is a two step process. First, the administrator must enable message route tracking by defining a report queue for the sending computer. Second, the application must set the correct message property. The application developer must set this property, or enable some way for it to be set by the user (for example, through the Windows registry), so that message route tracking can be enabled for the application.

Because all test messages are sent with message route tracking enabled, if the sending computer has a report queue, the message path is tracked.

For information on how to enable message route tracking, see MSMQ Explorer Help. For information on how to set message properties, see the Microsoft Message Queue Server Software Development Kit (MSMQ SDK).

Report Message Format

Report message labels are written in the following formats:

gggg:dddd:hh sent from <*computer*> to <*address*> at <*time*>

gggg:dddd:hh received by <*computer*> to <*address*> at <*time*>

where *gggg* is the first 4 hexadecimal digits of the source queue *globally unique identifier* (GUID), *dddd* is the internal message identifier, and *hh* is the hop count.

The internal message identifier is the message sequence number, which is part of the entire message identifier. The entire message identifier is not used because it is too big to be easily read (a 16 byte source GUID and a 4 byte sequence number).

Avoiding Report Conflicts

Each message supports only one report queue. If a message is being tracked, and it passed through an MSMQ routing server that also has message route tracking enabled, a report conflict message is sent to the report queue.

For example, suppose report message tracking is set for computer A, and the MSMQ-based application on computer A is sending message M to computer C through MSMQ routing server B. If the computer A report queue is queue R, a report message is sent to queue R when:

- Message M leaves computer A
- Message M arrives at computer B
- Message M leaves computer B
- Message M arrives at computer C

However, if computer B (an MSMQ routing server) is also configured with report message tracking, with report messages being sent to queue R, queue R receives a report queue conflict message when Message M leaves computer B. Queue R continues to receive correct tracking messages.

Using Network Monitor

MSMQ includes a Microsoft Network Monitor parser you can use to monitor MSMQ traffic on your network. Two parsers are included with MSMQ. One parser is compatible with Windows NT Server version 4.0 Network Monitor and SMS version 1.2 Network Monitor. The other parser is compatible with SMS version 1.1 Network Monitor. Both parsers are installed on your hard disk when you install the MSMQ SDK on a computer running Windows NT Workstation or Windows NT Server. The parsers are not installed on Windows 95 computers.

To install the MSMQ parser

1. If the MSMQ SDK is not already installed, click **Start**, point to **Programs**, point to **Microsoft Message Queue**, click **Setup**, click **Add/Remove**, and select **Software Development Kit**.
2. If Network Monitor is not yet installed, install it from either Windows NT Server or SMS version 1.1 or later.
3. Run Inst.bat from the MSMQ SDK\NMParser folder (typically c:\Program Files\Msmq\Sdk\Nmparser).
   - To install the parser that is compatible with Windows NT Server version 4.0 Network Monitor and SMS version 1.2 Network Monitor, type the following at the Windows NT command prompt:

     inst <*folder*> 4.0

     where <*folder*> is your Network Monitor folder.
   - To install the parser that is compatible with SMS version 1.1 Network Monitor, type the following at the Windows NT Command Prompt:

     inst <*folder*> 1.1

     where <*folder*> is your Network Monitor folder.

Note If **N/A** appears in the MSMQ protocol fields, you probably have the wrong MSMQ parser installed. To correct this, quit Network Monitor and then install the correct parser.

For more information on the Network Monitor, see *Windows NT Server version 4.0 Concepts and Planning* or your SMS documentation.

Securing Your MSMQ Enterprise

MSMQ supports access control, auditing, authentication, and encryption to secure your MSMQ enterprise. The "Access Control" section of this chapter explains how to limit access to MSMQ resources on your MSMQ enterprise. The "Auditing" section explains how to audit MSMQ events. The "Authentication" and "Encryption" sections explain administrative consequences and costs of using authentication and encryption.

If you do not need to set up a secure MSMQ enterprise, see "Simplified MSMQ Security" later in this chapter.

Windows NT Domain Considerations

To control access to MSMQ objects, MSMQ computers must log on to a Windows NT Server domain. If you choose not to use access control, MSMQ dependent clients, independent clients, and servers need not log on to a Windows NT Sever domain.

If MSMQ servers are located in different domains, the servers must trust the domains in which subordinate servers reside. In other words:

- The domain of which the PEC is a member must trust each domain of which the PEC’s PSCs are members.
- The domain of which the PEC is a member must trust each domain of which the PEC’s BSCs are members.
- The domain of which a PSC is a member must trust each domain of which the PSC’s BSCs are members.

These trusts need be only one-way trusts, and there is no need for the PEC to trust domains of with BSCs of other sites are members. There is no need for any trust between The domains of which PSCs are members.

If a user accesses a server in a domain that does not trust the domain of which the user is a member, the user is considered a guest user on that server. If the guest user account is disabled on that server, that user will not be able to connect to the server.

If a user sends a message to a queue that has a restriction on who can send a message to the queue, the computer that owns the queue must be a member of a domain that trusts the domain of the user that sent the message. Otherwise the message is rejected with access denied.

Security Requirements and Limitations

To use MSMQ security features, both the sending and receiving computers must be logged on to a Windows NT Server domain. If you are not logged on to a domain (for example, the computer is a member of a workgroup), you cannot create secured objects. That is, everyone will have full control over queues you create and your computer properties stored in the MSMQ information store (MQIS). In this case, you will only be able to send messages to queues and access objects that grant access to everyone.

By default, all users have Read access to the MQIS database. You cannot change this default behavior. However, you can set specific access rights on each object in MQIS. For example, without the correct permissions, users cannot read from queues.

Users who log on locally (instead of being validated by a domain controller) cannot create or register certificates. Certificates are only useful for authenticating the sender's security identifier (SID) when the user has logged on and been validated by a domain controller. In addition, local users cannot send messages with their SID in the message sender ID property.

Access Control

Access control is the process of restricting access to resources. MSMQ access control is based on Windows NT security, which functions only when users are logged on and have access to domain controllers. However, MSMQ supports store-and-forward message queuing, so the source and destination computers do not need to be online at the same time. For MSMQ to support access control for disconnected users, the destination computer must check the access rights of the user who sent the message. Because the user who sent the message may not be logged on, MSMQ uses the sender's security identifier (SID) to verify the sender's access rights on a queue.. If a queue restricts who can send messages to it, the sending application must attach the sender's SID to a message.

Caution In MSMQ, security is defined on each object. Therefore, if you deny yourself the Get Properties permission to a queue, you will not be able to see the queue, and will not be able to regain access to it, even if you still have the Change Security permission for the queue. However, it is still possible to regain control programmatically using the MSMQ API.

Also, any user who has the Take Ownership right on a PEC, PSC or BSC, can take ownership of any object within the enterprise without explicitly having the Take Ownership access right on the specific object.

Required Access to Install MSMQ Servers

To install a PSC, you must have the required MSMQ access permissions and belong to the Administrator's group on the PEC computer.

To install a BSC, you must have the required MSMQ access permissions and belong to the Administrator's group on the PSC or PEC computer that the BSC will support.

You can limit access to the enterprise, sites, CNs, computers, and queues on an individual or group basis. For information on how to set permissions, see MSMQ Explorer Help.

Setting Permissions for the Enterprise

By default, Everyone has Read permission for the enterprise, and the enterprise administrator (the person who installed the enterprise) has Full Control permission. With Read permission for an enterprise, users can retrieve the enterprise permissions settings and register certificates in the MQIS. In other words, users have Get Permissions and Create User permissions.

When granted Write permission for an enterprise, users can create sites, create CNs, retrieve the enterprise permissions settings, and register certificates in the MQIS. In other words, users have Create Site, Create CN, Get Permissions, and Create User permissions.

You can specify Full Control, Read, Write, or Special Access permissions for all users and groups. You can set the following Special Access permissions for the enterprise.

Table 5.1

Special Access Permission	**Use to allow a user or group to…**
Create Site (Cs)	Create a site.
Create CN (Cc)	Create a connected network.
Create User (Cu)	Register certificates in the MQIS.
Set Properties (Sp)	Set enterprise properties. This includes creating, modifying, and deleting site links and deleting certificates of other users.
Delete Enterprise (D)	Delete the enterprise.
Get Permissions (Pg)	View the enterprise permissions.
Set Permissions (Ps)	Set the enterprise permissions.
Take Ownership (O)	Take ownership of the enterprise.

Only users who have Full Control for an enterprise can create a PEC in the enterprise. Only users who have the Create Site permission for an enterprise can create a site in the enterprise.

Setting Permissions for a Site

By default, everyone has Write permission for a site, and the Site administrator (the person who installed the site) has Full Control permission. With Write permission for a site, users can retrieve the site's permissions settings and install a computer in the site. In other words, users have Get Permissions and Create Computer permissions.

When granted Read permission for a site, users can retrieve the site's permissions settings. In other words, users have Get Permissions permissions.

You can specify Full Control, Read, Write, or Special Access permissions for all users and groups. You can set the following Special Access permissions for a site.

Table 5.2

Special Access Permission	Use to allow a user or group to…
Create Route Server (Cr)	Install an MSMQ routing server in the site.
Create BSC (Cb)	Install a BSC in the site.
Create Computer (C)	Install a computer in the site.
Set Properties (Sp)	Set site properties.
Delete Site (D)	Delete the site.
Get Permissions (Pg)	View site permissions.
Set Permissions (Ps)	Set site permissions.
Take Ownership (O)	Take ownership of the site.

Setting Permissions for a CN

By default, Everyone has Read permission for a CN, and the CN administrator (the person who created the CN) has Full Control permission. With Read permission for a CN, users can retrieve the CN's permissions settings. In other words, users have Get Permissions permissions. Write permission for a CN is identical to Read permission.

You can specify Full Control, Read, Write, or Special Access permissions for all users and groups. You can set the following Special Access permissions for a CN.

Table 5.3

Special Access Permission	**Use to allow a user or group to…**
Open Connector	Run an MSMQ connector application and open MSMQ connector queues.
Set Properties (Sp)	Set CN properties.
Delete CN (D)	Delete the CN.
Get Permissions (Pg)	View CN permissions.
Set Permissions (Ps)	Set CN permissions.
Take Ownership (O)	Take ownership of the CN.

Setting Permissions for a Computer

By default, Everyone has Write permission for a computer, and the computer administrator (the person who installed the computer) has Full Control permission. With Write permission for a computer, users can retrieve the computer's permissions settings and create queues on the computer. In other words, users have Create Queue, and Get Permissions permissions.

When granted Read permission for a computer, users can receive messages from the computer's journal and dead letter queues and retrieve the computer's permissions settings. In other words, users have Receive Dead Let., Peek Dead Letter, Receive Journal, Peek Journal, and Get Permissions permissions.

When you use MSMQ Explorer to grant either Receive Dead Let. or Receive Journal permissions to a user or group, that user or group also automatically gets the corresponding Peek permission because the combination of the Receive permission without the Peek permission is not useful. You can assign the Receive permission without assigning the Peek permission only programmatically.

You can specify Full Control, Read, Write, or Special Access permissions for all users and groups. You can set the following Special Access permissions for a computer.

Table 5.4

Special Access Permission	**Use to allow a user or group to…**
Receive Dead Let. (Rd)	Receive dead letter messages from the computer's dead letter queue.
Peek Dead letter (Pd)	View all dead letter messages on the computer's dead letter queue.
Receive Journal (Rj)	Receive journal messages from the computer's journal queue.
Peek Journal (Pj)	View all journal messages on the computer's journal queue.
Create Queue (C)	Create queues on the computer.
Set Properties (Sp)	Set the computer properties.
Delete Computer (D)	Delete the computer.
Get Permissions (Pg)	View computer permissions settings.
Set Permissions (Ps)	Change permissions for a computer.
Take Ownership (O)	Take ownership of the computer.

Setting Permissions for a Queue

By default, everyone has Send permission for a queue, and the Queue administrator (the person who created the queue) has Full Control permission. With Send permission for a queue, users can retrieve the queue's properties and permissions settings and send messages to the queue. In other words, users have Send Message, Get Properties, and Get Permissions permissions.

When granted Receive permission for a queue, users can retrieve the queue's properties and permissions settings and view/receive messages from the queue. In other words, users have Receive Journal, Receive Message, Peek Message, Get Properties, and Get Permissions permissions.

When you use MSMQ Explorer to grant either Receive Message or Receive Journal permissions to a user or group, that user or group also automatically gets the Peek Message permission because the combination of the receive permission without the peek permission is not useful. You can assign the Receive permission without assigning the Peek permission only programmatically.

You can specify Full Control, Receive, Send, or Special Access permissions for all users and groups. You can set the following Special Access permissions for a queue.

Table 5.5

Special Access Permission	Use to allow a user or group to…
Receive Journal (Rj)	Read journal messages from the queue.
Receive Message (Rq)	Read messages from the queue.
Peek Message (P)	View messages in the queue and queue journal.
Send Message (Sq)	Send a message to the queue.
Set Properties (Sp)	Set the queue properties.
Get Properties (Gp)	View the queue properties.
Delete Queue (D)	Delete the queue.
Get Permissions (Pg)	View the queue permissions settings.
Set Permissions (Ps)	Change permissions for the queue.
Take Ownership (O)	Take ownership of the queue.

Queue Permissions

To open a queue, both the user running the MSMQ-based application and the user account that runs the MSMQ service must be able to retrieve the queue's properties from the MQIS. If either the user running the MSMQ application or the user account used to run the MSMQ service does not have the Get Properties permission, the open operation will fail. By default, all queues are created with the Get Properties permission enabled for everyone. If you restrict this permission, ensure that the user account used to run the MSMQ service has this access permission.

In addition, the user accounts used to run the MSMQ service on every server that forwards the message also must have the Get Properties permission. Because it is not possible to predetermine message routes, all MSMQ servers should have the Get Properties permission on all queues in the enterprise.

Users need the correct permission to perform each specific action, and they need the Get Properties permission to locate queues.

Simplifying the Administration of Access Control

You can simplify the administration of MSMQ access control by creating one or more global groups for administrators. For example, you might create one global group for Enterprise Admins and Site Admins global groups for each site.

The Enterprise Admins group should have administrator privileges on each Windows NT Server used as an MSMQ server (this privilege must be granted with User Manager for Domains). This group should also be granted Full Control permissions on the MSMQ enterprise, each site, each CN, and each computer used as an MSMQ server. (These permissions can be granted through MSMQ Explorer.)

A Site Admins group should have administrator privileges on each Windows NT Server used as an MSMQ server within the site. This group should also be granted Full Control permissions on the site and each computer used as an MSMQ server within the site. (These permissions can be granted through MSMQ Explorer.)

Using these groups, Enterprise Admins can install and configure an MSMQ server in any site, as well as install PSCs. Site Admins can install and configure BSCs and MSMQ routing servers within a site, but cannot create new sites.

Permissions and Rights Required to Install MSMQ

To install MSMQ clients and servers, users need one or more of the following:

- **Administrative rights on the local computer**

  Administrative rights apply only to computers running Windows NT—For that reason, they do not apply to installing MSMQ clients on computers running Windows 95.
- **Appropriate MSMQ permissions**

  The appropriate MSMQ permissions are required to install MSMQ servers and MSMQ independent clients.
- **SQL database administrative rights**

  SQL database administrative rights are required to install MSMQ controller servers (PEC, PSCs, or BSCs).

These required rights and permissions are explained in the following sections.

Administrative Rights on the Local Computer

To install an MSMQ server or client on a computer running Windows NT, the user account under which you logged on must belong to the local Administrators group on that computer.

Note If you add a user to the local Administrators group on a server, and that user is currently logged on (to another computer), that user must reboot his or her computer and log back on before he or she is granted administrative rights.

MSMQ Permissions

The following table describes the MSMQ permissions required to install MSMQ servers and clients.

Table 5.6

Installation type	Permissions required	Who has them by default
PSC	Create Site and Set Properties on the enterprise	The person who installed the enterprise (as a subset of Full Control)
BSC	Create BSC on the site	The person who installed the site (as a subset of Full Control)
MSMQ routing server	Create Route Server on the site	The person who installed the site (as a subset of Full Control)
Independent client	Create Computer on the site	Everyone
Dependent client	None.	N/A

SQL Database Administrative Rights

To install MSMQ controller servers (PEC, PSCs, or BSCs) you must have database administrative rights on both the computer on which you are installing MSMQ and the controller server above it. In other words, to install a PSC, you must have database administrative rights on the PEC, and to install a BSC, you must have database administrative rights on the PEC or PSC it is supporting.

By default, the local Administrators group has database administration rights on the computer.

Note If you add a user to the local Administrators group on a server, and that user is currently logged on (to another computer), that user must reboot his or her computer and log back on before he or she is granted administrative rights.

Auditing

You can use auditing to record which users access which objects, the type of access attempted, and if that access succeeded or failed. Audited events are recorded in the Windows NT security log, and can be viewed from Windows NT Event Viewer.

To enable auditing, you must set the audit policy for all MSMQ servers (using User Manager for Domains on each server), and then use MSMQ Explorer to specify which actions to audit, for which objects, for which users, and whether to audit successful access, failed access, or both.

You can audit actions for the enterprise, sites, CNs, computers, and queues on an individual or group basis. For each object, you can audit different actions. These are explained later in the following sections:

- Auditing the Enterprise
- Auditing a Site
- Auditing a CN
- Auditing a Computer
- Auditing a Queue

Note To write messages to the security audit log, the user account that runs the MSMQ service must have the "**Generate security audits**" right. If the account does not have this right, the MSMQ service writes a warning message in the application log. By default, the MSMQ service runs in the local system account which, by default, has the “**Generate security audits**” right.

Choosing the MSMQ Service User Account

the MSMQ service (called Microsoft Message Queue Service) and the SQL Server service (called MSSQLServer) both run under a user account. Both the SQL Server service and the MSMQ service can run under the local system account, or a specific user account. Setup installs the MSMQ service to run under the local system account.

The following table shows the advantages and disadvantages for using the local system account to log on the MSMQ service.

Table 5.7

local system account advantages	local system account disadvantages
The local system account belongs to the local administrators group.	Queues that restrict the Get Properties and Get Permissions permissions cannot be managed by MSMQ services that run under the local system account on MSMQ independent clients and MSMQ servers.
The local system account already has the required rights to run as a service.	Communication with other MSMQ computers is less reliable. When a domain account is used on both computers running MSMQ, the communications between the two services is more reliable.
The local system account already has the "**Generate security audits**" right.	

The following table shows the requirements and advantages of using a user account to log on the MSMQ service.

Table 5.8

User account requirements	User Account advantages
The account should be added to the local administrators group on the computer.	The Get Properties permissions can be granted selectively to certain queues. Only MSMQ services that run under the proper user account can handle those queues.
The account should be granted the "**Log on as a service**," "**Act as part of the operating system**," "**Create a token object**," "**Increase quotas**" and "**Generate security audits**" rights.	

Note If you use Control Panel to specify the account MSMQ logs on under, the "**Log on as a service**" right is granted automatically for the account you are currently using, but you must specify the other rights.

Changing the MSMQ Service User Account

You can change the account used by MSMQ using Control Panel.

To change the account the MSMQ service uses

1. double-click **Services** on Control Panel.
2. Click **Microsoft Message Queue Service** and click **Startup**.
3. type the new user account in **This Account**.
4. type the account password in **Password** and **Confirm Password**.

Setting Windows NT Rights

The account under which the MSMQ service is running needs one or more of the following Windows NT rights.

Table 5.9

Windows NT right	**Required to…**
Log on as a service	Allow the MSMQ service to start.
Generate security audits	Generate audits.
Act as part of the operating system	Handle users who are not logged on to a Windows NT domain.

If the MSMQ service is running under the local system account, all of the rights in the preceding table are granted by default. If you use Control Panel to change the user account of the MSMQ service, that account is automatically granted the “**Log on as a service**” right. the rest of the rights should be set with User Manager.

To grant the required rights to an account

1. Run **User Manager** or **User Manager for Domains**.
2. On the **Policies** menu, click **User Rights**.
3. Select **Show Advanced User Rights**.
4. Select **Generate security audits** from the **Rights** list.
5. Click **Add**.
6. In **Add Names**, type the account name in the domain\user format, Or click **Show Users** and then double-click the user’s name.
7. Repeat this procedure for the **Act as part of the operating system**, **Create a token object**, and **Increase quotas** rights.

Repeat this procedure on each MSMQ controller server and each MSMQ independent client that is set to run the MSMQ service under an account other than the local system account.

Audit Log Messages

Audit log messages are written in the event log on the server that performs the actual operation, but not necessarily the server that owns the object. So, audit log messages for a single object are usually scattered across the enterprise. For example, in Figure 5.1, if the Test queue on User 1's computer in the San Francisco site is deleted by the User 2 (located in the Seattle site), the audit message is written on the PEC of the Seattle site.

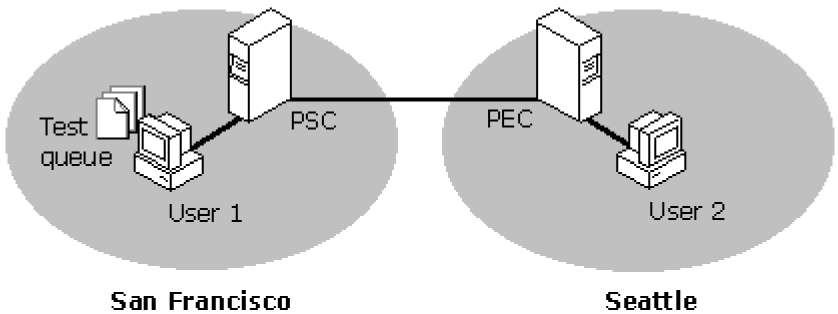


Figure 5.1

Because all queue operations except Read Message and Peek Message are performed by an MSMQ controller server, audits are generated on the server that carried out the operation. However, when a queue is opened for read or peek, the audits for this operation are always generated on the computer where the queue resides. The audit happens when the queue is opened (for peek or receive), not each time a message is received.

Note Each audit has an open entry and close entry in the event log. For Read Message and Peek Message audits, the close audit is written to the event log immediately after the open audit is written to the event log.

Setting the Audit Policy

Before you can use MSMQ Explorer to specify which events to audit, you must enable file and object access auditing with User Manager. You can audit success and failure audits, or both. If audits are set on an object and auditing is not enabled on the MQIS that is used for carrying out the operation, the audit message is not generated. The following table shows which computer you must enable auditing on for each action.

Table 5.10

Action	Enable auditing
Peek Message, Receive Message, Peek Journal, or Receive Journal	On the computer where the queue resides.
Peek Computer Journal Message, Receive Computer Journal Message, Peek Dead Letter Message, Receive Dead Letter Message, or Open Connector	On the computer you want to audit.
Any other actions	On all MSMQ controller servers.

When you use MSMQ Explorer to set auditing, you must have the “**Manage auditing and security log**” right on the MSMQ controller server that will perform the action.

When you use MSMQ Explorer to set auditing, a message may appear stating that the current audit policy does not have auditing enabled. This indicates auditing is turned off on the computer that you using—it does not mean that auditing is turned off on the computer you are going to audit.

Auditing the Enterprise

You can audit the success or failure of the following enterprise actions for individual users or groups:

- Create Site (Cs)
- Create CN (Cc)
- Create User (Cu)
- Set Properties (Sp)
- Delete Enterprise (D)
- Get Permissions (Pg)
- Set Permissions (Ps)
- Take Ownership (O)

Auditing a Site

You can audit the success or failure of the following site actions for individual users or groups:

- Create Route Server (Cr)
- Create BSC (Cb)
- Create Computer (C)
- Set Properties (Sp)
- Delete Site (D)
- Get Permissions (Pg)
- Set Permissions (Ps)
- Take Ownership (O)

Auditing a CN

You can audit the success or failure of the following CN actions for individual users or groups:

- Open Connector
- Set Properties (Sp)
- Delete CN (D)
- Get Permissions (Pg)
- Set Permissions (Ps)
- Take Ownership (O)

Auditing a Computer

You can audit the success or failure of the following computer actions for individual users or groups:

- Receive Dead Let. (Rd)
- Receive Journal (Rj)
- Create Queue (C)
- Set Properties (Sp)
- Delete Computer (D)
- Get Permissions (Pg)
- Set Permissions (Ps)
- Take Ownership (O)

When you use MSMQ Explorer to audit the use of Receive Dead Letter for a user or group, use of Peek Dead Letter is also audited for that user or group. Similarly, when you use MSMQ Explorer to audit the use of Receive Journal for a user or group, use of Peek Journal is also audited for that user or group.

Auditing a Queue

You can audit the success or failure of the following queue actions for individual users or groups:

- Receive Journal (Rj)
- Receive Message (Rq)
- Set Properties (Sp)
- Get Properties (Gp)
- Delete Queue (D)
- Get Permissions (Pg)
- Set Permissions (Ps)
- Take Ownership (O)

When you use MSMQ Explorer to audit the use of either Receive Message or Receive Journal for a user or group, use of Peek Message is also audited for that user or group.

Authentication

Authentication is the process of verifying that the message sender is authentic (in other words, the sender has not been impersonated by another user). MSMQ uses digital signatures to provide message authentication. MSMQ authentication also guarantees message integrity (the message has not been tampered with) and *non-repudiation* (because no user can sign a message with another users identity, no user can refute that he or she sent a message if it contains his or her signature). Authentication causes reduced throughput of MSMQ messages.

MSMQ authenticates messages using internal or external security certificates. A certificate is used to generate a digital signature that uniquely identifies a user who sends an authenticated message. Both the user's certificate and digital signature are attached to a message when it is sent. The digital signature is encrypted by the MSMQ service on the sending computer and decrypted by the MSMQ service on the destination computer.

Internal certificates, provided by MSMQ, authenticate the Windows NT security identifier (SID) of the sender. When an internal certificate is used, MSMQ guarantees only that the SID attached to the message is valid. Internal certificates are created by MSMQ the first time a user runs the **MS Message Queue** applet in Control Panel. Users must register their internal certificates in the MQIS before sending authenticated messages.

External certificates use information about a user supplied by a certificate authority (CA), rather than the sender's SID, to verify the sender's identity. The information in the external certificate is guaranteed by the certificate authority that created the certificate. External certificates are required for sending authenticated messages to operating environments other than Windows NT. If external certificates are used for sending authenticated messages to other Windows NT computers, users must register their certificates only if they also want their SID to be used in authenticating the message. Doing so provides an additional, though optional, measure of authentication.

A common way to obtain an external certificate is to request a class 1 certificate from a certificate authority, using Microsoft Internet Explorer (IE) version 3.0 or later. Because external certificates are stored in the IE certificate store, users who rely on external certificates must have IE 3.0 or later installed on their computers.

To obtain external certificates, contact one of the following certificate vendors:

- AT&T
- InternetMCI Mail
- Keywitness Canada
- VeriSign (http://digitalid.verisign.com/)

Note Internal certificates and SIDs are stored in the MQIS. If you reinstall Windows 95 or Windows NT on an MSMQ independent client (rather than upgrade an existing installation), you must reinstall MSMQ and rerun the **MSMQ** Control Panel applet to remove the user's previous obsolete internal certificate from the MQIS. You can then register the user's new internal certificate.

The encryption process necessary for generating digital signatures is not the same as the encryption of *messages* discussed in the next section. (Encrypted messages need not be authenticated, and authenticated messages need not be encrypted). Both processes, however, require the use of a cryptographic service provider (CSP). A CSP is the actual code that implements cryptographic operations.

A CSP must be installed on any server, independent client, or dependent client computer that sends or receives authenticated messages. The Microsoft Base Cryptographic Provider version 1.0 is a CSP that provides full RSA support; that is, it can be used for both authentication and encryption of messages. The Microsoft Base Cryptographic Provider is included and installed with Windows NT version 4.0 and Internet Explorer version 3.0. The CSP that you use to authenticate MSMQ messages does not require full RSA support, only the ability to generate digital signatures. If you run an application that requires a CSP other than the Microsoft Base Cryptographic Provider, you must install that CSP before enforcing authentication on MSMQ queues. You must also install a CSP on computers that are not running Windows NT 4.0 or IE version 3.0.

Important It is important to note that the same CSP must be installed on both a sending and receiving computer. For example, a computer using the Microsoft Base Cryptographic Provider cannot send an authenticated message to a computer that uses a different CSP, and vice-versa.

The Microsoft Base Cryptographic Provider is available to computers running Windows 95 only if Internet Explorer version 3.0 or later is installed.

For more information on internal and external certificates, and for information on implementing authentication in an MSMQ-based application, see the Microsoft Message Queue Server Software Development Kit (MSMQ SDK) documentation.

Encryption

Encryption is the process of encrypting and decrypting messages, ensuring they cannot be read or used by anyone not authorized to do so. MSMQ supports encryption through the use of public and private keys. The MSMQ public key implementation is based on the Microsoft® CryptoAPI, and uses the Microsoft Base Cryptographic Provider version 1.0.

As with authentication, encryption of MSMQ messages requires the use of a cryptographic service provider (CSP). Even though you can install a variety of CSPs for authentication, the Microsoft Base Cryptographic Provider version 1.0 must be installed on any MSMQ server, independent client, or dependent client computer that sends or receives encrypted messages. (When you install the Microsoft Base Cryptographic Provider version 1.0, non-MSMQ applications can continue to use other CSPs for encryption.) The Microsoft Base Cryptographic Provider is included and installed with Windows NT version 4.0 and Internet Explorer version 3.0.

Note Computers running Windows 95 must have IE 3.01 or later installed to send and receive encrypted MSMQ messages. This is an MSMQ Beta 2 limitation.

You can specify whether a queue accepts encrypted messages, unencrypted messages, or both. If the queue privacy level is None, the queue accepts only unencrypted messages. If the queue privacy level is Optional, the queue accepts both unencrypted and encrypted messages. If the queue privacy level is Body, the queue accepts only encrypted messages.

Messages are encrypted at the source computer and decrypted at the destination computer–they appear in the destination queue as clear text. If a queue’s privacy level is Optional, a user can verify whether a message sent to the queue was encrypted or unencrypted by checking the message’s properties in MSMQ Explorer.

When you use encryption, MSMQ message throughput is reduced significantly, although typically not as significantly as when using authentication. Encryption performance losses are most significant when you send few encrypted MSMQ messages to many different computers. However, when you send multiple messages to the same destination, only the first message sent takes significantly longer to send.

For more information on cryptography, public key encryption, and the Microsoft CryptoAPI, see the Microsoft Crypto Application Programmer’s Guide, available on the Microsoft Web site at http://www.microsoft.com/intdev/security/. For information on implementing encryption in an MSMQ-based application, see the MSMQ SDK documentation.

Securing Communication with Controller Servers

As previously explained, MSMQ controller servers (PECs, PSCs, and BSCs) hold copies of the MQIS database. The MQIS is a distributed database that holds enterprise topology, enterprise settings, computer information, queue information, and user information. MSMQ-based applications can query the MQIS to find queues and get queue properties.

Communication between MSMQ controller servers is inherently secure, because all the messages are signed and verified based on information found in the MQIS database. However, by default, MSMQ does not secure MQIS access by MSMQ clients (both independent and dependent) or MSMQ routing servers.

you, As an MSMQ administrator, can secure MQIS access by MSMQ clients or MSMQ routing servers by installing a certificate on each controller server. This ensures that unauthorized persons cannot install software that impersonates a controller server for the purpose of providing false information or obtaining confidential information. You can enable secure MQIS access on individual controller servers, and you can configure each MSMQ client to trust only specific certificate authorities.

Note Because MSMQ-based applications running on a controller server will access the MQIS database on the local computer, you do not need to use certificates to secure access to the local MQIS database on controller servers.

The certificate used to secure MQIS access on a controller server must be installed separately from any certificates installed for message authentication. However, if properly configured, the same certificate can be used by MSMQ for secure MQIS access, and by Microsoft Internet Information Server (IIS) for secure hypertext transfer protocol (HTTP).

Performance Considerations When Securing MQIS Access

When you use certificates to secure MQIS access, the performance cost in establishing the connection is significant. However, actual MQIS communication is only slightly degraded. So, MSMQ-based applications that frequently establish secure MQIS connections will see greater performance degradation than will MSMQ-based applications that establish few secure MQIS connections but make more MQIS queries.

For more information on MSMQ controller servers and the MQIS database, see “The MSMQ Information Store” in Chapter 1, “Understanding MSMQ.”

Configuring Controller Servers for Secure MQIS Access

To configure controller servers for secure MSMQ access, you must:

- Install IIS version 2.0 or later
- Obtain a certificate from a certification authority.
- Use the IIS Key Manager utility to install and configure certificates for authenticating MQIS access.

If you do not want to use IIS on the MSMQ server, you can save disk space and memory usage by selecting only the **Internet Service Manager** check box when installing IIS. This installs approximately 2 MB of IIS utilities, none of which run as a service.

You can obtain a certificate from any certificate authority supported by IE version 3.0 or later. If the certificate authority is not supported by IE by default, you must add the certificate authority to the IE configuration. If you add the certificate authority to the IE configuration after you install MSMQ, you must then use the **MS Message Queue** icon in Control Panel to update the MSMQ certificate authority configuration. In either case, this change must be made on each MSMQ client or MSMQ routing server that will be configured to use secure MQIS access.

The key to successfully installing a certificate for securing MQIS access is to give the certificate the same name as the MSMQ routing server's Windows NT computer name. If these names do not match, MSMQ clients cannot establish secured connection with the server.. For complete instructions on installing and configuring a certificate for authenticating MQIS access, see "Securing Controller Server Communications" in Chapter 3, "Managing MSMQ."

Using the Same Certificate for IIS and MSMQ

Although you can install multiple certificates for use with IIS, you can install only one certificate for use with MSMQ (for the purpose of securing MQIS communication). If one or more certificates is already installed for use with IIS, and the common name of one of the certificates matches the name of the computer running Windows NT, you can run Key Manager and copy and paste, or you can cut and paste one certificate from the WWW node to the MSMQ node. If you copy and past the certificate, it will be used by IIS and MSMQ.

However, if the certificate's common name matches the server's domain name system (DNS) name instead of it's Windows NT computer name, the certificate cannot be used with MSMQ.

Each MSMQ client and MSMQ routing server that you configure to use secured MQIS access must be configured to trust the Certificate Authority that provided the certificates installed on the controller servers. This is covered in the following section.

Configuring MSMQ Clients and MSMQ Routing Servers for Secure MQIS Access

MSMQ Setup copies the IE certification authority configuration. Thereafter, the IE certification authority configuration and the MSMQ certification authority configuration are maintained separately, with one exception: When you run Control Panel, double-click **MS Message Queue**, click the **Security** tab, and then click the **Certification Authorities** button, any certification authorities added or removed from the IE certification authority configuration are added to, or removed from, the MSMQ certification authority configuration. However, the settings of the two certification authority configurations (which certification authorities are trusted by IE, and which certification authorities are trusted by MSMQ) are always maintained separately.

By default, the IE certification authority configuration trusts over a dozen certification authorities. If users don't change these settings, each MSMQ client and MSMQ routing server will trust the same certification authorities.

MSMQ client Setup automatically enables secure MQIS access if Setup can establish secure communication with an MQIS server. Each client can enable or disable secure MQIS access using the **MS Message Queue** icon in Control Panel.

For complete instructions on configuring MSMQ clients and MSMQ routing servers for secure MQIS access, see "Securing Controller Server Communications" in Chapter 3, "Managing MSMQ."

Securing MSMQ Files and Folders

On computers running Windows NT Workstation, Widows NT Server, or Windows NT Server, Enterprise Edition, all MSMQ message files, log files, and LQS files are secured on NTFS-formatted drives. Only members of the Administrators group on the local computer have full access to those files. All other users cannot access them.

By default, file security is set on the \Storage and \LQS folders so that each time a file is created in on of these folders, the file inherits the folder's security settings. However, if you move the files to a different folder using the **MS Message Queue** icon in Control Panel, these same security settings are not copied to the new folders (as you may want to use, or may already be using, the folder to store other files). Note that The files that are moved from the default folders to the new folders are moved with their default security settings. If you want to secure MSMQ files after using the **MS Message Queue** icon in Control Panel to specify new folders for storing the MSMQ message files, logger files, or transaction logger files, see "Changing the Security on MSMQ Files and Folders" in Chapter 3, "Managing MSMQ."

Simplified MSMQ Security

If you use MSMQ in a small office or in an environment that does not require security, you may wish to minimize MSMQ administrative tasks.

To install your MSMQ enterprise with minimal security

1. Install your PEC.
2. Run MSMQ Explorer.
3. Right-click on the PEC and then click **Properties**.
4. On the **Security** tab, click **Permissions**.
5. In **Name**, click **Everyone**.
6. In **Type of Access**, click **Full Control (All)**.

If you install additional sites (PSCs), BSCs, or MSMQ routing servers, select each one and give **Full Control (All)** access to **Everyone**.

You can also select each MSMQ independent client and give **Full Control (All)** access to **Everyone**.

Deploying MSMQ

The successful use of MSMQ-based applications depends on the successful installation of MSMQ dependent clients, independent clients, servers, and your MSMQ-based applications. The planned, staged, careful installation and testing of these is called *deployment*.

Creating the MSMQ deployment plan consists of:

- Planning your MSMQ enterprise
- Designing and building MSMQ-based applications
- Designing the rollout process

The first two topics are useful in planing any MSMQ enterprise, no matter how small. The last topic, “Designing the Rollout Process,” is the approach we suggest for mission-critical applications and cautious administrators.

This chapter is can be used as a step-by-step guide to planning your MSMQ enterprise and designing your rollout process.

Planning Your Enterprise

Assessing your needs and planning your enterprise accordingly is the first step in rolling out your MSMQ enterprise infrastructure and your MSMQ-based applications. To successfully deploy MSMQ, consider the following:

- Topology
- Dependent client and independent client configurations
- Server quantities
- Server hardware
- The Size of the information store
- Messaging performance with multiple disks
- Message queuing disk space requirements
- Windows NT Server domain issues
- Naming conventions
- Connectivity
- Integration
- Scaling up

Topology

You can make two MSMQ enterprise configurations decisions based solely on your network topology and protocol use. These are the definition of (1) Site boundaries and (2)Connected networks (CNs).

Sites

To plan your site layout, create a geographic profile that shows all the locations where your company has facilities. It may represent a small region or a large geographical area. Use a map or diagram to identify the number and types of users at each location. A visual representation helps you identify available network connections and network traffic.

Forward Compatibility

To ensure compatibility with the directory service used in future version of Windows NT Server, the following additional factors should be taken into consideration:

- Permanent connections
- Adequate available network bandwidth
- Security
- Administration
- Costs
- Performance
- Directory replication
- Organizational issues

Table 6.1

Windows NT Server consideration	**Requirement or suggestion**
Permanent connections	The network link must be permanent.
Adequate available network bandwidth	The bandwidth threshold must be able to support the volume of data transmitted within the site.
Security	Replicating Windows NT services must run under the same Windows NT security context so that they can authenticate each other. This includes core services such as Windows NT replication and routing, as well as other installed servers. In other words, replication services must belong to either the same Windows NT domain or to domains that trust each other.
Administration	Keep the number of sites to a minimum to ease administration tasks.
Costs	To control costs of expensive WAN links, place servers that are connected by an expensive link in separate sites.
Performance	The number of servers in each site affects site and server performance. More servers in a site will generate more network traffic throughout that site.
Directory replication	If automatic, frequent replication

	between servers is needed, place them in the same site (as long as they meet the necessary conditions described above). Directory replication occurs more often within a site than it does among sites.
Organizational Issues	Group people together that work on the same servers and sites to improve overall performance of the system, reduce network traffic, and reduce resource use.

Using Existing Domain or Site Mapping

MSMQ site boundaries should not be based on your Windows NT Server domain structure - each should be set according to the appropriate criteria. However, do not worry if the site boundaries and the domain structure happen to match.

Note With Microsoft Exchange, all servers in an organization are also grouped together into sites. In Exchange, a site is a group of servers that shares the same directory information and can communicate over high-bandwidth, permanent, and synchronous connections. If your organization uses Microsoft Exchange, and the sites defined for Exchange fall within the MSMQ site definition, use the same sites for both products.

Site Links

If your enterprise has more than one site, you must establish site links and their associated costs. Site link costs are simply relative numbers used to calculate the best inter-site message route. After you determine the number of sites in your enterprise, choose the appropriate values for the site links. In most cases, this is a fairly straightforward process.

For example, if you choose to base site link costs on delay (the speed of the link), you might use the following values.

Table 6.2

Line speed between sites	Site link value
100 Mbps	1
4 to 16 Mbps	2
1 to 4 Mbps	3
128 Kbps	4
56 Kbps	5

For information on how to set and change site link costs, see the PSC installation procedure in Chapter 2, Installing MSMQ or MSMQ Explorer Help.

Connected Networks

The key to establishing easy-to-use CNs is using a good naming scheme when your enterprise contains more than one CN. Clear, usable names will allow users to choose the correct CN.

For more information on naming schemes, see “Naming Conventions” later in this chapter. To review the definition of CNs, see “MSMQ Connected Networks” in Chapter 1, Understanding MSMQ.

Dependent Client and Independent Client Configurations

Determine whether you will used MSMQ dependent clients or independent clients, or both. The following table shows the advantages of each.

Table 6.3

MSMQ independent client advantages	MSMQ dependent client advantages
Does not require synchronous access to another computer running MSMQ.	Transfers resource (memory and disk space) loads to a server which can be more easily backed up or upgraded.
Can be installed on Alpha computers.	With fewer points of administration you have fewer computers to back up and fewer journal and dead-letter queues to monitor.
	Supports transactional messages on computers running Windows 95.
	Less resources are used if the dependent client is frequently offline and the server is not.

Server Quantities

As explained in Chapter 1, MSMQ uses four types of servers to control message queuing:

- Primary enterprise controller (PEC)
- Primary site controller (PSC)
- Backup site controller (BSC)
- MSMQ routing server

You install one PEC on an enterprise network, on one PSC for each additional site.

One or more BSCs should be installed in each site that runs MSMQ-based applications that are critical to your businesses success. These BSCs can provide failure recovery if a PSC fails. BSCs also provide load balancing for MQIS queries, although MSMQ write requests must still be handled by the site controller (PSC or PEC).

The use of additional MSMQ routing servers (remember that the PEC, each PSC, and each BSC also functions as an MSMQ routing server) depends on the number of MSMQ dependent clients in your MSMQ enterprise and your use of session concentration. If your mobile independent clients don't use MSMQ-based applications while they are disconnected from the network, and you don't plan on using MSMQ session concentration, you probably don't need to install additional MSMQ routing servers.

Tip Installing additional BSCs and MSMQ routing servers is very easy, and does not require changes to your MSMQ enterprise settings (unless you are using session concentration). Because it is difficult to predetermine exactly how many BSCs and MSMQ routing servers your enterprise will require, start with a conservative number, but be prepared to add more as your message queuing load increases.

For best performance, do not install MSMQ servers on primary domain controllers (PDCs) or a backup domain controller (BDCs). PDCs and BDCs perform the resource-intensive tasks of maintaining and replicating the network accounts database and performing network login authentications.

Planning for Mobile Independent Clients

Mobile users who run MSMQ-based applications while disconnected from the network place an additional load on the system, as they attempt to send or receive a large number of messages when they reconnect to the system. This temporary load can be transferred to additional MSMQ routing servers to prevent reduced performance.

You might choose to have mobile independent clients assigned to use specific MSMQ routing servers (using InRSs and OutRSs) to evenly distribute the load across several servers. Because InRS and OutRS settings are not used when independent clients connect to the MSMQ network from sites other than their original site, this configuration works best if the independent clients do not connect to the MSMQ network at other sites.

Planning for MSMQ Dependent Clients

MSMQ dependent clients require synchronous access to an MSMQ server. Each additional MSMQ dependent client places additional memory, disk space, and network loads on the server.

Planning for Session Concentration

Session concentration can be used to:

- Reduce sessions across slow, expensive links
- Reduce sessions between computers within a site

Reducing sessions can minimize network bandwidth use, improve sever performance, and save time an money (when fewer sessions are being established across a slow or expensive links).

InRSs and OutRSs provide intra-site session concentration by controlling the flow of messages within sites. Site gates provide intra-site session concentration by controlling the flow of messages between sites.

Session concentration places an additional load on MSMQ servers. To compensate for this increased load, you may need to install additional MSMQ routing servers.

In a small enterprise, a PSC or the PEC could function as the InRS or OutRS for each MSMQ independent client in the site. However, to provide redundancy, each independent client should also be configured to use the site BSC as an InRS and/or OutRS.

Server Hardware

Your hardware decisions will affect the performance of your MSMQ enterprise. When selecting equipment, carefully consider the following hardware issues:

- I/O subsystem
- Memory needs
- Processor (CPU) needs
- Network adapters
- Distributing or concentrating servers
- Planning for Growth

Note If the server also runs other services, such as mail or Internet services, additional resources may be need, or the server may support fewer MSMQ dependent clients.

Number of Disks

MSMQ controller servers (the PEC, PSC, and BSCs) can benefit from two or more physical disks. For optimal MQIS performance, one disk should be used for the MQIS data device, and the other for the MQIS log device. The log devices may be created on separate drives when you install the controller servers. However, once the MSMQ server is installed you cannot move the data or log devices.

For optimal messaging performance on MSMQ servers, put the messaging files, message log files, and transaction files on separate physical disk. For more information, see “Messaging Performance” later in this chapter.

Disk Speed

Faster disks also increase the processing of recoverable messages and those express messages that have been written to disk when memory resources are in demand.

Microsoft recommends that you choose a server that has a caching disk controller with a high-speed bus interface such as PCI SCSI®. These controllers provide optimal performance for MSMQ servers. Most of these controllers support hardware disk striping, which offers greater performance and less CPU use than Windows NT Server software striping.

Disk Size

To determine the minimum disk size for your servers, consider the following:

- The total number of messages likely to accumulate at the server at any one time and the size of the messages.
- The disk space used by the MSMQ files and potentially, the MSMQ installation folder.
- The size of the MQIS database.

Note When an MSMQ independent client or server runs out of disk space it can no longer send or receive messages, nor can it create queues.

For more information on determining the appropriate size of the MQIS for your enterprise, see “Determining the Size of the Information Store” later in this chapter. For more information on determining disk space requirements for your message queuing, see “Determining Message Queuing Disk Space Requirements” later in this chapter. For more information on the error messages returned when a computer has insufficient disk space to send or receive messages, see the Microsoft Message Queue Server Software Development Kit (MSMQ SDK) documentation.

Disk Partitioning

Because the Windows NT file system (NTFS) provides more reliable file recovery after a power failure or system crash, it is the recommended file system for MSMQ servers. Using three or more disks formatted with the NTFS will allow you to use Windows NT Server software RAID (redundant array of inexpensive disks) support for fault tolerance.

Memory Needs

For optimal performance, your MSMQ servers should have enough physical memory to avoid heavy use of the Windows NT paging file. Because accessing the paging file is much slower than accessing physical memory, increase the amount of physical memory can greatly improve performance. However, other hardware resources, such as the CPU or I/O subsystem, may become the limiting performance factor.

Processor (CPU) Needs

If your I/O subsystem is fast and is not the single limiting performance factor, and if you have enough memory to handle MSMQ messaging without excessive use of the page file, a fast processor improves performance. For MSMQ servers, you may also want to consider a RISC processor. MSMQ supports both Intel-compatible and Alpha platforms for MSMQ servers.

Network Adapters

High-speed network adapters and network drivers optimize MSMQ server performance. High-speed network adapters can achieve high throughput rates with low CPU use.

Because many dependent clients and independent clients will simultaneously access the server, you should install one or more high-performance network adapters, based on a high-speed bus such as PCI or EISA. Installing multiple adapters in a multiprocessor server means that a single server can handle more requests, with each adapter servicing a separate network segment or protocol.

You can use two network adapters to provide fault tolerance. To do so, associate the CNs supported by the computer with both network adapters.

Distributing or Concentrating Servers

Hardware that can be used with MSMQ ranges from single-processor 486 computers to the most powerful configuration that Windows NT Server supports. Therefore, you could have an organization of 10,000 users, with 100 users on 100 servers, or with 500 users on 20 servers. This decision depends on your organization's needs and budget.

The following table lists the advantages and disadvantages of using more, less powerful computers.

Table 6.4

Advantages	Disadvantages
It's less expensive to increase capacity.	There is more hardware to maintain.
There are more hardware choices available.	Customized hardware increases support costs.
Each server can be physically close to the users, reducing network traffic for dependent client/server interactions.	Minor increases in users or load require more hardware.

The following table lists the advantages and disadvantages of using fewer, more powerful computers.

Table 6.5

Advantages	Disadvantages
The system is designed for upgrades.	Network adapters must be able to handle more traffic to each server.
There are less storage requirements.	
Fewer replication changes are required.	

Planning for Growth

When choosing hardware for servers, consider future needs and choose servers that can be easily expanded. As the number of users and connectors in a site grows, you can reconfigure the existing hardware rather than add more servers. There are several ways to make the most of existing server hardware for MSMQ:

Increase the amount of system memory for each server.

- Upgrade the processors on existing servers or use multiprocessors in existing servers.
- Increase the number, speed, and size of the disks or disk arrays.
- Add more disk drives to the volume set.
- Replace disk drives and volume sets with striped drive arrays.
- Limit the number of non-MSMQ software processes on the server by moving them to other computers.

By planning ahead, you can choose hardware that is easily upgraded. If you identify a bottleneck after installation, you can upgrade rather than purchase a new server or change the site layout.

Choosing Between the Full and Limited versions of SQL Server

MSMQ controller servers (the PEC, PSCs, and BSCs) use a Microsoft SQL Server version 6.5 database to store the MQIS. You can purchase SQL Server 6.5 for each MSMQ controller server, or you can install a limited version of SQL Server 6.5 when you install each controller server.

The MSMQ limited version of SQL Server version 6.5 is limited to a cumulative total of 100 MB in database files, which leaves, at most, 87 MB for a maximum MSMQ database (with 13 MB left for the logging of MSMQ database updates). If your enterprise outgrows the limited version of SQL Server 6.5, you can purchase SQL Server 6.5, upgrade your existing MQIS installations, and increase the size of the MQIS database.

Determining the Size of the Information Store

By default, MSMQ creates 50-MB MQIS database. This database uses a 50 MB file for data (called the data device in SQL terminology) and an 8 MB file for logging database transactions (called the *log device* in SQL terminology). In general, the log device should be 15 percent of the size of the data device.

The default 50 MB MQIS allows for approximately 3800 users and 3800 computers (with four public queues each), or 2000 users and 2000 computers (with 10 public queues each). If computers in your enterprise will not be sending authenticated messages, the default MQIS supports approximately 4800 computers with four public queues each, or 2200 computers with 10 public queues each.

Using the following formula, you can calculate the approximate MQIS size required for your enterprise:

computers * (2.4K+ (2K * public queues per computer)) + 2.6K * authenticated users

Use the following table to determine the appropriate values for your enterprise.

Table 6.6

Formula item	Value to use
computers	Number of independent clients and servers in your MSMQ enterprise. (No need to count MSMQ dependent clients).
Computer object size	2.4 K
Public queues per computer	The average number of public queues on each computer
authenticated users	Number of users that will send authenticated messages

Notes The computer object size increases by 2 K for each user account that you assign more than 8 access rights to (instead of assigning access rights to a group).

Because private queues are stored locally (on each computer), they need not be considered when calculating the required MQIS size.

The MSMQ site object and other internal MSMQ tables also require some MQIS storage space, but their size is relatively insignificant.

If the 60 MB limit of the limited version of SQL Server 6.5 is not adequate for your enterprise, you must obtain a retail release of SQL Server 6.5.

Because the MQIS is replicated between all controller servers, you should use the same size data and log devices on all MSMQ controller server within your enterprise.

If you need to increase the data or log devices after you have installed MSMQ, you can do so using Microsoft SQL Enterprise Manager. For more information, see “Increasing the Size of the MQIS Database” in Chapter 3, Managing Your MSMQ Enterprise.

Improving Messaging Performance with Multiple Disks

MSMQ independent clients and servers with multiple physical disks can improve their messaging performance by storing the various files used to implement messaging across multiple physical disks. MSMQ can use up to three different drives to implement message queuing. Folders on the different drives are used to store the message files (.mq files), message logger files, and the transaction logger files.

If you have only two disk drives, put the messaging files on one drive and the message logger and transaction logger files on the other drive.

For optimal transactional performance, put the MSMQ message files, message logger files, and the transaction logger files on three different drives, and put the data and log files for the application you are transacting with on two or more different drives.

Adding additional disk drives ensures that disk access has a minimal impact on messaging performance. However, if you CPU usage is at or near 100 percent, adding additional disk drives will not improve messaging performance. To determine if your computers are limited by disk access or processing power, use Performance Monitor to track the **% Processor Time** and **Avg. Disk Queue Length** counters. If sustained CPU use is above 75% while sending messages, adding additional processing capacity may improve messaging performance. If the average disk queue length for any drive is greater than 0.6 while sending messages, additional drives may improve messaging performance.

Determining Message Queuing Disk Space Requirements

When calculating the required disk space for servers, consider the maximum number of messages likely to be stored on the server at any one time. Because messages may be stored on target computer, source computer, and any MSMQ routing server in between the two, it is difficult to anticipate message storage requirements. However, if you configure MSMQ independent clients to use InRSs or OutRSs, or sites to use site gates, you will need to allow for additional message space on those MSMQ servers. Also, those servers that support mobile independent clients will likely need additional disk space.

Because journaled messages are stored until an administrator removes them, or application reads the journal queue, you must allow for additional disk space when journaling messages.

The amount of disk space used for messages depends on the message type (recoverable or express), message size, and messaging volume. Recoverable messages are allocated in 4K blocks. Each message using approximately 100 bytes of MSMQ overhead (depending on the message properties). This yields approximately 256 messages per megabyte (MB) of disk space for messages under approximately 4,000 bytes.

Because express messages are written to disk when insufficient memory is available to store them all, you must also allow for their disk space usage. Because express messages are stored in 64 byte blocks, disk space is used more efficiently. For example, you can store approximately 1,280 1,024-byte recoverable messages in 5 MB, or you can store approximately 5,120 1,024-byte express messages in 5 MB.

Note MSMQ allocates disk space for both express and recoverable messages in 4 MB files. This ensures that enough disk space will be available when memory is flushed, and messages are written to disk.

As a rule, you should base your message type selection on business needs, not disk space requirements. For more information on MSMQ message types, see “MSMQ Message Delivery,” in Chapter 1, Understanding MSMQ.

Windows NT Server Domain Issues

To control access to MSMQ objects, MSMQ computers must log on to a Windows NT Server domain. If you choose not to use access control, MSMQ dependent clients, independent clients, and servers need not log on to a Windows NT Sever domain.

For more information on Windows NT domain trusts and their impact on MSMQ installations, see “Windows NT Domain Considerations” in Chapter 5, “Securing Your MSMQ Enterprise.”

Naming Conventions

You must provide names for the enterprise, sites, connected networks, and folders when you install and configure MSMQ. Carefully planned naming conventions make it easy for you to add sites and connected networks. You should choose names for your sites and CNs that won't be affected by organizational changes. The following table lists some naming conventions used by the hypothetical company Volcano Coffee.

Table 6.7

Element name	Naming convention
Enterprise	Company name; for example, Volcano Coffee
Site	Geographical location; for example, Seattle Metro Area.
Connected Network	Company, protocol, and if necessary, location: for example, Volcano IP (North America)

The Enterprise and CN names can be up to 126 characters long, and can contain any characters. Site names can be up to 31 characters long, and can contain any characters.

The enterprise name, site names, and CN names can be changed from MSMQ Explorer. However, server names are defined when you install the operating system, and cannot be changed from MSQM Explorer.

Connectivity

If you use the IPX with MSMQ, ensure you:

- Define unique non-zero internal network numbers for computers with multiple network adapters
- Provide IPX name resolution

Unique Non-zero Internal Network Number

If computers within your enterprise have multiple network adapters and you are using IPX on some of the adapters, you must define a unique non-zero internal network number for those computers. If you do not, you will likely have problems using the MSMQ RPC transport over IPX.

To configure the IPX/SPX-compatible protocol on a computer running Windows 95

1. Run Control Panel and double-click **Network**.
2. Click **IPX/SPX-compatible protocol** and click **Properties**.
3. In the **Property** box select **Network Address** and then define a unique non-zero internal network number in the **Value** box.

To configure the NWLink IPX/SPX compatible transport on a computer running Windows NT

1. Run Control Panel and double-click **Network**.
2. On the **Protocols** tab, click **NWLink IPX/SPX compatible transport** and then click **Properties**.
3. define a unique non-zero internal network number in the **Internal Network Number** box.

IPX Name Resolution

MSMQ requires name resolution on IPX networks. If no Novell server is available on your IPX network, you can use the Windows NT Server SAP Agent to provide name resolution. The Windows NT Server SAP Agent should be installed on all MSMQ servers using the **Network** icon in Control Panel.

Integration

Level 8 Systems provides a product that supports the MSMQ APIs on many platforms including IBM MVS and CICS; Sun Solaris, HP-UNIX, and AIX UNIX platforms; as well as OS/2; VMS; and AS/400 platforms. The Level 8 Systems product also supports the mapping of native IBM MQSeries API (MQI) calls to the MSMQ API to provide seamless message exchange between IBM MQSeries applications and MSMQ-based applications.

For more information on Level 8 Systems products and product availability, contact Level 8 Systems, or see the Level 8 Systems web page at http://www.level8.com/.

Scaling Up

When planning your MSMQ enterprise, consider your companies potential growth, and possible increased use of MSMQ. By carefully selecting your MSMQ hardware, and correctly installing your MSMQ enterprise, you can decrease the effort needed to later increase the scale of your MSMQ enterprise.

As your use of MSMQ increases:

- PSCs and BSCs need more disk space to store MQIS data (sites, CNs, computers, and queues)
- MQIS activity increases as more queues are created and deleted and other objects are modified more often.
- Messaging traffic increases

You can minimize effort needed solve these problems by installing the PEC and PSCs on computers that can be expanded to provide more computing power and network requests. When you determine you need to increase your message queuing capacity you can:

- Upgrade the PEC and PSCs CPUs and add additional network adapters.
- Installing additional BSCs to provide load balancing for MQIS queries.
- Installing InRSs, OutRSs, and site gates to steer network traffic into wider-bandwidth network links. However, to avoid single points of failure in your MSMQ enterprise, configure independent clients to use multiple InRSs and OutRSs, and configure sites to use multiple site gates.

Designing and Building MSMQ-based Applications

To successfully rollout MSMQ you must design, build, and deploy useful MSMQ-based applications. The MSMQ APIs, as well as programming examples, are documented in the MSMQ SDK documentation. This documentation can be installed when you install MSMQ independent clients and server, or anytime thereafter by running MSMQ Setup. The MSMQ SDK documentation is installed on MSMQ independent clients when you choose **Development Workstation**, and on MSMQ Servers when you click **Custom**, and then select **SDK**.

The MSMQ SDK documentation contains:

- Background information on MSMQ topics.
- Descriptions of each function, ActiveX component, property, and structure that may be needed to develop your application.
- Sample code for several types of MSMQ-based applications.

The sample applications can also be used to see how an MSMQ application can be developed. Although the samples may not fit your exact needs, they do provide the basic framework for different types of applications.

Designing the Rollout Process

This section contains one approach to deploying MSMQ in a medium to large corporate environment. This approach is planed, calculated, and cautious. It is intended to be used as a model for deploying MSMQ and mission-critical MSMQ-based applications. You must choose an approach and plan for your business that meets your business needs.

Designing the rollout process consists of:

- Setting the administrative policy
- Preparing the teams
- Conducting lab tests
- Conducting one or more pilot rollouts
- Conducting the final rollout

Setting the Administrative Policy

Your enterprise plan is a working document. How it is implemented and changed can be influenced by the policies that you establish. You need to do the following:

- Establish permissions within the enterprise
- Delegate administrative duties
- Document work
- Prepare for emergencies
- Schedule regular backups

You should establish administrative policies for emergencies such as a server being corrupted, removed, or destroyed. With emergency plans, you can lessen the severity of disruptions.

You should define roles and tasks for the people administering MSMQ. You can give one administrator permissions for the entire enterprise, another administrator permission for individual sites, and another administrator permission for individual servers. You can also give an administrator permission to view, but not change the MSMQ enterprise, its elements, and its configuration.

To reduce administrative duties, you can also create a Windows NT Server global group that includes the Windows NT user accounts for all MSMQ server administrators in a site. Membership in this group gives a user all the permissions granted to the group. You can add or remove members. You can also add or remove the permissions assigned to the entire group, rather than to each user account.

If you are using multiple domains within a site, create this group in the domain where you centralize your administrative functions for the other domains. In a single master or multiple master domain model, this group and all the user accounts for MSMQ administrators should be created in the master domain.

Preparing the Teams

This phase involves gathering the resources, including equipment, software, and staff, to properly plan for testing and evaluating MSMQ. Members of the support team should receive training during this phase.

Acquiring Staff and Software

The deployment project manager participates in the executive team and leads the planning team. This individual is usually the head of the Information Systems department.

When setting up the planning team, try to include individuals from the various groups involved in the deployment process. This includes people from the Corporate Support and Employee Training departments, the Corporate Standards Committee, and key installation team members. Individuals from the Finance and Accounting group will need to take part in planning and evaluation later on, but need not be assigned to the team for the full duration of the deployment process.

Obtain sufficient MSMQ and SQL Server licenses to develop and test MSMQ and your MSMQ-based application.

Testing Lab Setup and Equipment

To effectively evaluate and test the MSMQ installation process, and the performance of MSMQ and your MSMQ-based application, you need to set aside sufficient physical space. Your testing lab should accommodate a sufficient number of computers to test everything from server-based Setup, your MSMQ enterprise configuration, and performance of your MSMQ-based application. In addition, if your MSMQ-based application will be running on portable computers that dial in to the company, or interacting with mainframe computers, you need to make sure that the lab computers have full access to the network and an analog phone line.

It is important that you test and implement all of the MSMQ features that your MSMQ-based application will use.

Training the Teams

By reviewing this book, the installation and planning teams can gain an extensive understanding of MSMQ features, administration procedures, and installation procedures.

Support team members must become familiar with your MSMQ-based application.

Conducting Lab Tests

Lab testing allows you to both test your MSMQ-based application design and determine what type of load it puts on your MSMQ servers.

Preparing for the Lab Test

Preparing the test site and equipment involves the following tasks:

- Ensuring that the computers not only meet the minimum MSMQ requirements, but meet your message queuing needs.
- Ensuring you MSMQ-based application works correctly.

Installing MSMQ in the Lab

Before setting up MSMQ in your lab, verify that each computer running Windows 95 and Windows NT is working properly. Then, using the instructions in Chapter 2, Installing MSMQ, install a small-scale enterprise to mirror your planned enterprise. For example, if you plan to have a total of three sites, each with two BSCs and two additional MSMQ routing servers:

- Create three sites in your lab by installing one PEC and two PSCs.
- Establish site link costs based on your enterprises actual site links.
- If you plan to use site gates, configure the site controllers as site gates
- Install the MSMQ routing servers.
- Have representative users install a few dependent clients or independent clients in each site.
- If you plan to use InRSs, OutRSs, configure the independent clients to use the InRSs and OutRSs

Evaluating the Installation Process

Evaluate how smoothly the installation process went. Ask questions that will help improve your ability to roll out MSMQ and your MSMQ-based application:

- Were there any problems?
- Were the team members properly trained?
- Did the representative users have enough information to install the dependent client or independent client software? Did they have any questions?

Use the information you gained from the lab rollout to improve your rollout plans.

Stressing MSMQ in the Lab

Using either a modified or automated version of your MSMQ-based application, or a specially written load-testing program, simulate the load you plan to place on the MSMQ servers. Evaluate the results to determine if your MSMQ enterprise can meet your business needs. Ask questions that will help evaluate and improve your MSMQ enterprise design. For example:

- Does your MSMQ enterprise adequately handle the message volume?
- Do site gates, InRSs, or OutRSs cause bottlenecks?
- Are there an adequate number of BSCs and MSMQ routing servers?
- Did you cause some servers to fail?
- Did redundant or backup servers handle the messaging load?

If necessary, use the information you gained from the lab stress tests to improve your MSMQ enterprise plan, or improve or redesign your MSMQ-based application.

Conducting a Pilot Rollout

Once you have tested and refined your rollout procedures in the lab, you are ready to test them in the production environment. The pilot rollout introduces the variables of user reactions and the activity in the production network, but on a limited scale−usually 15 to 50 users. Based on this new information, you might make adjustments to your plan and or MSMQ-based application for the final rollout. If you make extensive changes to the final rollout plan or your MSMQ-based application, consider doing a second pilot rollout.

Planning the Pilot Rollout

This phase involves three major efforts: testing your installation procedures for MSMQ and your MSMQ-based application, documenting the logistics of the pilot installation, and preparing the user training plan. These efforts are a combination of planning and lab-testing work that involves:

- Developing user training
- Developing the administrative plan
- Developing the support plan
- Developing the administrative plan
- Notifying users of the rollout

Developing User Training

The first steps in developing a training plan are to acquire a training lab, set up computers in the lab, and appoint a team member as instructor. If in-house resources are not available, use a vendor to develop and conduct the training. The instructor will be responsible for creating and testing the training program.

Developing the Administrative Plan

Developing an administrator training plan is similar to developing a user training plan. Even if the deployment team members will be doing all the MSMQ related administration, you will likely need to have an administrator training plan for backup administrators, and to train replacements as the MSMQ administrators move on to other jobs.

Developing the Support Plan

Similar to the training plan, the support plan must be ready to go online the first day you begin installing MSMQ and your MSMQ-based application. Because the quality of support that's available during the pilot rollout will be seen as an indicator of the quality of the rollout as a whole, it is important that you plan carefully to make sure effective support is available.

Developing the Administrative Plan

Developing the administrative plan is similar to developing the support plan. Your MSMQ and MSMQ-based application administrators must be trained, and understand their rolls and responsibilities the first day you begin installing MSMQ and your MSMQ-based application. Because the quality of administrative support that's available during the pilot rollout will be seen as an indicator of the quality of the rollout as a whole, it is important that you plan carefully to make sure the MSMQ infrastructure and MSMQ-based application are administered effectively.

Notifying Users of the Rollout

Another step at this stage is informing users about the pilot rollout plan. You can use a videotape presentation, an interoffice memo, or a company meeting as the means for communicating with user about the rollout. Regardless of the form used, the message must explain to users the benefits of moving to your new MSMQ-based application and describe the overall plan and process by which each group or department will make the move. This makes it easier for you users to plan for and accept the migration to your new MSMQ-based application as part of their schedules.

Conducting the Pilot Rollout

This phase consists of simulating the final installation process, testing the capabilities and performance of the system, surveying user feedback, and making adjustments as needed.

Simulating the Installation Process

The schedule for the pilot rollout should simulate—on a smaller scale—the schedule for the final rollout. As you conduct the pilot rollout, you might find that certain tasks take more or less time than expected, that some tasks need to be added, or that some tasks can be left out. Modify the pilot rollout schedule to account for such changes, and use the pilot schedule for projecting the final rollout timetable.

Testing MSMQ Performance and Capabilities

In addition to the technicians responsible for conducting the pilot installation, extra technicians should be assigned to measure, observe, and test the installation. By tracking the time per installation, handling problems that arise, and identifying areas for improvement or automation, these individuals help ensure the success of both the pilot and final rollouts by making the installation more efficient.

In addition, After MSMQ and your MSMQ-based application are installed, these technicians test system capabilities for proper operation and monitor the dependent client and independent client computers for performance, stability, and functionality, highlighting any inconsistencies with the lab configuration.

Surveying Users for Feedback

The final part of the pilot rollout involves surveying the users to gauge their satisfaction and proficiency with the new installation and to evaluate the level of training and support provided. Test users' proficiency by having them perform a few common tasks supported by your MSMQ-based application.

When collected, combine the survey results with the ideas for improvements identified during the pilot rollout. Use this information to prepare a checklist of open issues that must be resolved before the final rollout. Then assign team members to take the actions necessary for solving problems or making improvements. Indicate on the checklist how and when each item was resolved, adjusting the deployment plan if appropriate.

Conducting the Final Rollout

After analyzing the results of the pilot rollouts, you are ready to finalize the plans for the full-scale rollout. Your team and your users can then enjoy a well-planed and well-organized full deployment of MSMQ and your new MSMQ-based application.

Finalizing the Rollout Plan

The final rollout plan is an extension of the pilot planning process, with the added steps of documenting, budgeting for, and carrying out the final logistics.

As you prepare for final rollout, estimate the length and scope of the overall installation process. Also plan for all tools needed to complete the process within the stated timeframe. If necessary, propose a formal budget for the company-wide implementation, and present it to management for approval. Your budget should include the costs for development, administrative, and hardware resources.

After obtaining any necessary approval, purchase the resources required to facilitate the installation. If you need additional staff, be sure to hire experienced and qualified individuals for the team, and train them extensively before getting started.

Complete your training, communication, and staffing plans for the final rollout at this time.

Rolling Out MSMQ

Following weeks of planning, organization, development, testing, communication, and training, the deployment teams and your organization as a whole should be ready for full-scale rollout of MSMQ and your new MSMQ-based application. The extensive preparation for this event might make deployment seem almost routine for the teams involved; however, that's exactly the kind of uncomplicated rollout a systems administrator dreams of.

Troubleshooting

This appendix contains information on MSMQ and MSMQ tools that can help you troubleshoot problems. This appendix covers:

- Viewing MSMQ error codes
- Troubleshooting setup
- Troubleshooting security problems
- Troubleshooting independent client/site issues
- Troubleshooting MQIS purge problems
- Dumping the MQIS database

For information on MSMQ support, see Microsoft Technical Support in the Welcome section at the front of this book.

Viewing MSMQ Error Codes

MSMQ error codes are documented in the Reference section of the Microsoft Message Queue Server Software Development Kit (MSMQ SDK) Help. The error codes section contains a list of all the error codes. In addition Each API function reference page has a list of the return codes that can be returned by to the function. (for example, MQOpenQueue has a section that contains all the error codes that can be returned by MQOpenQueue).

Troubleshooting Setup

the following information may be used to troubleshoot MSMQ installation problems.

Insufficient Permissions to Install MSMQ

If you receive “insufficient permission” errors when installing MSMQ, you may not have:

- Administrative rights on the local computer
- Appropriate MSMQ permissions
- SQL database administrative rights (if you are installing a PEC, PSC, or BSC)

Administrative rights on the local computer are required to install MSMQ on computers running Windows NT. The appropriate MSMQ permissions are required to install MSMQ servers and MSMQ independent clients. SQL database administrative rights are required to install MSMQ controller servers (PEC, PSCs, or BSCs).

For information on granting users the correct permissions to install MSMQ clients and servers, see “Permissions Required to Install MSMQ” in Chapter 5, “Securing Your MSMQ Enterprise.”

Database Creation Errors

If PEC, PSC, or BSC installation fails due to database creation problems:

- Ensure you restarted Windows NT after installing and configuring SQL Server.
- Check for required disk space /drive
- Check SQL database configuration

Server Connection Errors

If Setup displays a connection error, or indicates that it cannot find the controller server, ensure that the Microsoft Message Queue Service is running on the PEC or PSC.

Unexpected or Inexplicable Setup Results

Setup always searches for an Msmqinst.ini file, even if you are installing MSMQ from the MSMQ CD. If Setup unexpectedly reads default settings from Msmqinst.ini, your MSMQ dependent client, independent client, or server can be installed with settings you did not specify.

Setup searches for this file in several different locations, depending on the operating system you are running. Setup uses the first Msmqinst.ini file it finds.

On computers running Windows 95, Setup first searches for the Msmqinst.ini file in the following places:

1. The Windows folder
2. The default folder (the folder from which you are running Setup)

On computers running Windows NT Workstation or Windows NT Server, Setup first searches for the Msmqinst.ini file in the following places:

1. The Windows folder
2. The HKEY_LOCAL_MACHINE\Software\Microsoft\
   Windows NT\CurrentVersion\IniFileMapping key in the registry.
3. The default folder

Problems Reinstalling MSMQ

Two problems can occur when you attempt to install an MSMQ dependent client on a computer that was previously used as an MSMQ independent client or server.

DLL Conflict When Remove All Fails

On computers running Windows NT, if you install an MSMQ independent client, server, or the MSMQ RAS connectivity service, remove the MSMQ installation, and then install an MSMQ dependent client, the dependent client may not function correctly.

This problem occurs when a previous MSMQ installation was not completely removed prior to installing an MSMQ dependent client. This problem occurs because the MSMQ dependent client installation and the other installations use different .dll files with identical version numbers.

To work around this conflict, run MSMQ Setup and remove the current installation. If Setup does not recognize the current installation, and does not give you the option to **Remove All**, manually delete Mqdscli.dll, Mqutil.dll, and Mqrt.dll from your System32 folder (typically c:\Windows\System32 or c:\Winnt\System32). You should then be able to successfully install an MSMQ dependent client.

MS DTC Conflict When Installing an MSMQ Dependent Client

You cannot install an MSMQ dependent client if MS DTC is installed. Because MSMQ independent client and server Setup installs MS DTC, but does not remove it when you click **Remove All**, you cannot install an MSMQ dependent client without manually removing MS DTC.

Note Do not remove MS DTC if it is used by another application, such as Microsoft SQL Server, or Microsoft Transaction Server.

To manually remove MS DTC

1. Run MSMQ Setup and click **Remove All**.
2. Type the following commands at a Windows 95 MS-DOS prompt or a Windows NT command prompt:

```
Net stop msdtc
Msdtc -remove
```

Troubleshooting Security Problems

You can Use the following information to troubleshoot MSMQ security implementation problems.

Internal Certificate is Unusable After Uninstalling SP 2

If you uninstall Windows NT Service Pack 3 and then install Windows NT Service Pack 2, your internal certificate is unusable.

To work around this problem:

1. Run Control Panel and double-click **MS Message Queue**.
2. On the **Security** tab, click **Remove**.
3. Select your personal certificate and then click **Remove**.
4. Click **Register**.
5. Select a certificate and click **OK**.

Troubleshooting Independent Client/Site Issues

You can Use the following information to troubleshoot MSMQ independent client site connectivity issues.

Automatic Site Recognition

On MSMQ independent clients, the MSMQ service sends out a broadcast when it starts, and monitors all replies. If a site controller other than the independent client’s current site controller replies, and if the new site controller can communicate with the site controller in the independent client’s original site, the independent client connects to the new site.

If you have more than one site controller in a broadcast segment (for example, in an MSMQ lab) more than one site controller can reply to the independent client’s broadcast. If one of the site controllers that replies is the controller for the independent client’s current site, the independent client does not connect to another site.

If the site controller for the independent client’s current site is offline, the independent client does not move to a new site.

Control Panel Does Not Display Current or New Sites

When you run Control Panel on an MSMQ independent client, double-click the **MS Message Queue** icon, and view the **Mobile** property page, the **Current Site** and the list of new sites may be blank. This situation occurs when you view the **Mobile** property page immediately after installing the independent client because the MSMQ service doesn’t create these entries in the registry until two minutes after the MSMQ service starts. The entries are then updated every 12 hours.

To work around this situation, select anther property page, wait 2 minutes, and then return to the **Mobile** tab.

MSDTC Gets Uninstalled

If you install an MSMQ independent client on a computer that also has SQL Server version 6.5 installed, and then you uninstall SQL Server, MSDTC is also uninstalled.

To correct this problem, reinstall the MSMQ independent client.

Troubleshooting MQIS Purge Problems

The MQIS database maintains some information about deleted objects. To avoid wasting MQIS database space, an MQIS purge algorithm periodically deletes this information when it is no longer needed. Information about deleted objects is deleted when the information has been replicated to the PEC and all PSCs and it is more than one week old.

If problems occur during the purge of the MQIS database, one of the following error messages is written to the Windows NT application log:

- Cannot perform purge up to sequence number *x* because PSC *y* did not ACK. Check for connectivity with the PSC. Last ACK was at sequence number *z*.

  This error indicates that the MQIS controller could not purge the database because a PSC named *y* did not acknowledge. Make sure the server has connectivity with the PSC. If a PSC is not connected, the purge algorithm cannot complete.
- Did not receive an ACK from BSC *x* for a few days. Make sure to reconnect the BSC, or uninstall it if it is obsolete.

  This error indicates that a PSC did not get an acknowledgement from its BSC within four days.
- MQIS database reached *x* percent of its capacity. Consider expanding it.

  This error appears when the MQIS database usage reaches 70% of capacity. If you are using the full retail version of SQL Server 6.5, you can increase the size of your MQIS database. If you are using the limited version of SQL Server 6.5 included with MSMQ, you must upgrade to the full release version of SQL Server version 6.5 to increase your database size.

  For more information increasing the size of the MQIS database, see “Increasing the Size of the MQIS” in Chapter 3, “Managing Your MSMQ Enterprise.”

Dumping the MQIS Database

DumpIS is a command line utility that can be used to dump the contents of the MQIS database. MSMQ server Setup installs DumpIS when you click **Server** or **Installation Server**, or if you click **Custom**, and leave **Administration Tools** selected. DumpIS cannot be installed on MSMQ independent clients.

For instructions on using DumpIS, type **dumpIs /?** at the Windows NT command prompt.

APPENDIX B

Installing and Configuring the MSMQ Exchange Connector

This appendix contains information on:

- Using the MSMQ Exchange connector
- Installing the MSMQ Exchange connector
- Managing an MSMQ Exchange connector
- Monitoring an MSMQ Exchange connector
- Removing an MSMQ Exchange connector

For information on developing MSMQ-based applications that operate with Microsoft® Exchange, see the MSMQ SDK.

Using the MSMQ Exchange Connector

Exchange users can use the MSMQ Exchange connector to send messages and forms to MSMQ queues. MSMQ-based applications can use the MSMQ Exchange connector to send messages to Exchange users. Exchange software does not have to be running on other MSMQ dependent clients, independent clients, and servers that send messages to Exchange clients, nor does MSMQ software need to be running on Exchange client computers.

The MSMQ Exchange connector is compatible with Microsoft Exchange version 4.0a (Exchange 4.0 with Exchange Service Pack 2 installed), and Exchange version 5.0.

Note The references to *Exchange clients* in this appendix are generic. These references apply to the Microsoft Exchange client provided with Microsoft® Exchange Server, Windows® 95, Windows NT® Workstation version 4.0, and Windows NT Server version 4.0. The references also apply to Microsoft® Outlook™, included with Microsoft® Office 97.

For information on creating MSMQ-based applications that use the MSMQ Exchange connector, see the MSMQ SDK.

Sending Messages from MSMQ to Exchange

As shown in the following Figure, an MSMQ-based application can send an e-mail message to an Exchange user by creating an MSMQ message and sending it to the MSMQ Exchange connector queue. The MSMQ Exchange connector then reads the MSMQ message from the queue, parses the contents, creates an Exchange message or form, and sends the message or form to the appropriate Exchange recipient.

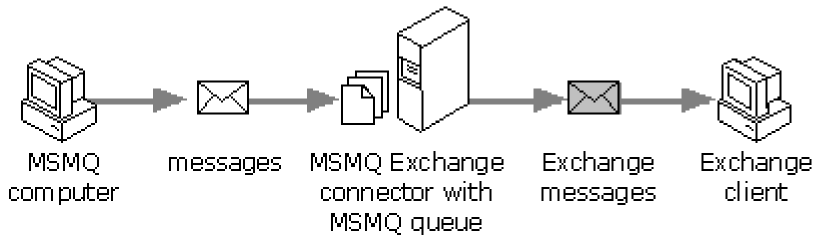


Figure B.1

MSMQ-based applications must create MSMQ messages in MSMQ Mail format (a subset of MIME). Application developers can create these applications using the MSMQ Mail SDK. For more information, see the MSMQ Mail SDK.

Sending from Exchange to MSMQ

Microsoft Exchange clients can send messages and forms to MSMQ queues by addressing messages or forms to the label of the destination queue (using an address type of MSMQ). This address can be predefined as a custom recipient in the Exchange Address Book.

Microsoft Exchange routes the appropriate message or form to the MSMQ Exchange connector. Next, the MSMQ Exchange connector translates the Exchange message or form into an MSMQ message and sends it to the appropriate destination queue. An MSMQ-based application can then read the message, parse the contents, and act upon the contents.

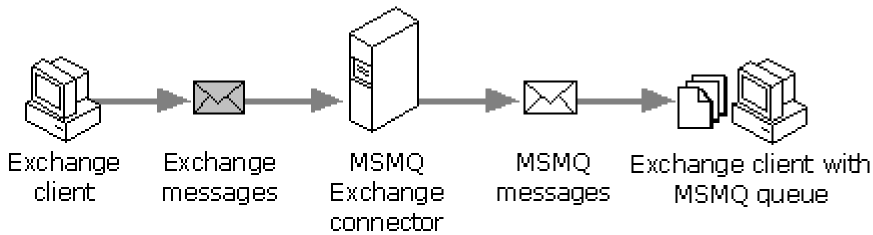


Figure B.2

Note The MSMQ queue must have a mail type ID of {5eadc0d0-7182-11cf-a8ff-0020afb8fb50}.

Because the MSMQ-based applications do not natively support the MAPI rich-text format Exchange uses, MSMQ address entries should be configured so they do not send messages in rich text.

Sending from Exchange to Exchange via MSMQ

Microsoft Exchange clients can send messages and forms to Exchange users in other MSMQ sites or enterprises via MSMQ queues. This is only possible if the destination site also has an MSMQ Exchange connector installed.

Microsoft Exchange Server routes the message or form to the MSMQ Exchange connector. The MSMQ Exchange connector then translates the Exchange message or form into an MSMQ message and sends it to the appropriate destination queue. An MSMQ Exchange connector in the destination site or enterprise then translates and forwards the message or form to the appropriate Exchange client.

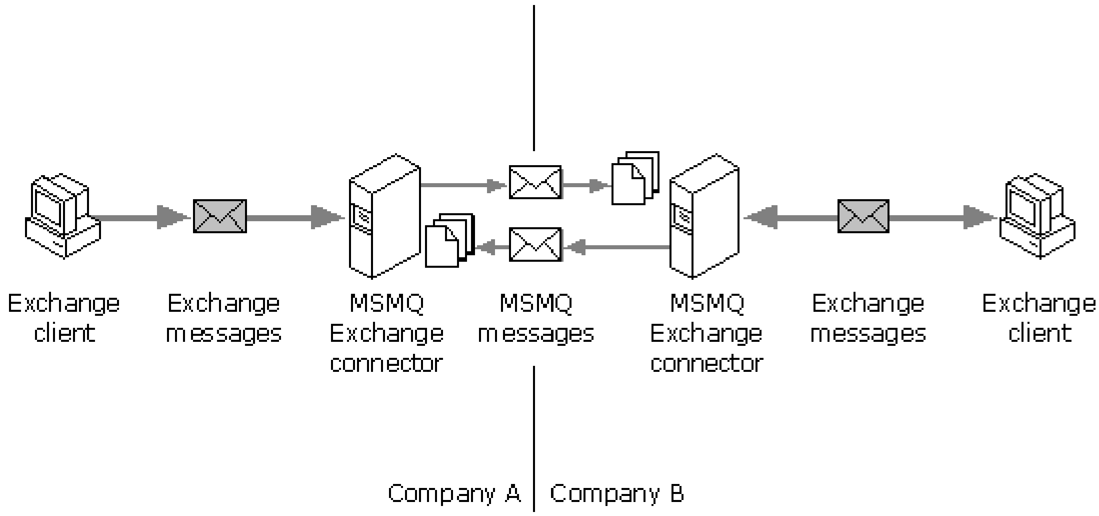


Figure B.3

As shown in Figure B.3, MSMQ and MSMQ Exchange connectors can be used to route messages between the two sites or enterprises. This configuration is useful when there is no physical network connection between Exchange sites or organizations. It is analogous to sending Exchange messages between two organizations via the Internet (using Exchange Internet connectors).

For the destination address, the sending Exchange client uses the queue label of the queue on the MSMQ Exchange connector in the destination site and the specific mail address of the destination Exchange client. The address format is *user mail alias@queue label* (for example, JoeUser@ExchangeQueue).

Messages sent from Exchange clients to other Exchange clients via MSMQ should have an address type of MSMQ. This notifies the Exchange server that the message should be routed through the MSMQ Exchange connector. Because both Exchange clients natively support the MAPI rich-text format, MSMQ address entries should be configured so they send messages in rich text.

Installing the MSMQ Exchange Connector

The following software is required to support an MSMQ Exchange Connector:

- Windows NT Server version 4.0 or later
- Microsoft Exchange Server version 4.0a (4.0 plus Exchange Service Pack 2) or later with the Exchange Administrator installed
- MSMQ independent client or server

MSMQ and the Exchange connector do not need to actually be installed on the Exchange server. You can install the connector on any MSMQ independent client or server (running on Windows NT Workstation or Windows NT Server), and configure the connector to communicate though the Exchange server. To install MSMQ Exchange connector in this configuration, you must first run Exchange Server Setup, click **Complete/Custom**, and select **Microsoft Exchange Administrator**.

Choosing a User Account To Run Setup

MSMQ Exchange Connector Setup must modify the Exchange site configuration so the site recognizes the connector. To modify the Exchange site configuration, Setup must run under a user account that has administrative permissions on the Exchange site. In other words, the user account you log on under before running Setup must have its Exchange site role set to **Permissions Admin**.

For information on setting Exchange site roles, see the Exchange Server documentation.

Running MSMQ Exchange Connector Setup

Determine the following before installing an MSMQ Exchange connector:

- The computer name of the Exchange Server the connector will use.
- The name of the Exchange site the MSMQ Exchange connector will support.
- The name of the queue from which the MSMQ Exchange connector will read incoming MSMQ messages. If the queue you specify during Setup does not exist, the MSMQ Exchange connector service creates the queue the first time it starts.
- The name and password of the user account under which the MSMQ Exchange connector service runs. The MSMQ Exchange connector service should run under a user account that has at least the appropriate permissions in the Exchange server's site to function as a connector. This user account must also have its Exchange site role set to **View Only Admin**. This need not be the account under which you are running Setup.
- The path to the folder which will store the MSMQ Exchange connector files. If the folder you specify during Setup does not exist, Setup creates it.

For more information on Exchange site roles, see the Exchange Server documentation.

To install an MSMQ Exchange connector

1. Run Setup from the \MSMQ\MQMail\exchconn\setup folder on the Windows NT Server, Enterprise Edition 4.0 Components CD.
2. Click **Continue**.
3. Click **Install**.
4. In **Choose Server**, type the computer name of the Exchange Server the connector will use, and then click **Continue**.

   By default, the name of the computer you are using is provided.
5. In **Choose Site**, click the name of the Exchange site the MSMQ Exchange connector will support, and then click **Continue**.
6. In the **Install** dialog box, provide the name of the **Inbound Queue**, **Service Account**, **Service Password**, and **Service Directory**.

In **Inbound Queue**, type the name of the queue from which the MSMQ Exchange connector will read incoming MSMQ messages. If the queue does not exist, Setup creates it.

In **Service Account**, type the name of the user account, in the domain\username format, under which the MSMQ Exchange connector service will log on.

In **Service Password**, type the domain password for the user account specified in Service Account.

In **Service Directory**, type the path to the folder which will store the MSMQ Exchange connector files. For example, **C:\exchq**. If the folder does not exist, Setup creates it.

7. Click **Continue**.

Managing an MSMQ Exchange Connector

You configure the MSMQ Exchange connector using the **MSMQ Connector Properties** dialog box in Exchange Administrator. You can use the **General** tab to specify a maximum message size and make an **Administrative note**. You can use the **Address Space** tab to configure the address space and specify the connection cost (this affects Exchange routing). You can use the **Options** tab to:

- Enable or disable the processing of inbound MSMQ messages.
- Specify the maximum number of threads that can be used to process inbound MSMQ messages.
- Enable or disable the processing of outbound MSMQ messages.
- Specify the maximum number of threads that can be used to process outbound MSMQ messages.

By default, four threads are used for inbound processing, and four threads are used for outbound processing. If the processing load is greater in one direction, or the server is not processing messages fast enough, increase the number of threads as needed.

Use the following procedure to view the MSMQ Exchange connector properties:

To configure the MSMQ Exchange connector

1. Run the Exchange Administrator and connect to the MSMQ Exchange connector server.
2. Click the plus sign next to the site name.
3. Click the plus sign next to **Configuration**.
4. Click **Connections**.
5. Double-click **MSMQ Connector (*queue label*)**, where *queue-label* is the name of the MSMQ queue created for receiving inbound MSMQ messages.
6. configure the MSMQ Exchange connector properties using the **MSMQ Connector Properties** dialog box.

Configuring the MSMQ Exchange Connector Service Using Control Panel

The MSMQ Exchange connector runs as a service under Windows NT. The service is called **MSMQ Exchange Connector (*queue label)***, where *queue label* is the name of the MSMQ queue created for receiving inbound MSMQ messages.

Using the **Services** option in Control Panel you can:

- Stop, start, pause, and resume the service.
- Specify whether the service starts automatically or manually.

Monitoring an MSMQ Exchange Connector

The MSMQ Exchange connector writes informational, warning, and error information to the Windows NT application log, which can be viewed from Windows NT Event Viewer.

When an MSMQ Exchange connector cannot parse or process a message, the connector puts the message in a folder called Error. This folder is located in the connector's installation folder. You can either delete the message or attempt to correct it. You can then send the message to the appropriate destination.

Removing the MSMQ Exchange Connector

You can remove an MSMQ Exchange connector by running Setup.

To remove an MSMQ Exchange connector

1. Run Setup from the \MSMQ\MQMail\exchconn\setup folder on the Windows NT Server, Enterprise Edition 4.0 Components CD.
2. Click **Remove**.
3. Click the connector you want to remove and then click **Continue**.

Click on a letter to scroll to that section of the glossary and then click on the term you would like defined.

A

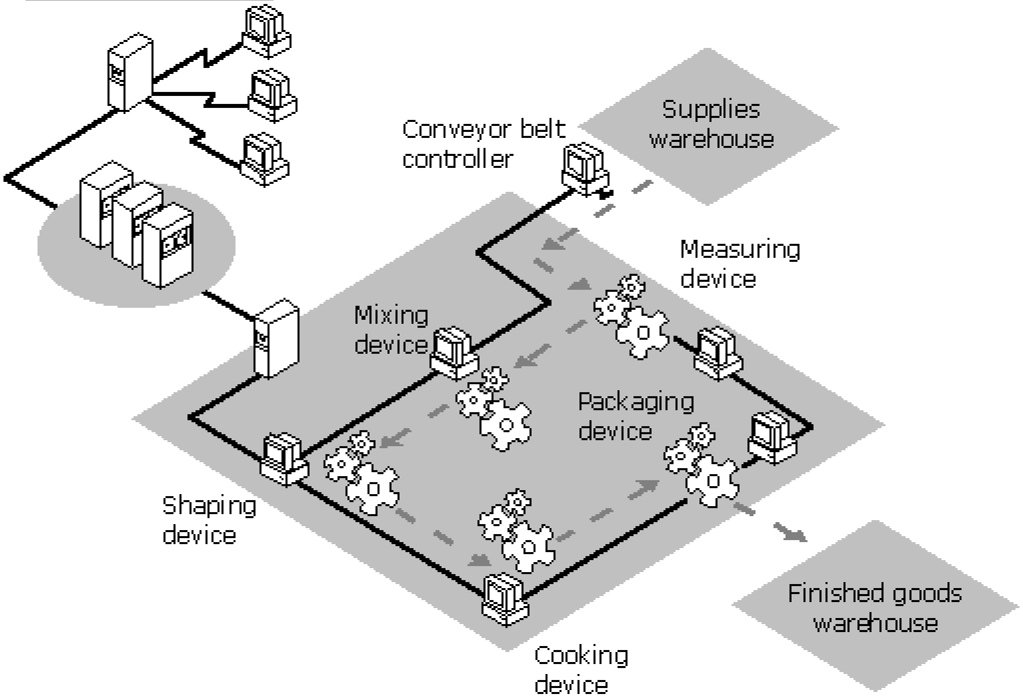
- ActiveX controls
- administration queue
- API
- authentication

B

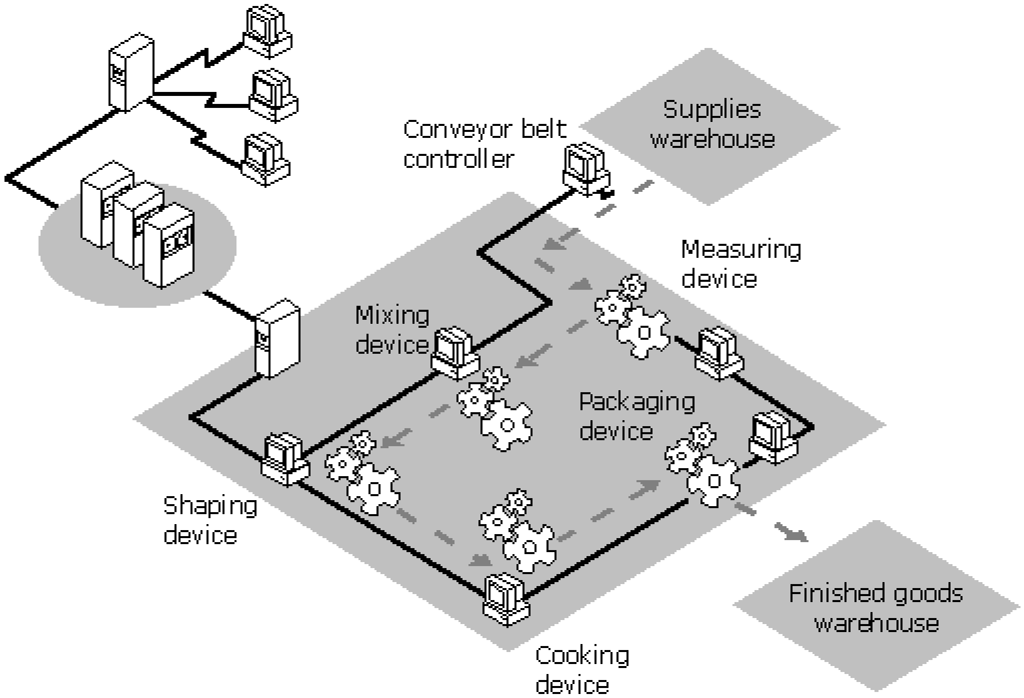
- Backup Site Controller
- base priority
- BSC

C

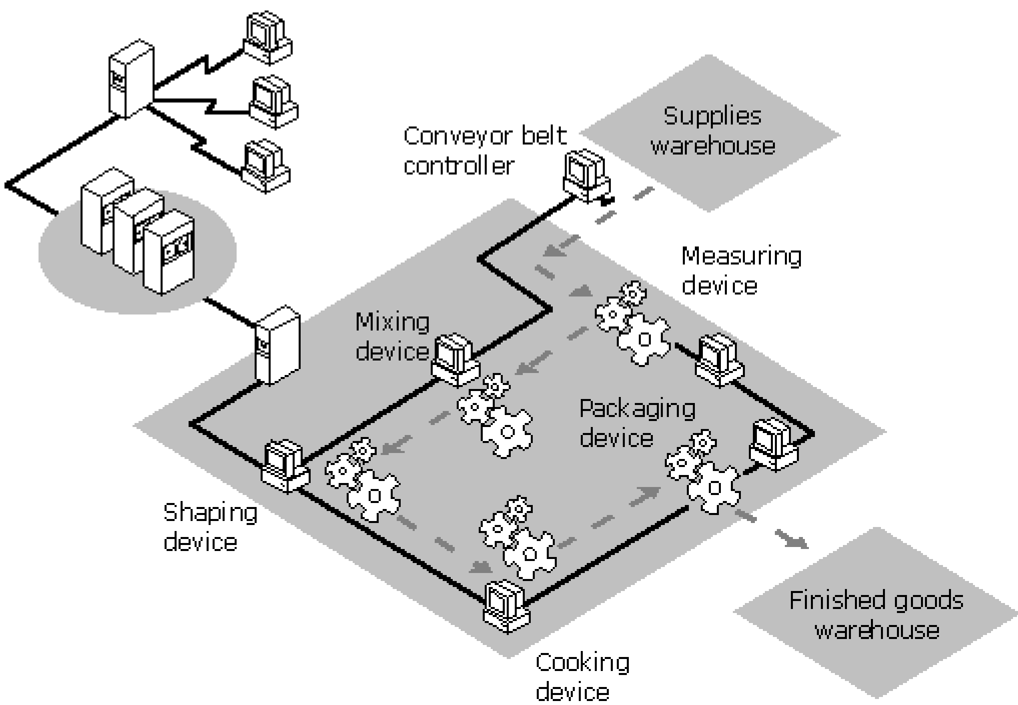
- connector queue
- controller server



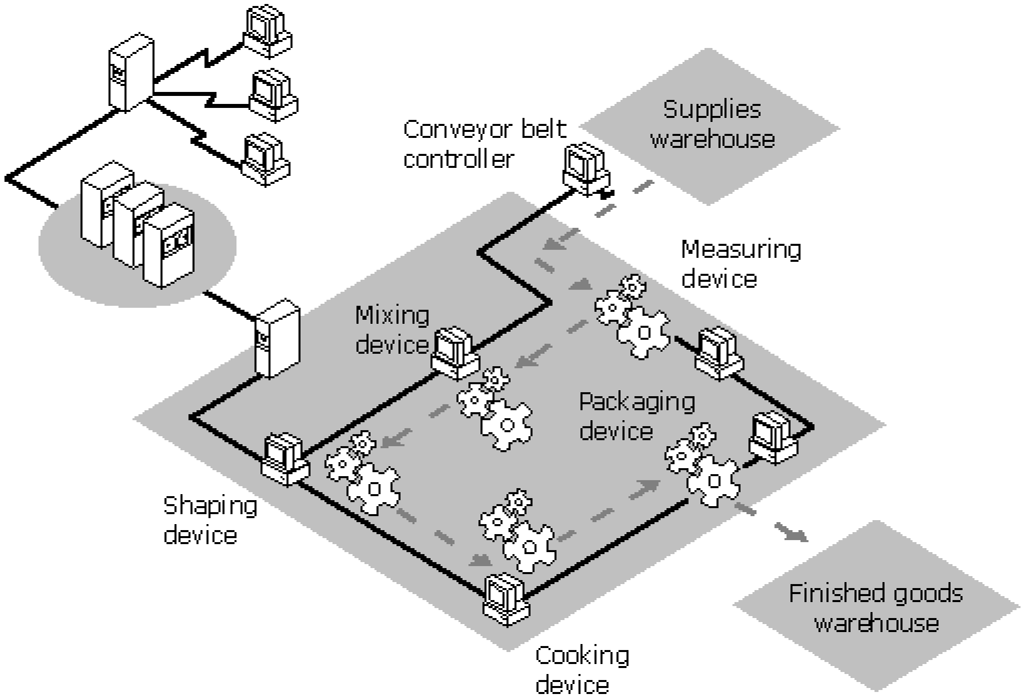
computer pathname



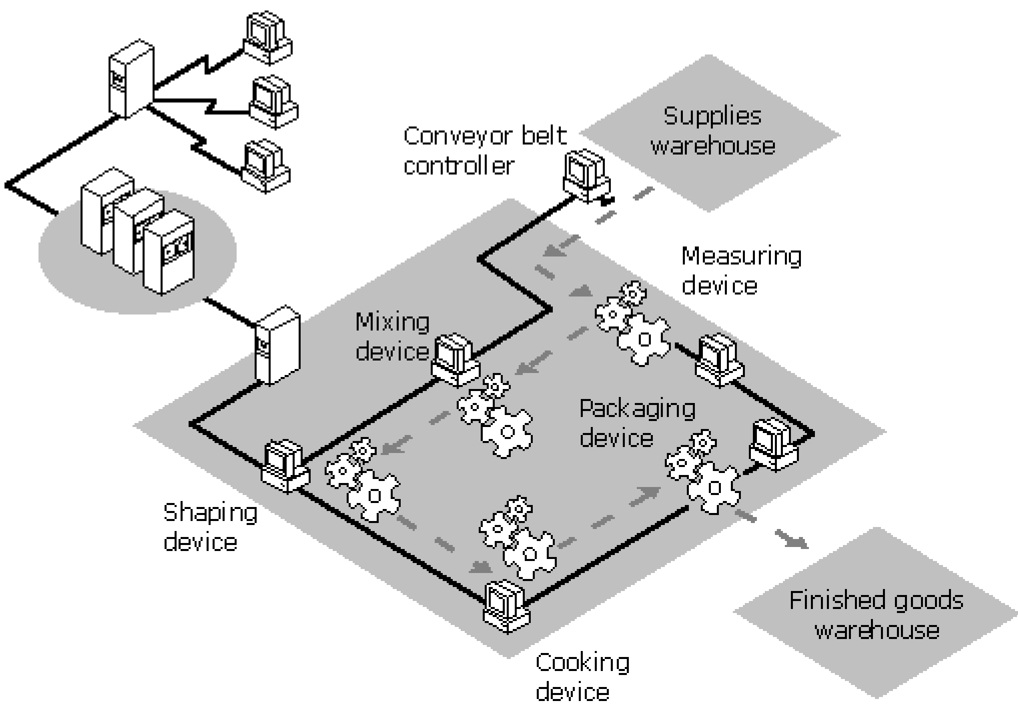
computer quota



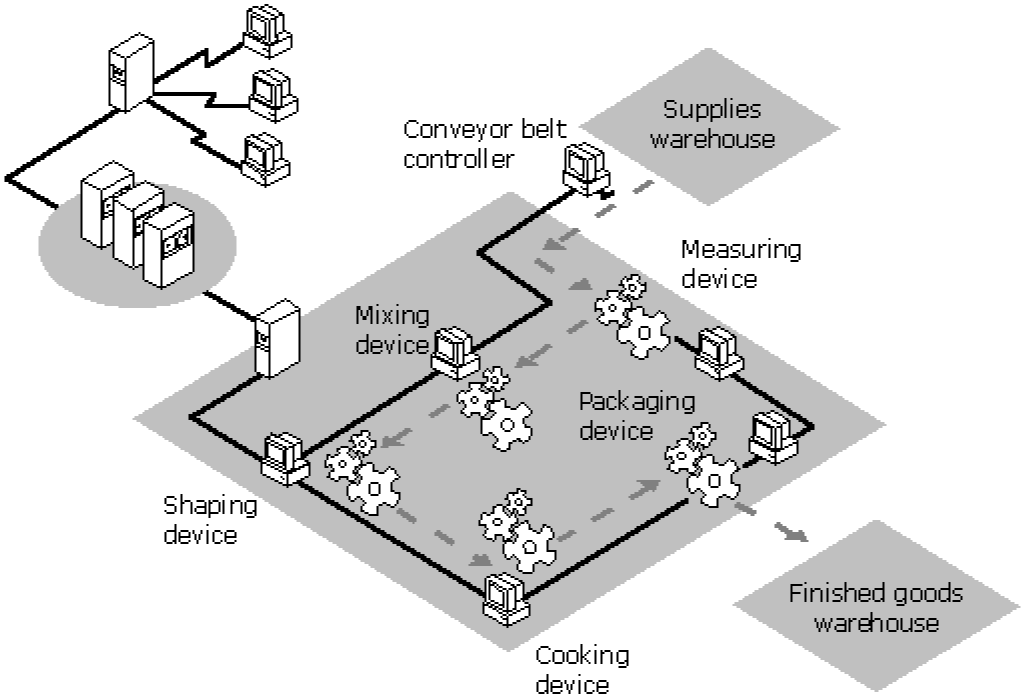
computer ID



CN

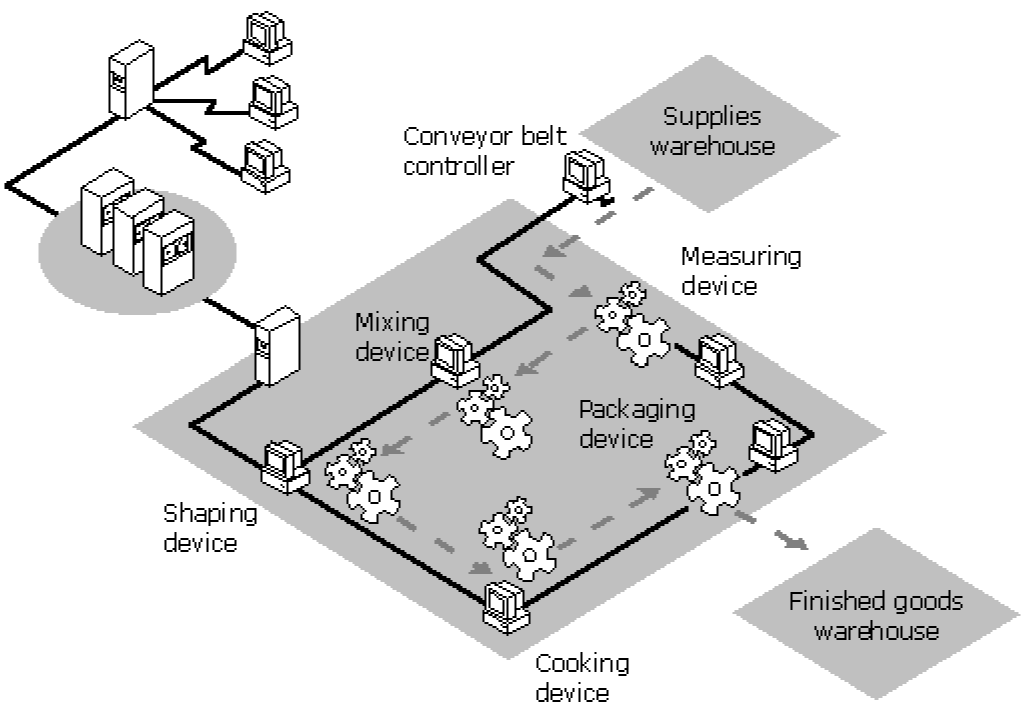


connected network

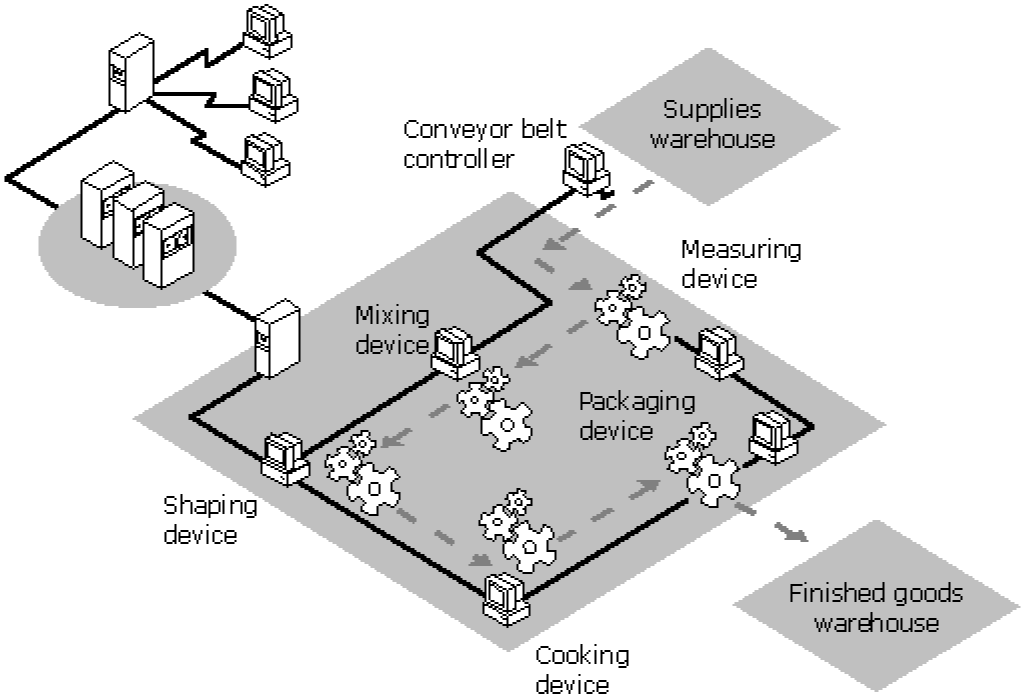


CSP

D



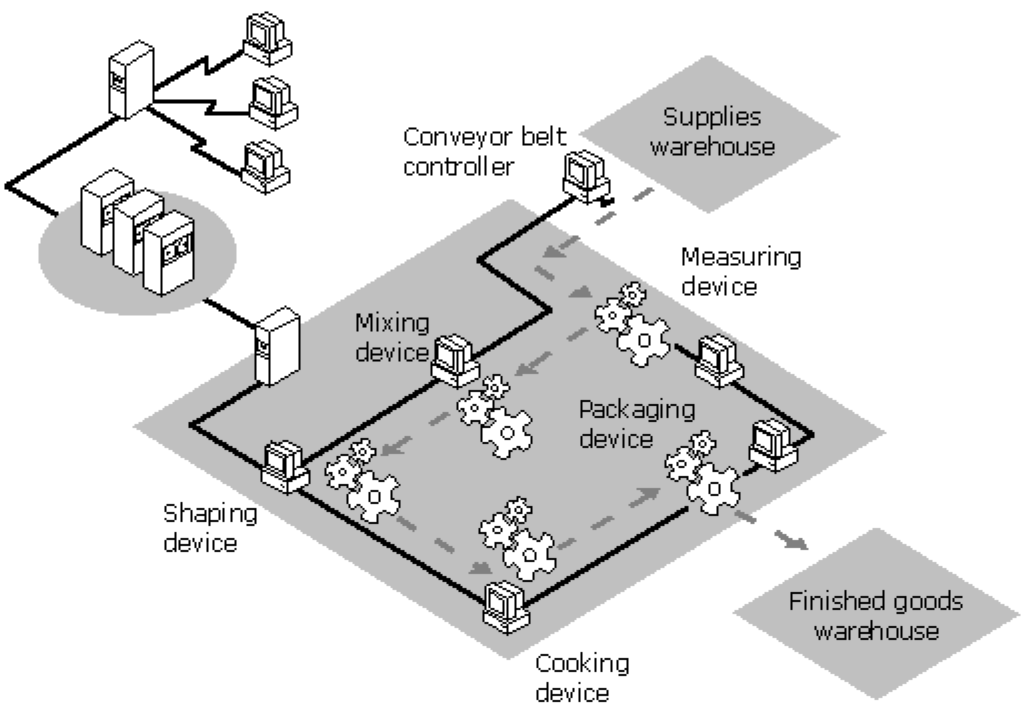
dead letter queue



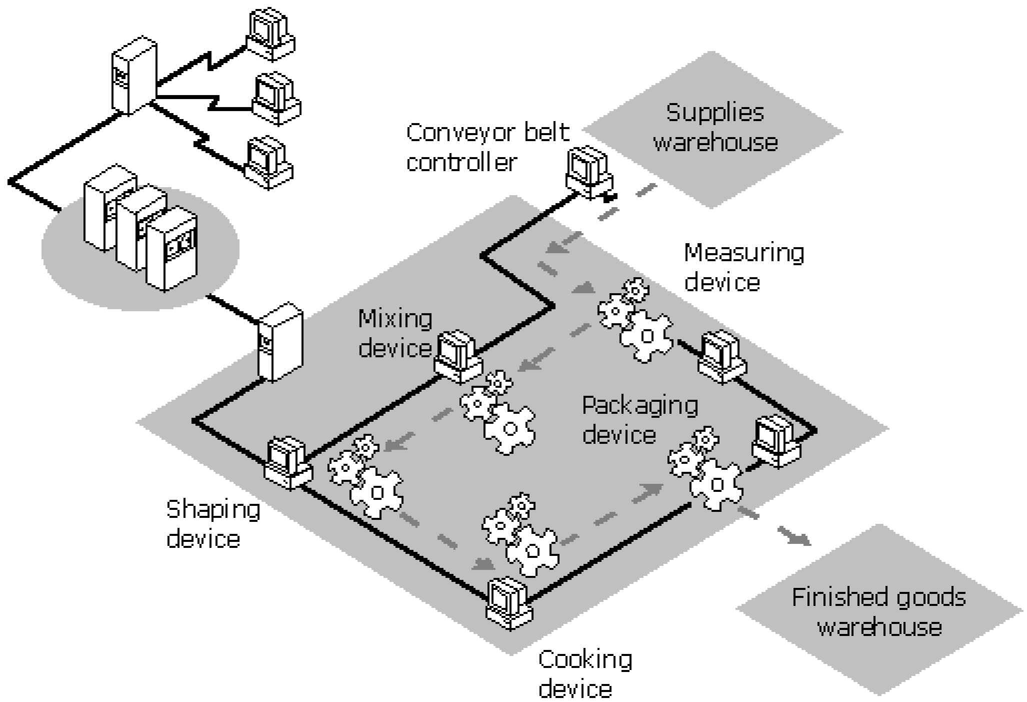
destination

queue

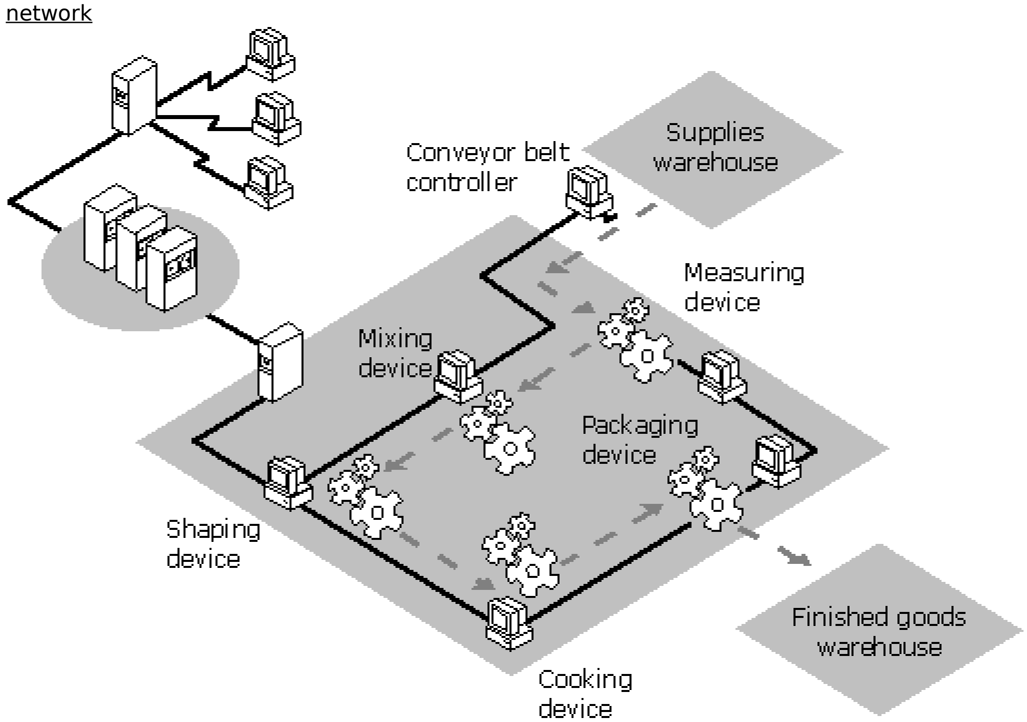
E



encryption

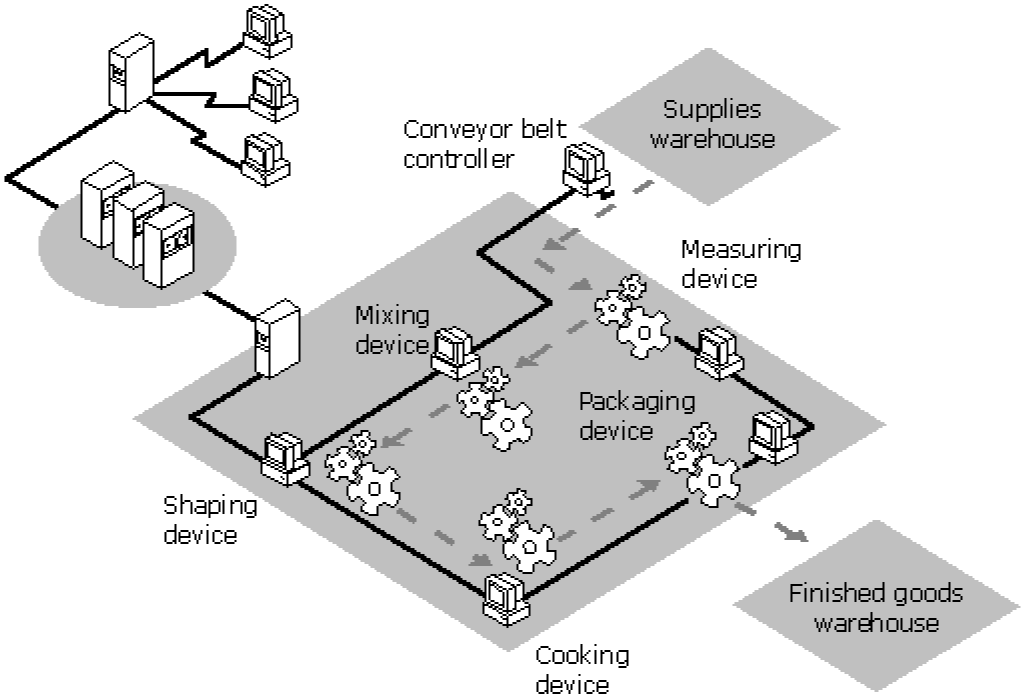


enterprise
network

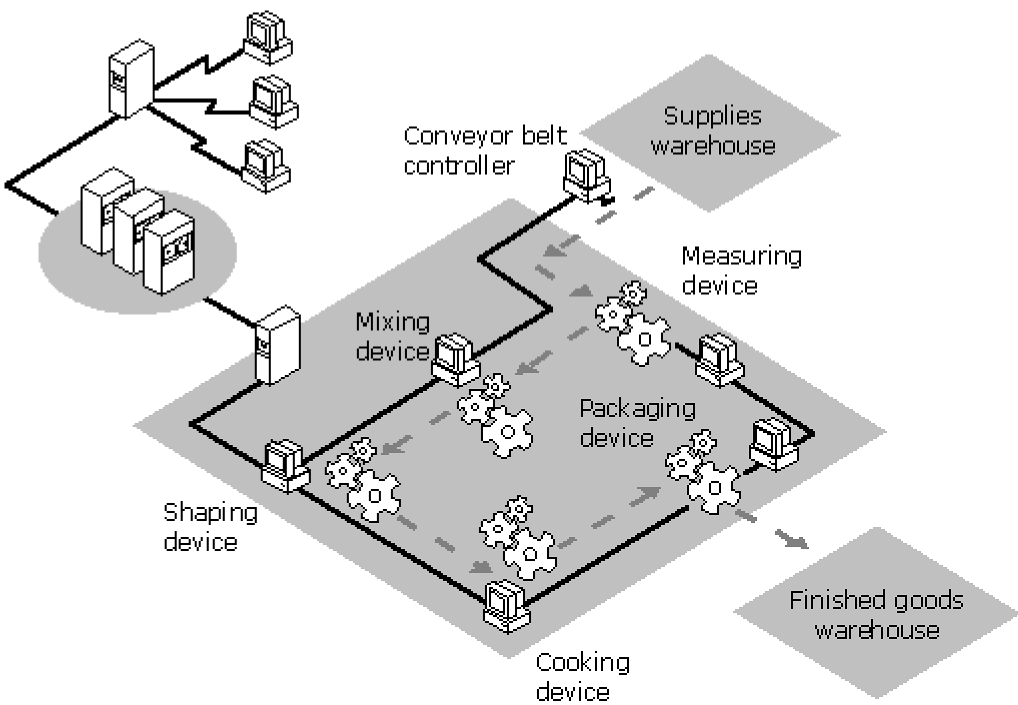


express message

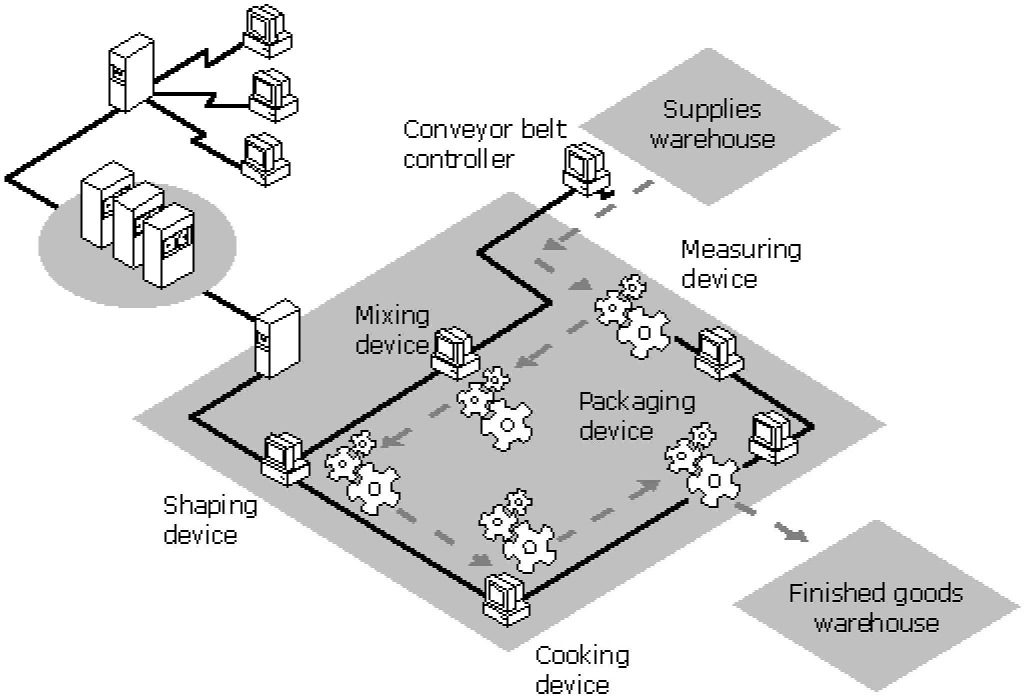
F



foreign computer

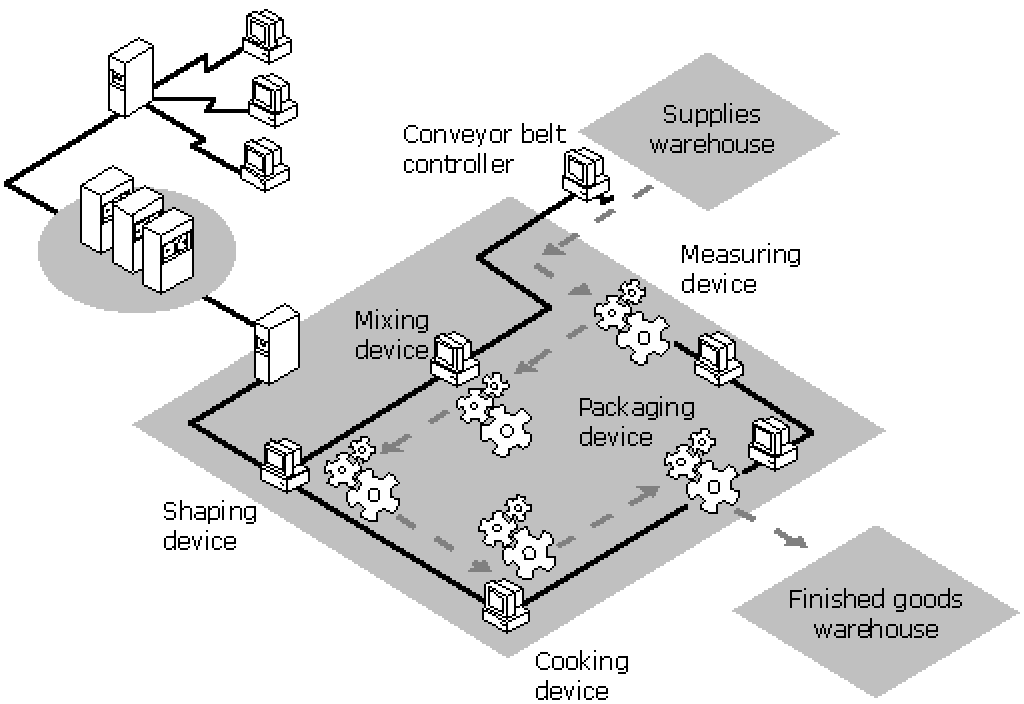


foreign CN



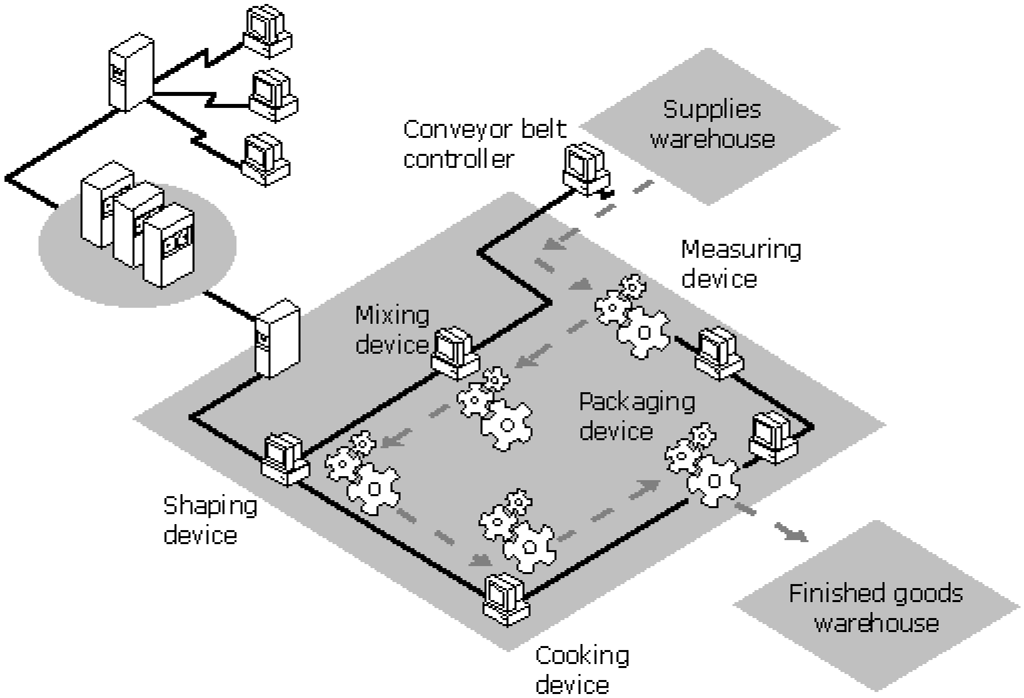
format name

G

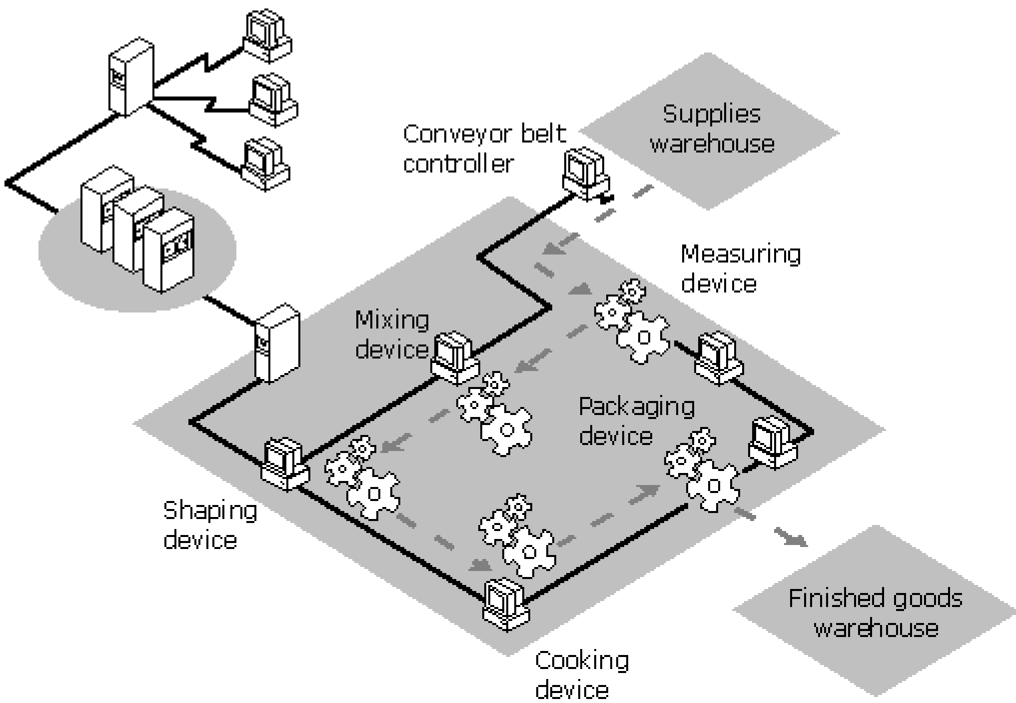


GUID

H

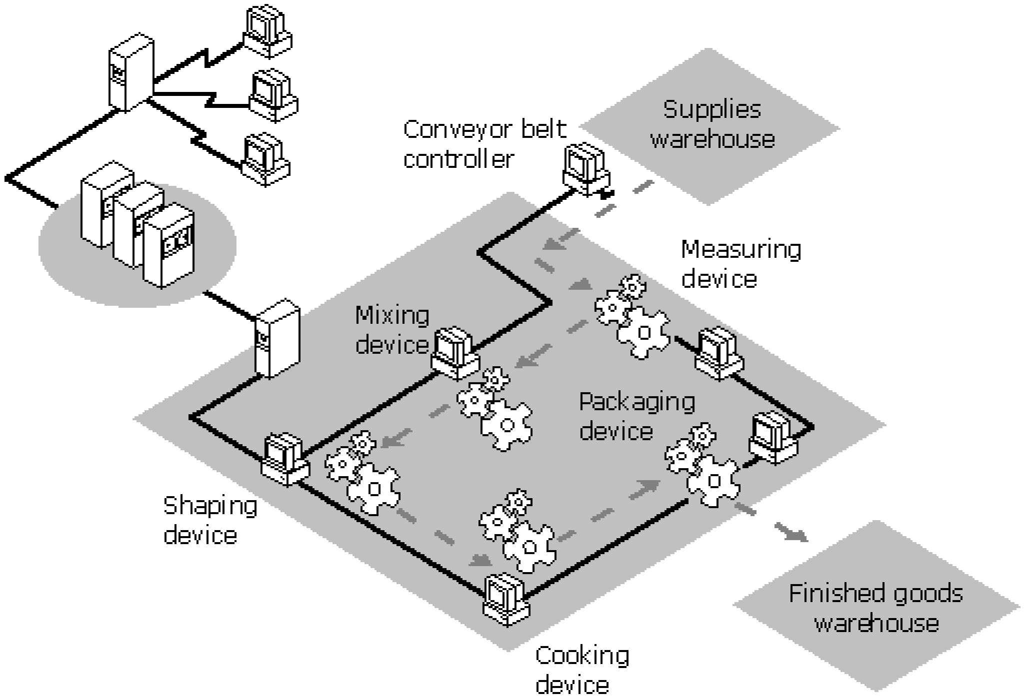


hash algorithm

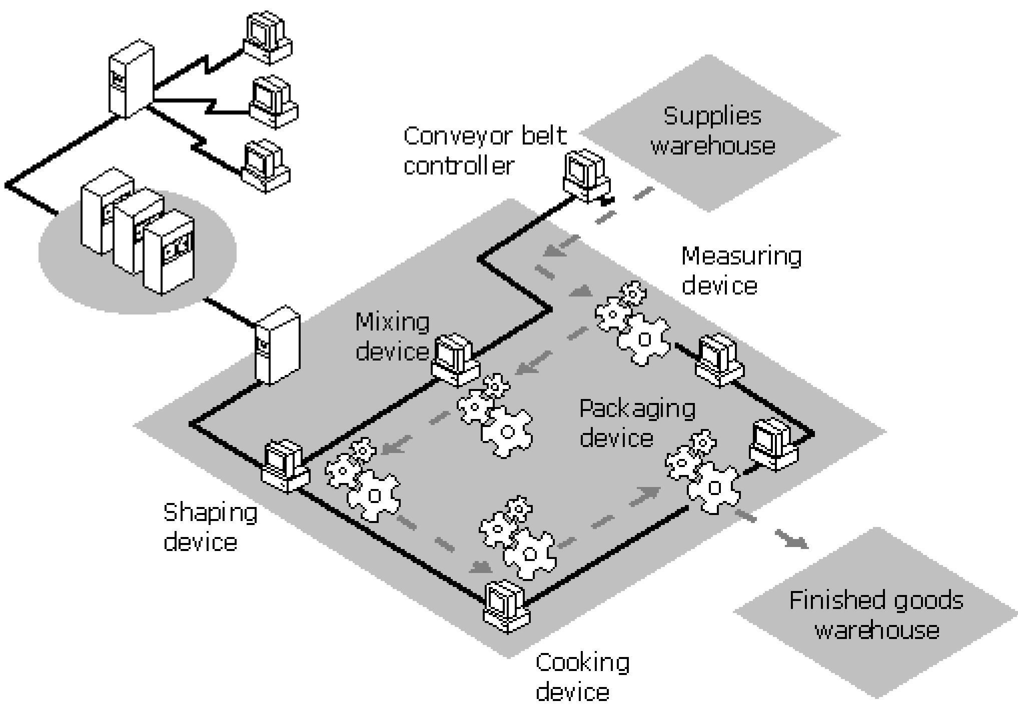


hop

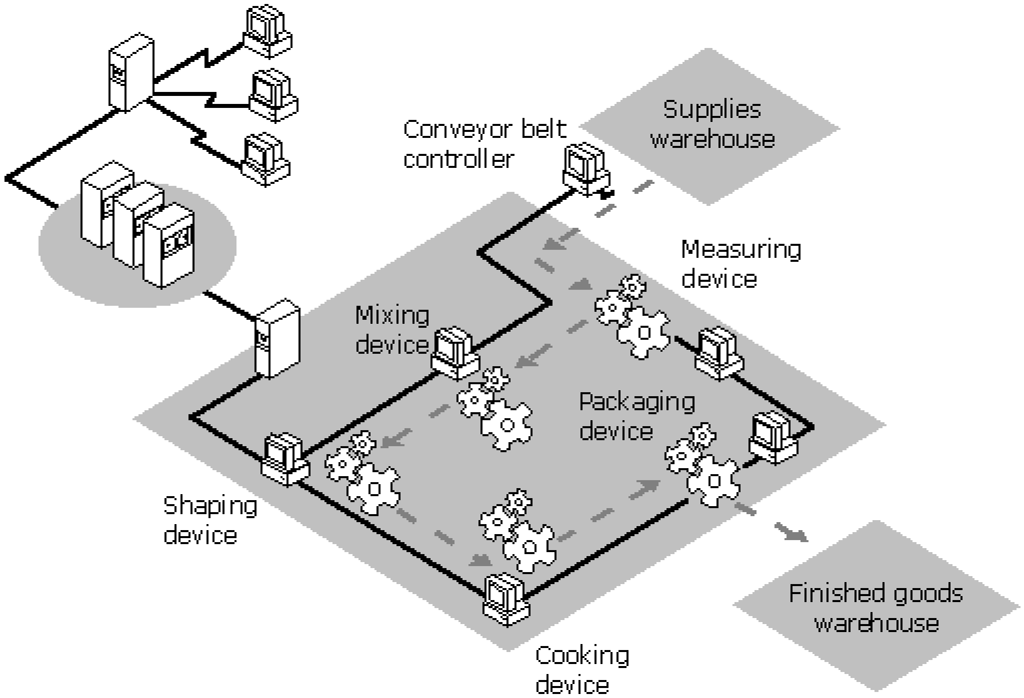
I



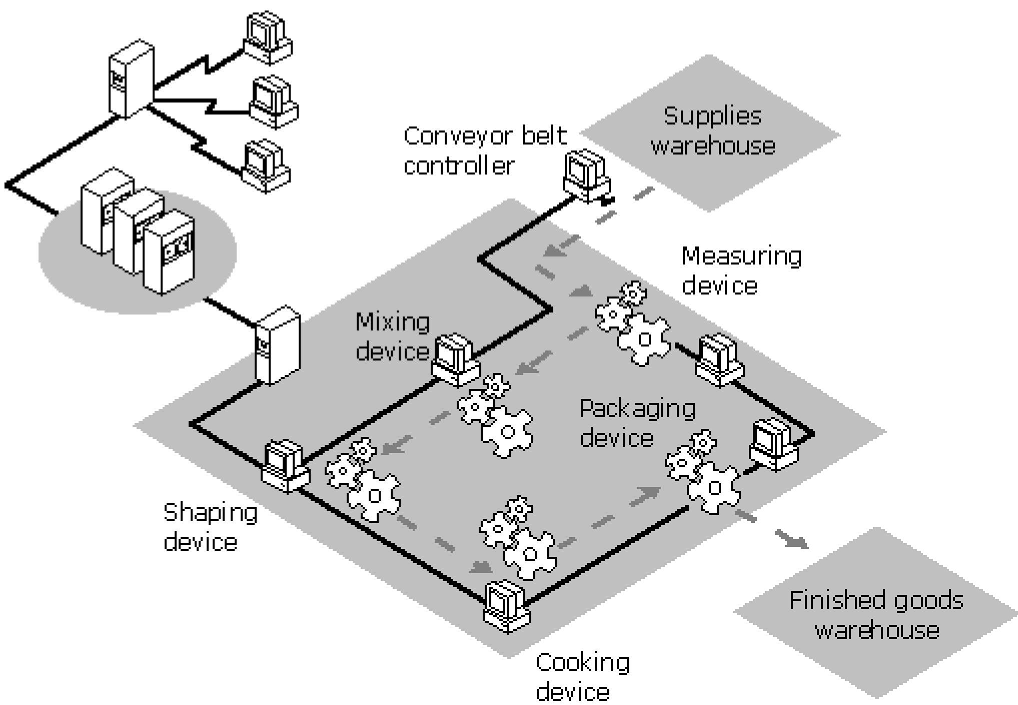
in routing server



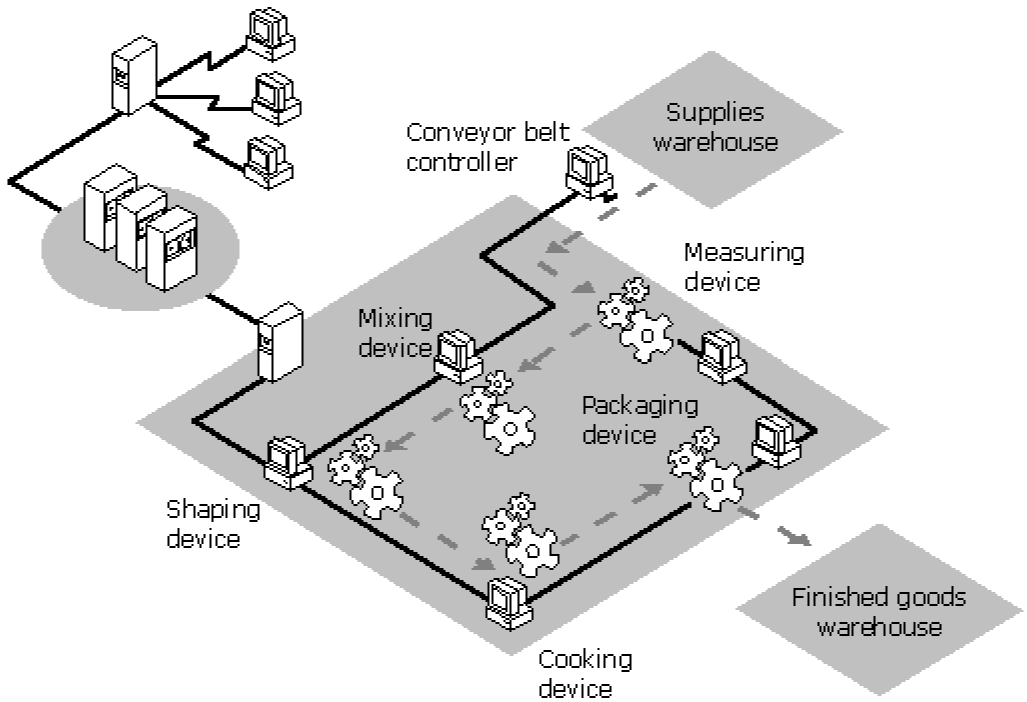
information server



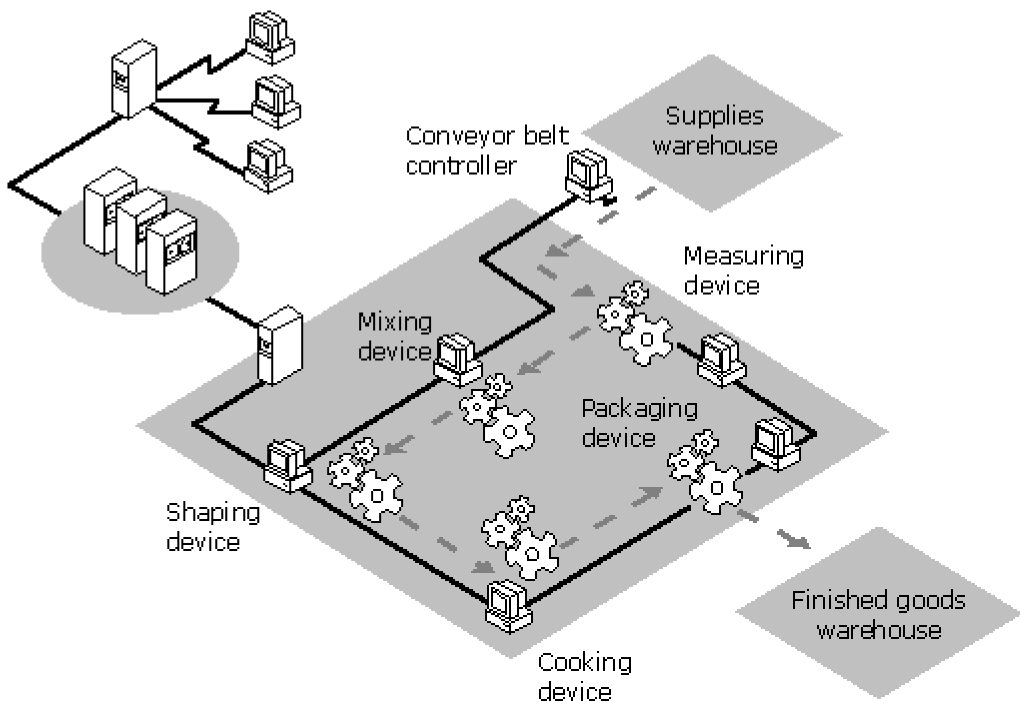
information store



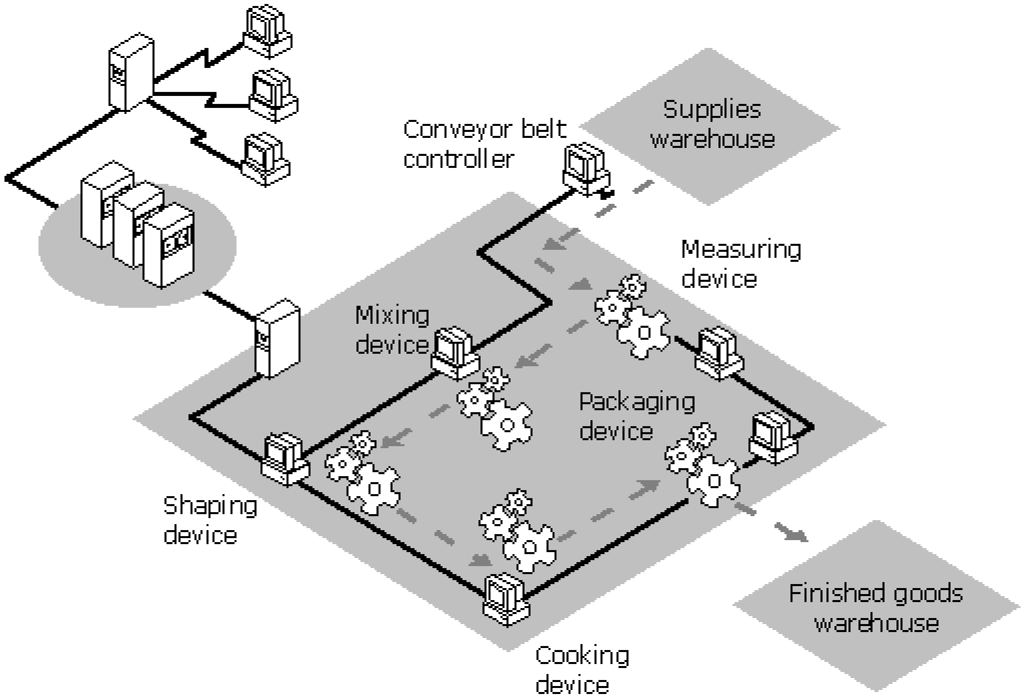
InRS



inter-site routing

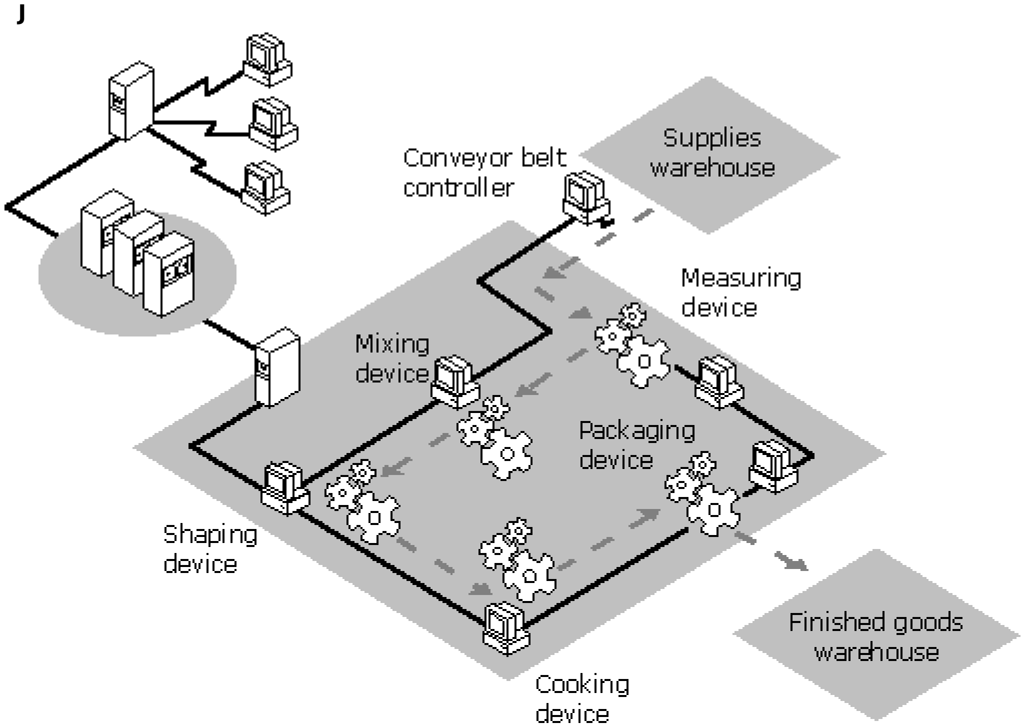


intra-site routing

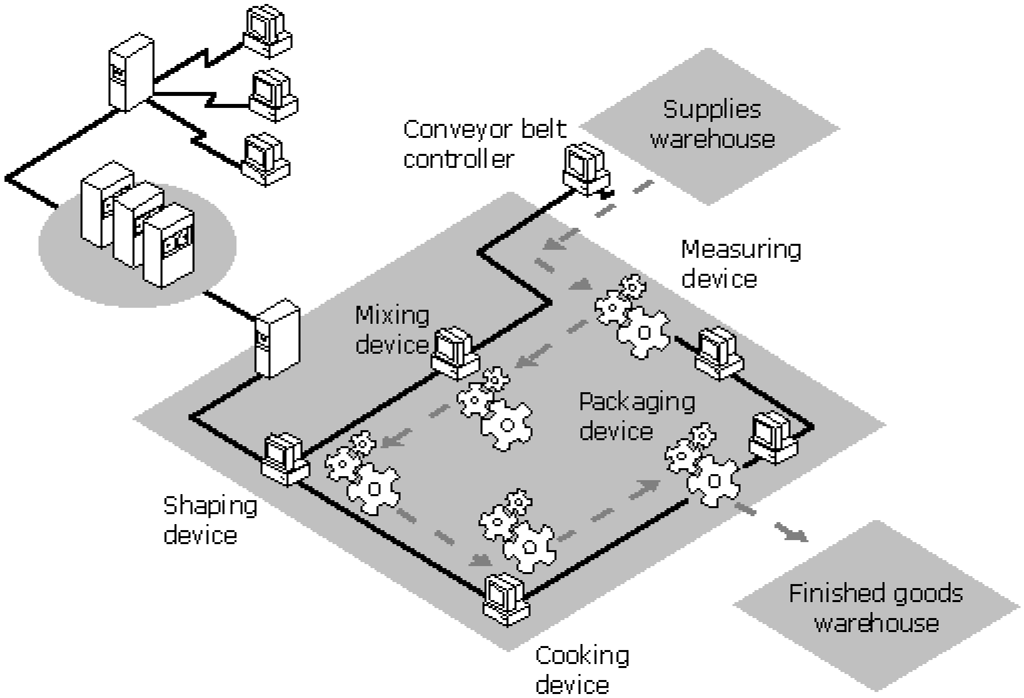


ISV

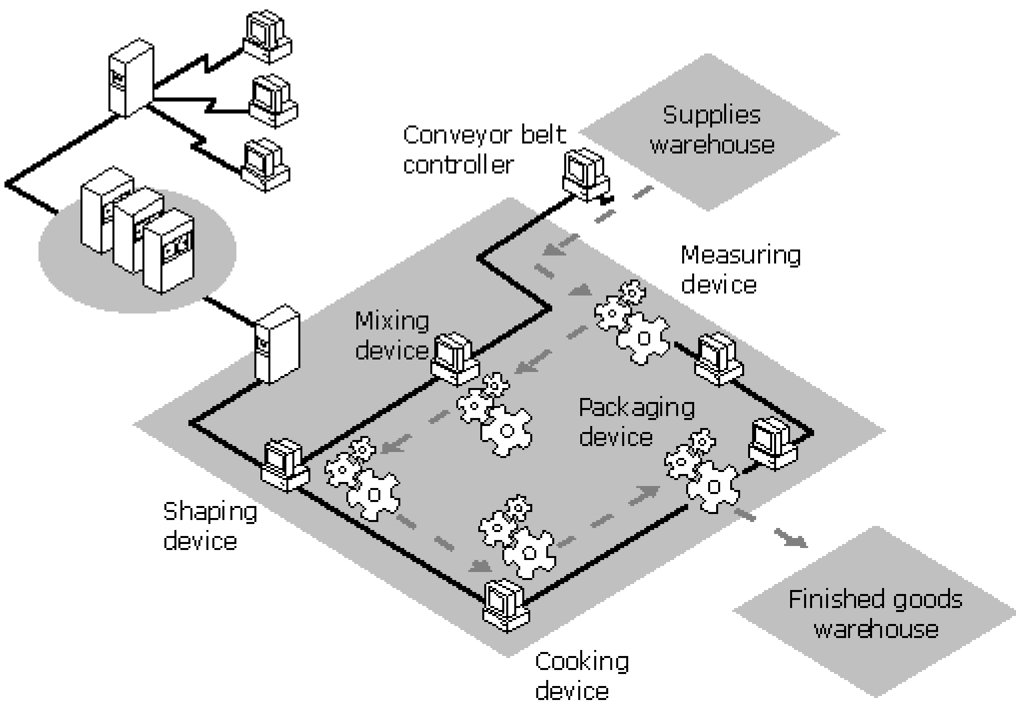
J



journaling



journal queue



journal quota

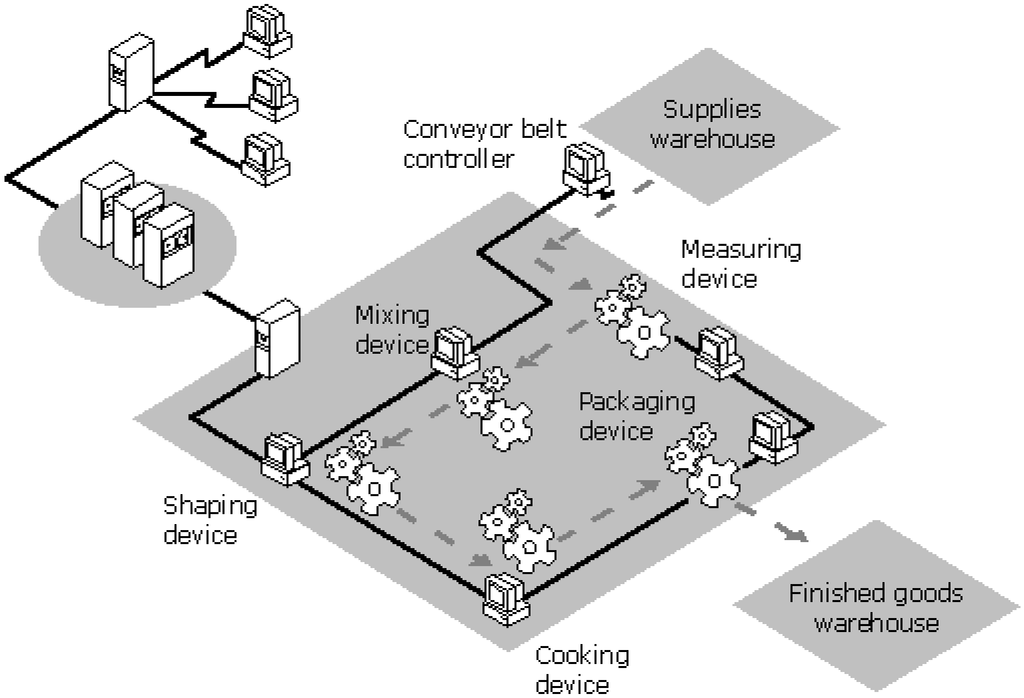
K

There are no glossary entries for this section.

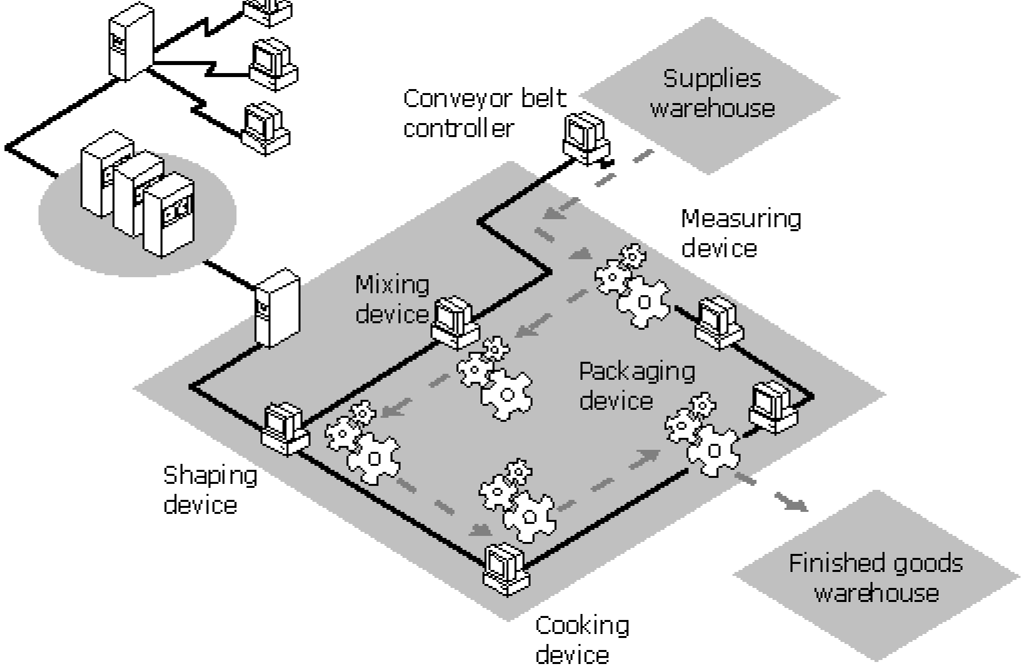
L

There are no glossary entries for this section.

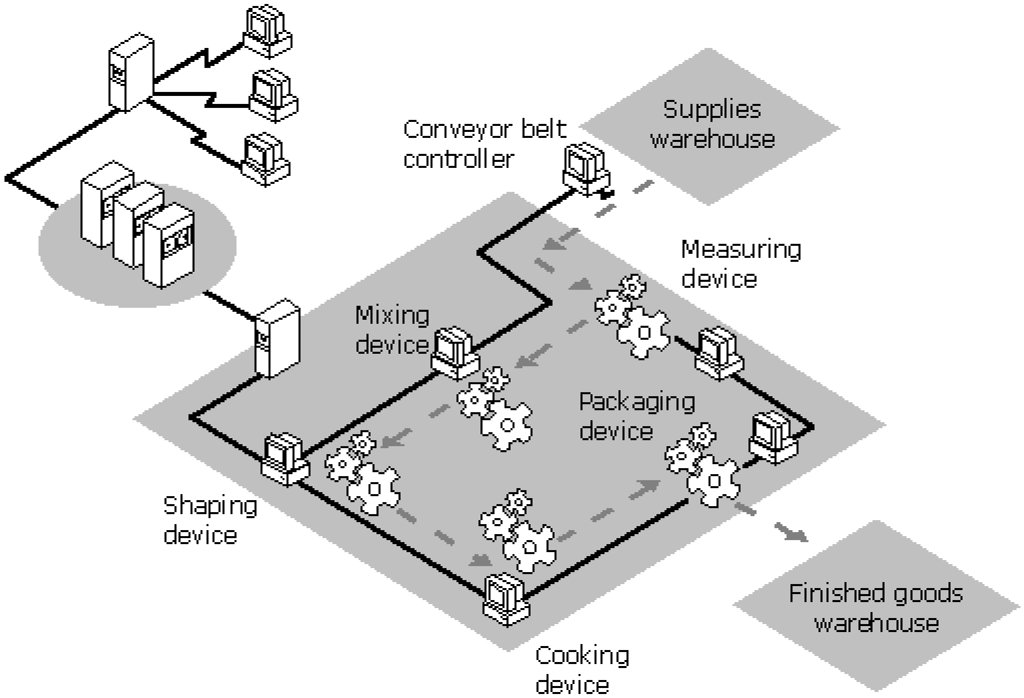
M



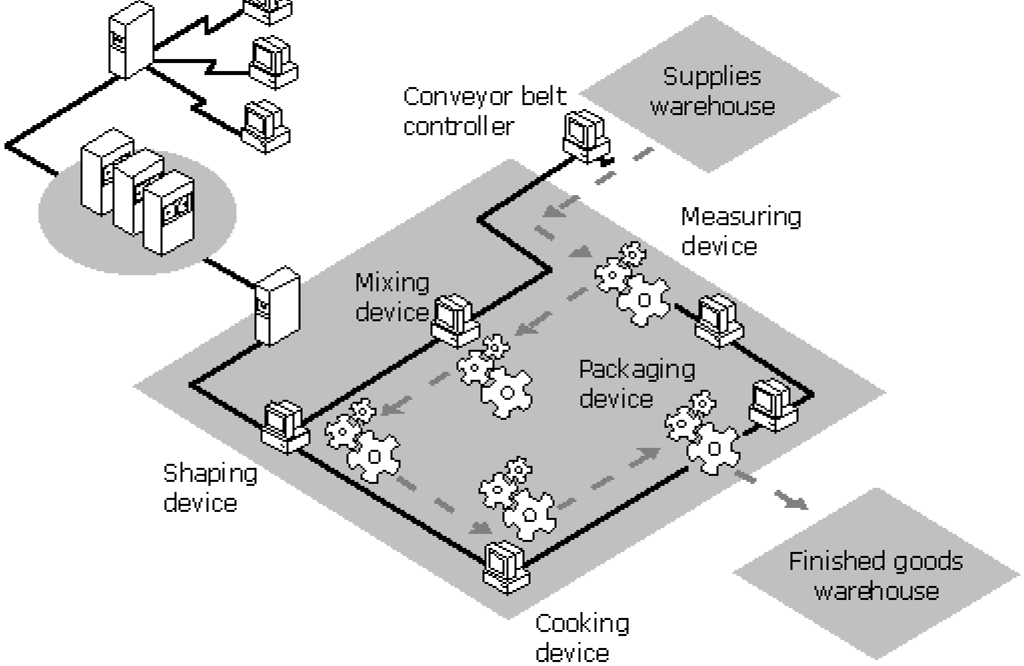
message



message class

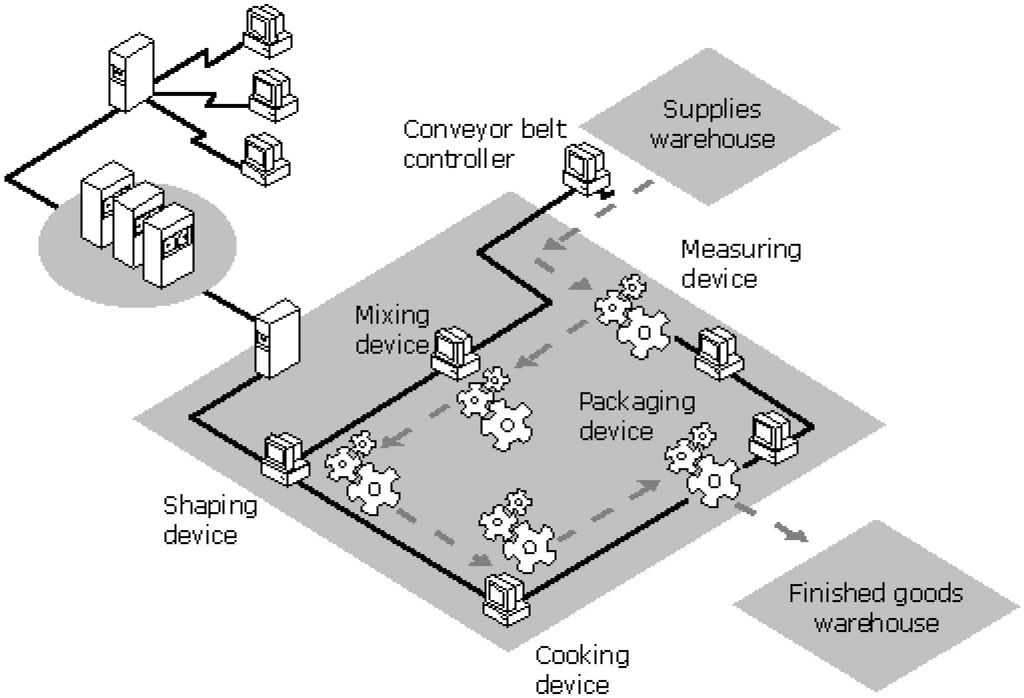


message label

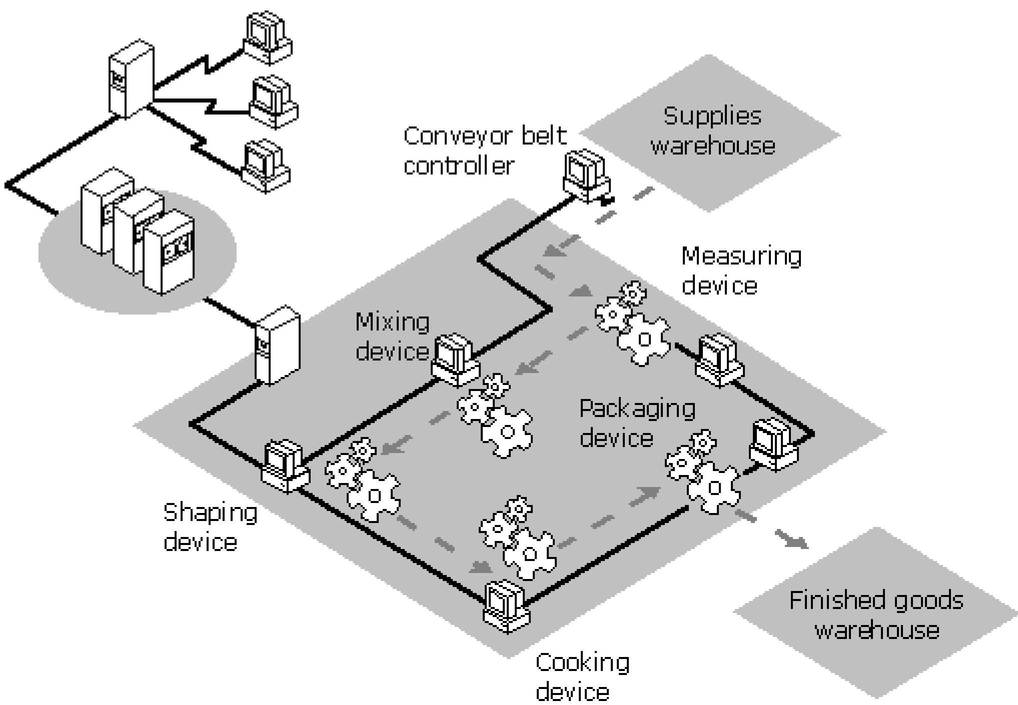


Microsoft

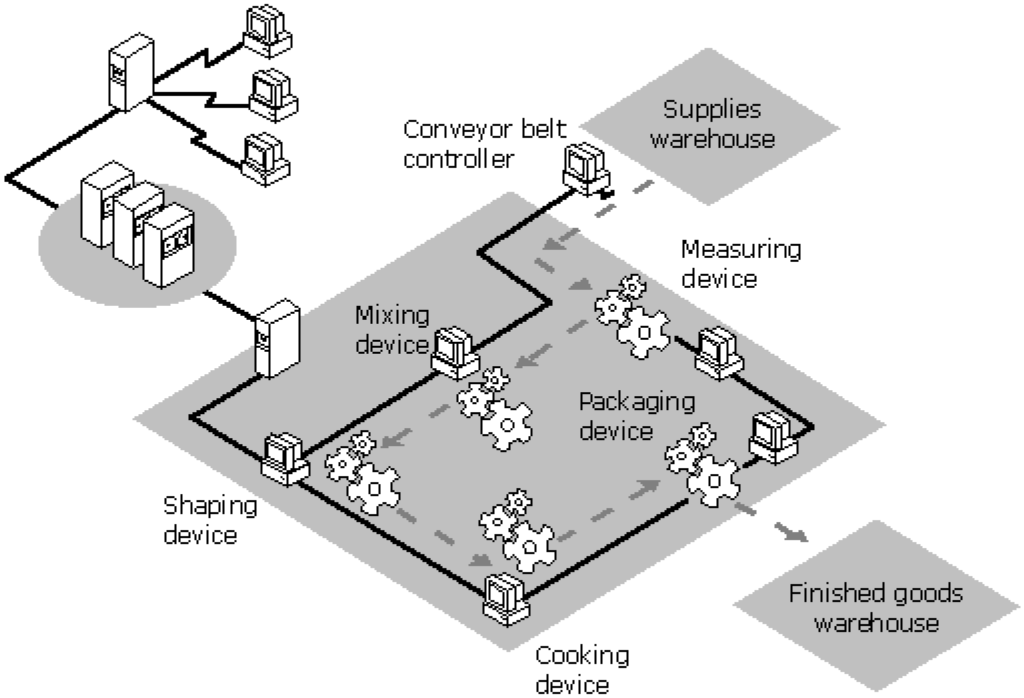
Message Queue Service


Supplies
warehouse
Conveyor belt
controller
Measuring
device
Mixing
device
Packaging
device
Shaping
device
Finished goods
warehouse
Cooking
device

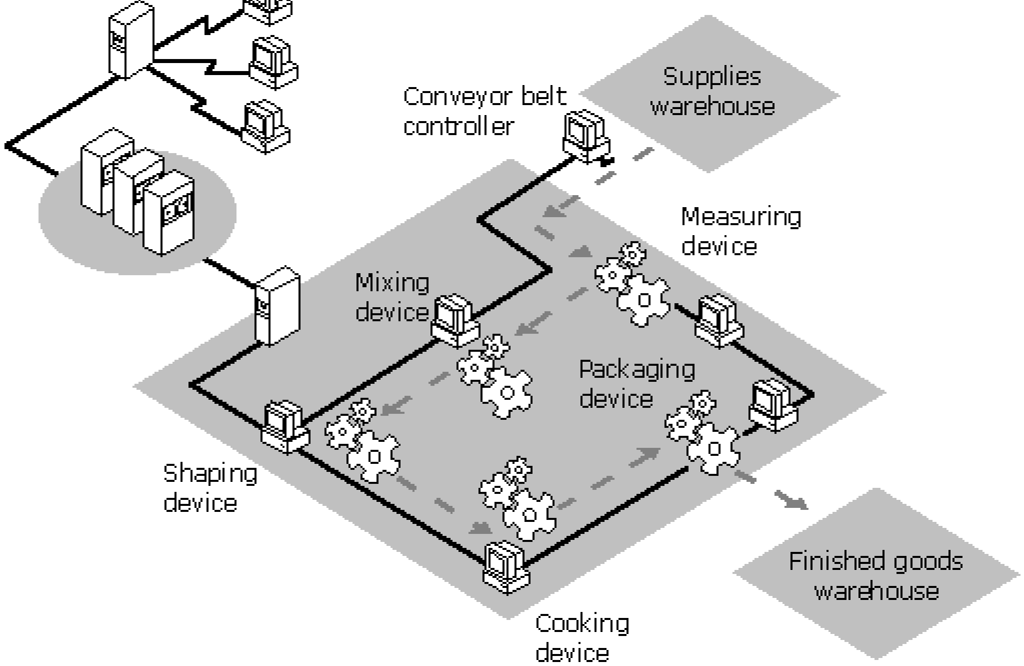
MQIS


Supplies
warehouse
Conveyor belt
controller
Measuring
device
Mixing
device
Packaging
device
Shaping
device
Finished goods
warehouse
Cooking
device

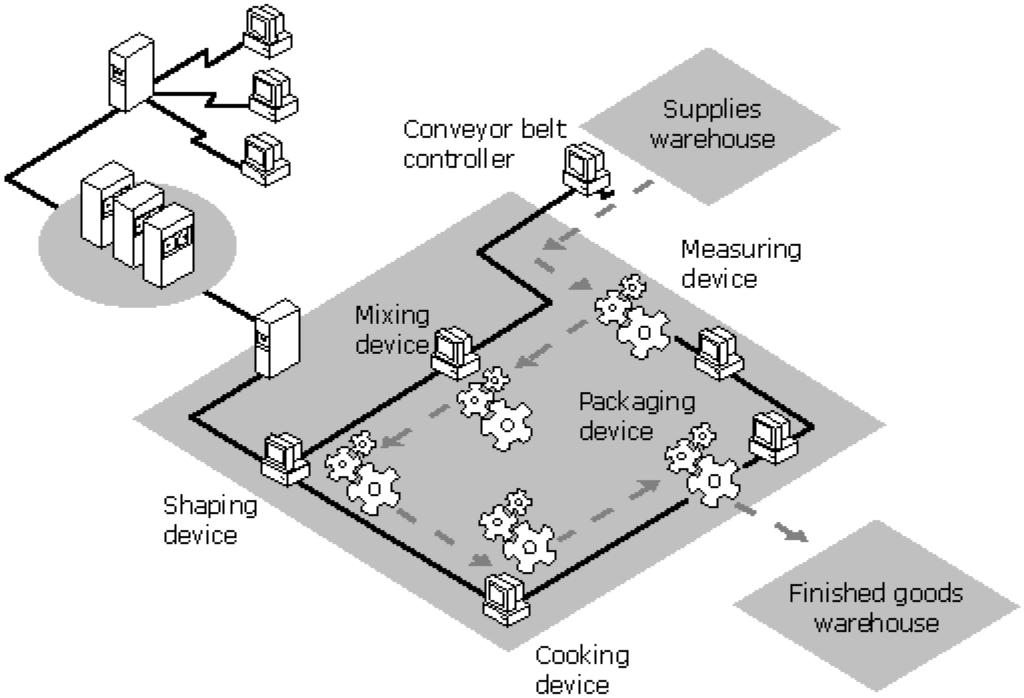
MS DTC



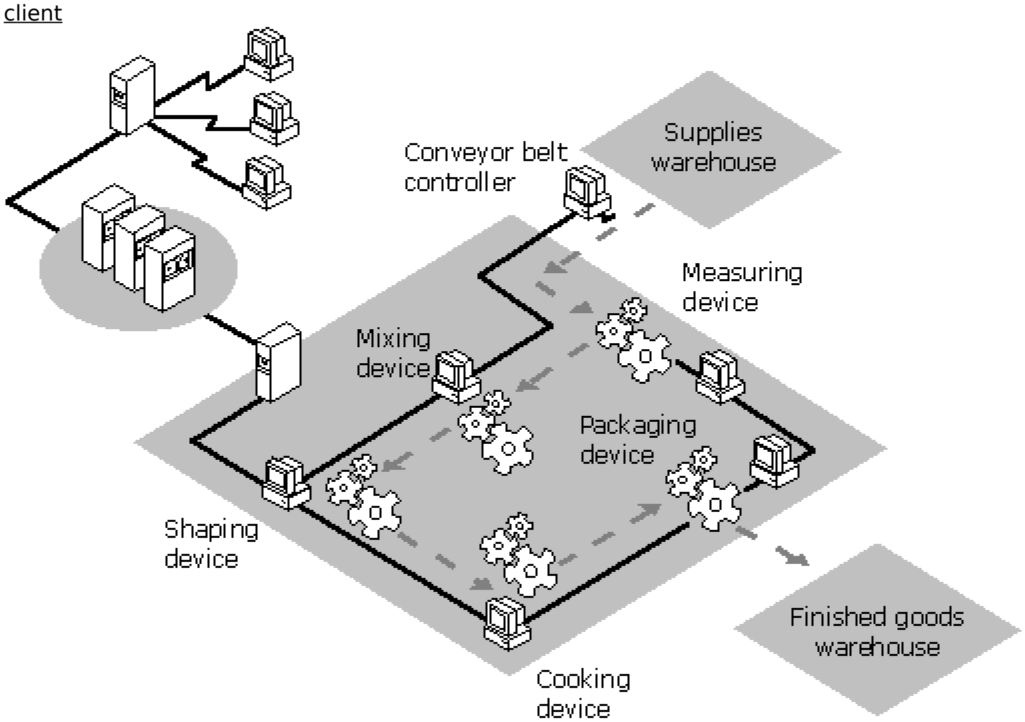
MSMQ



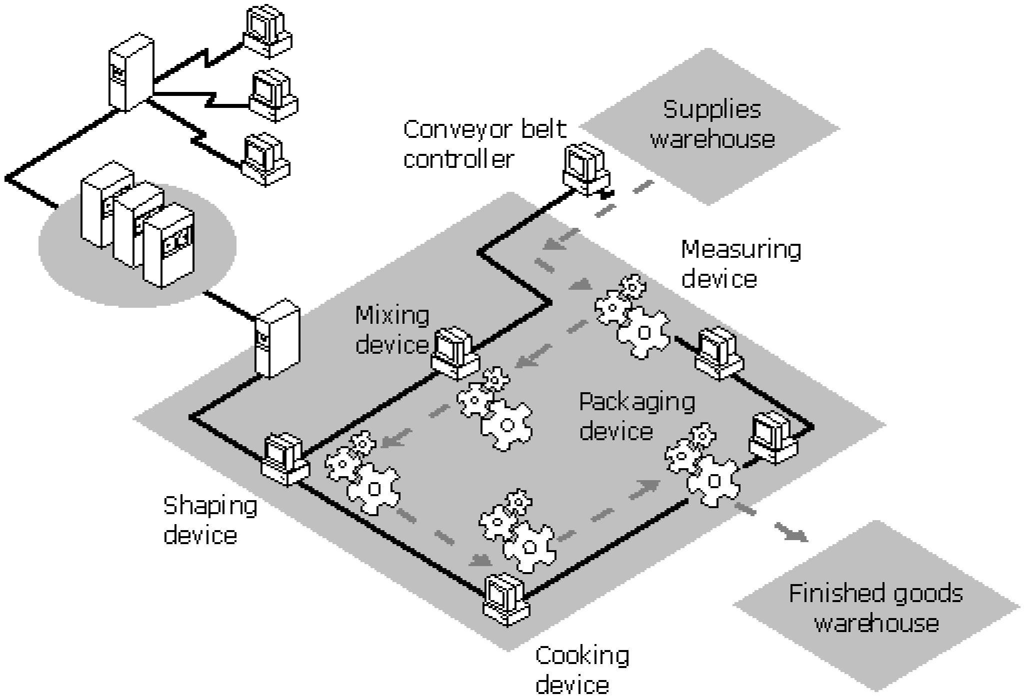
MSMQ connector server



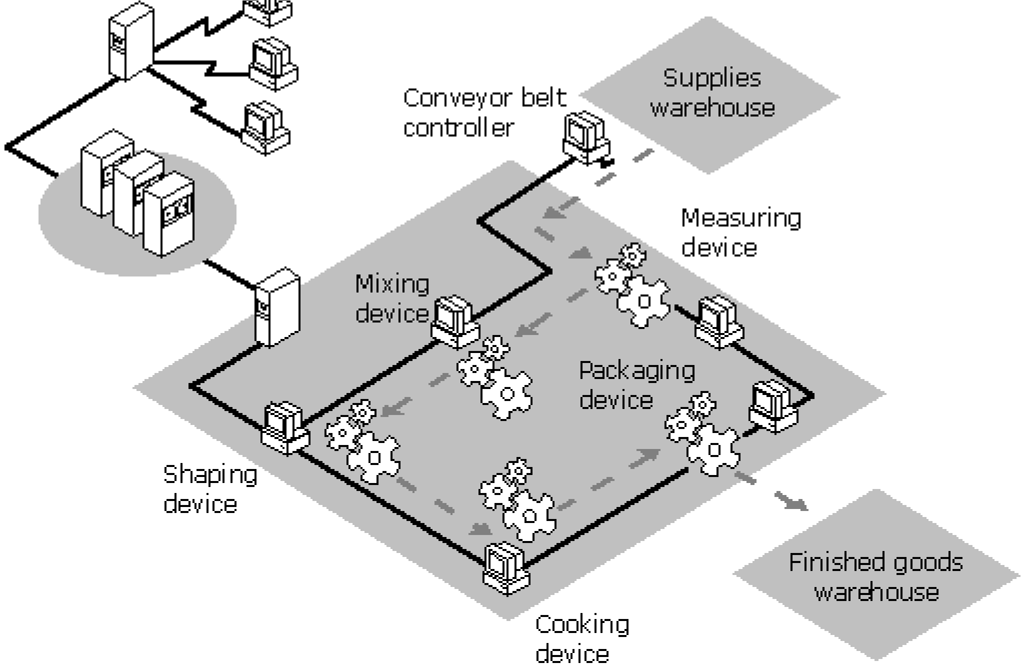
MSMQ dependent client



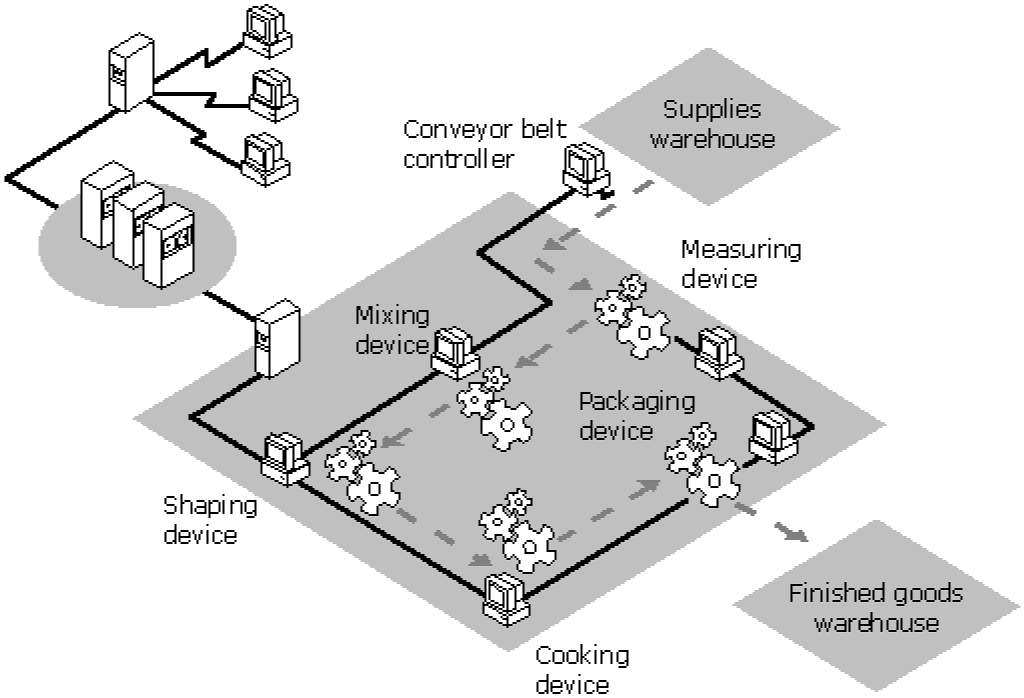
MSMQ independent client



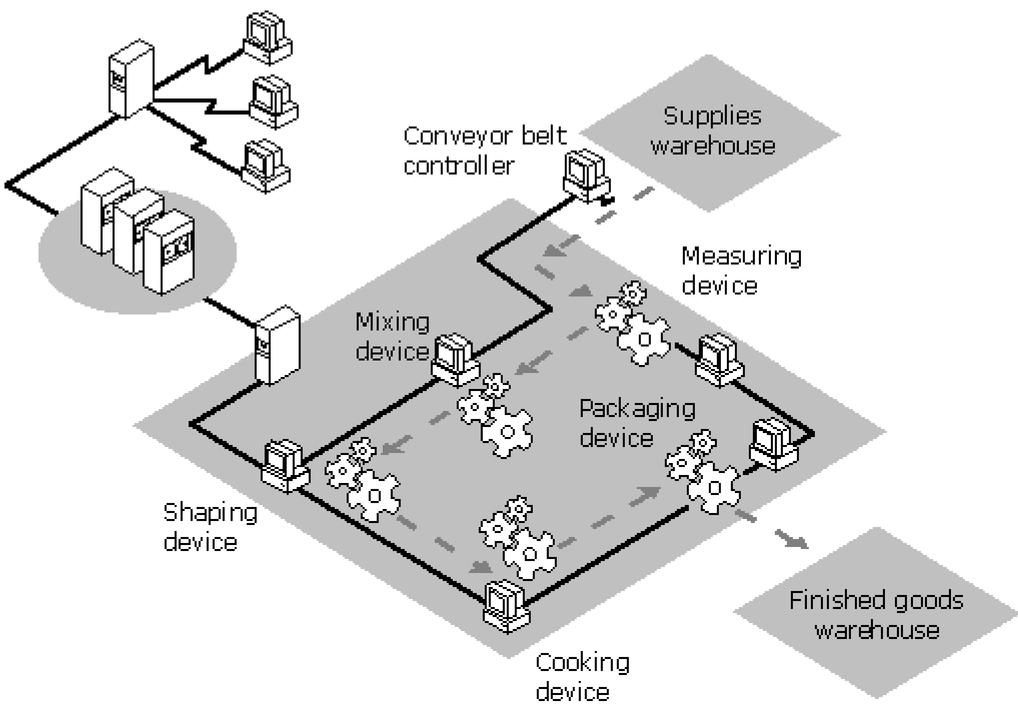
MSMQ RAS connectivity service



MSMQ routing server

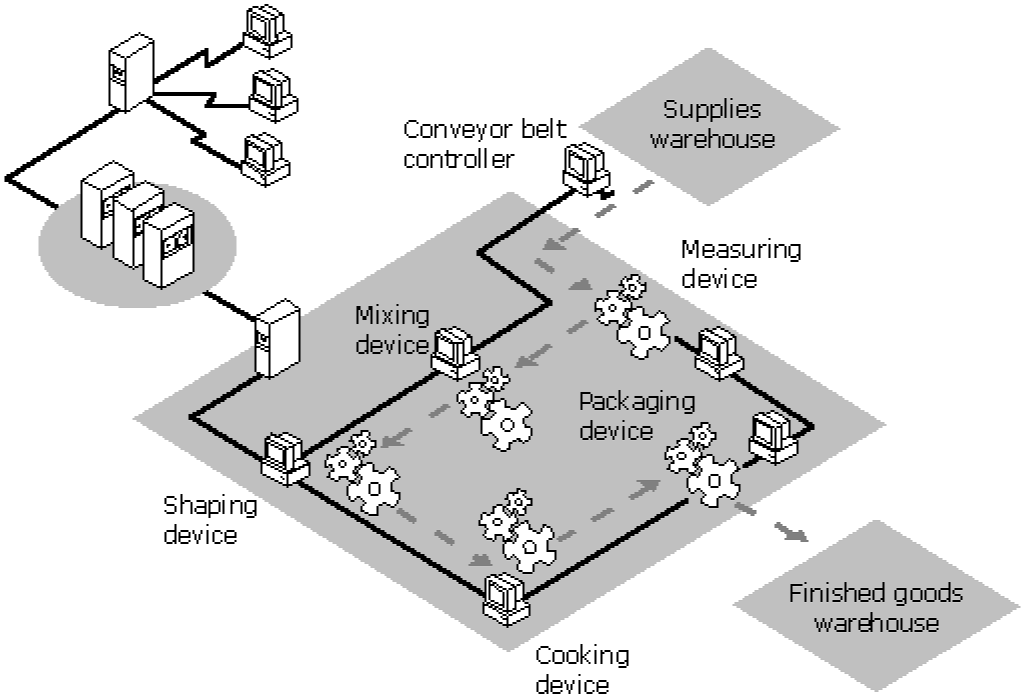


MSMQ service



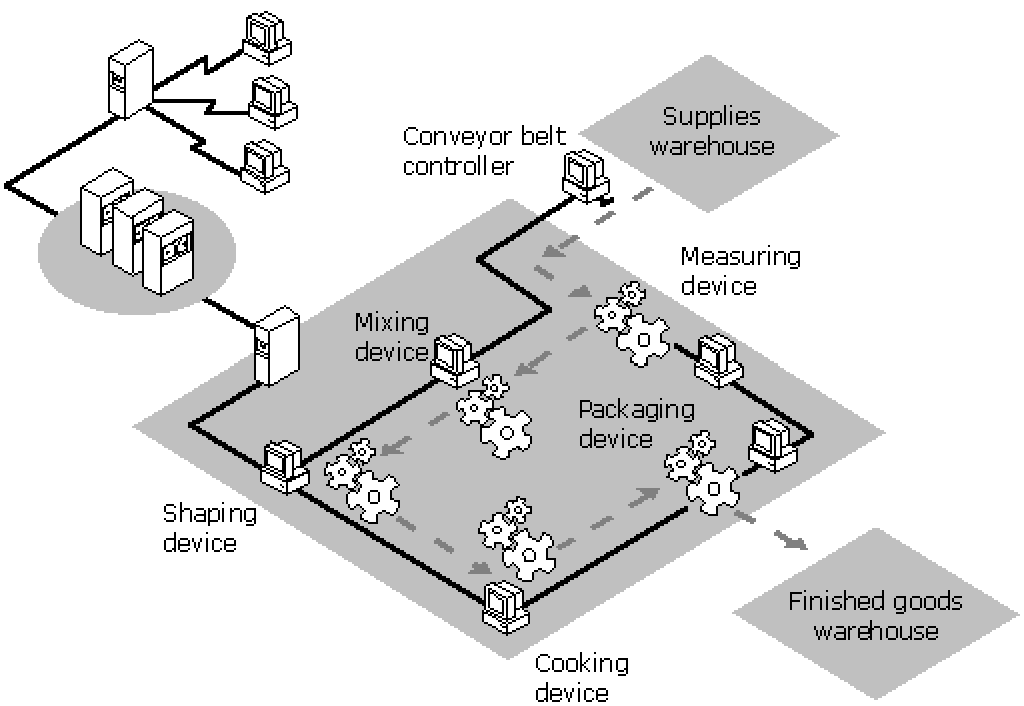
MSMQ server

N



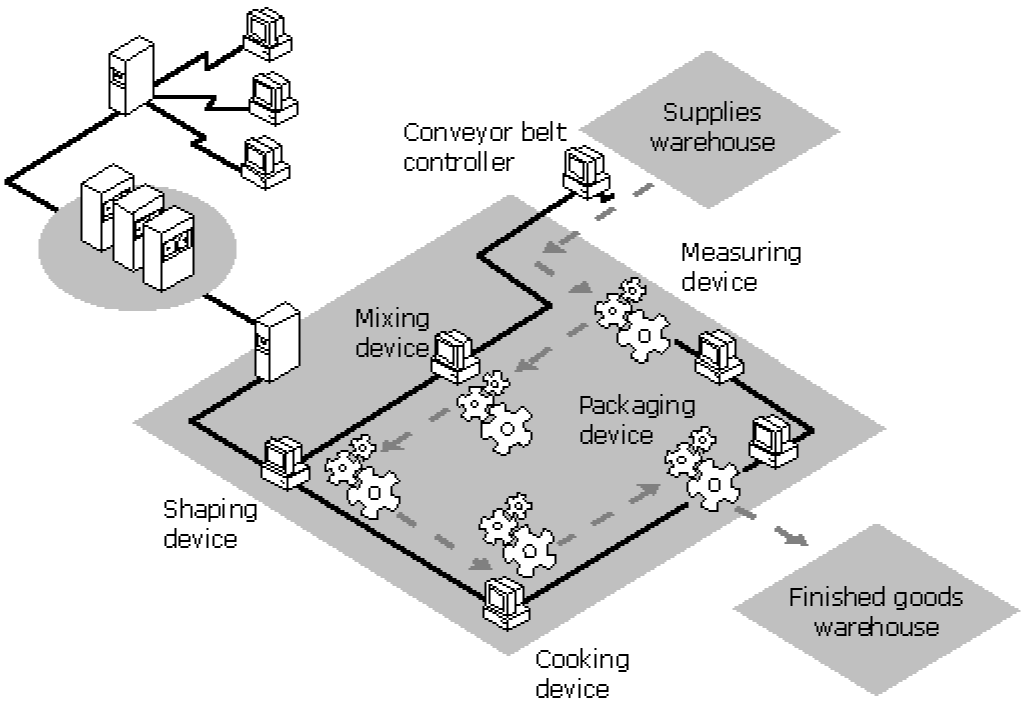
non-repudiation

o



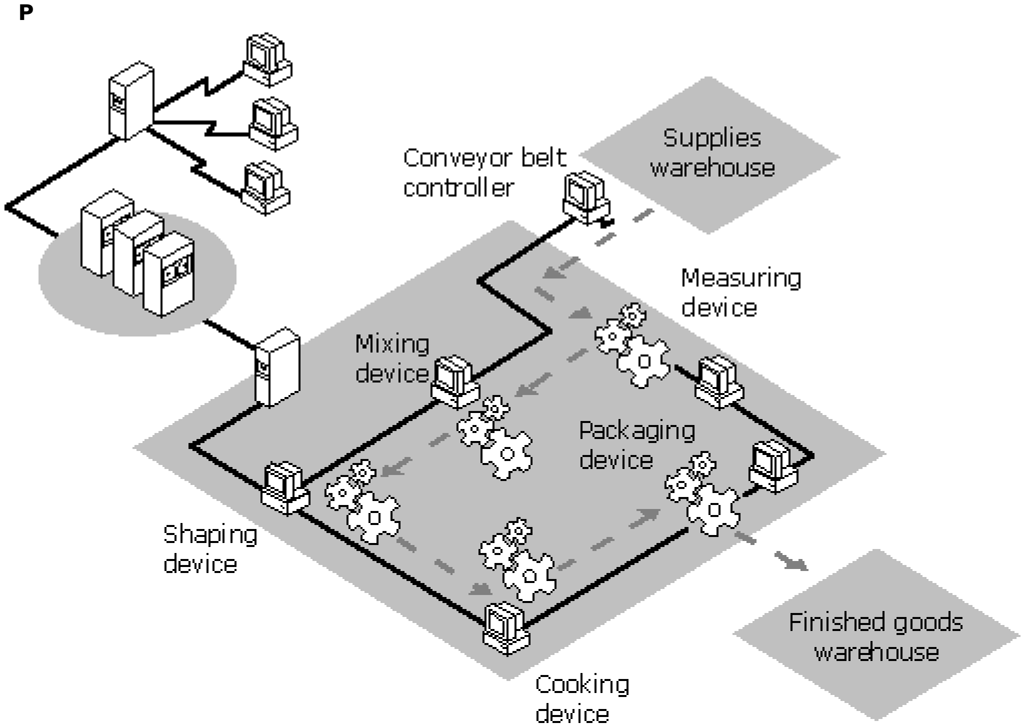
out routing

server

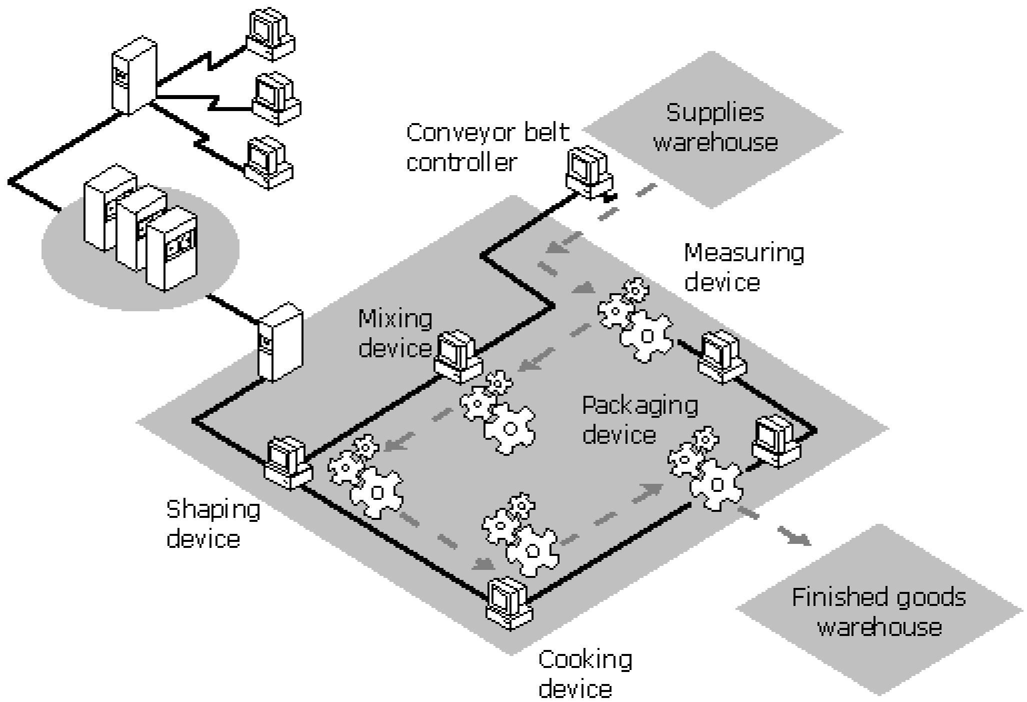

Supplies
warehouse
Conveyor belt
controller
Measuring
device
Mixing
device
Packaging
device
Shaping
device
Finished goods
warehouse
Cooking
device

OutRS

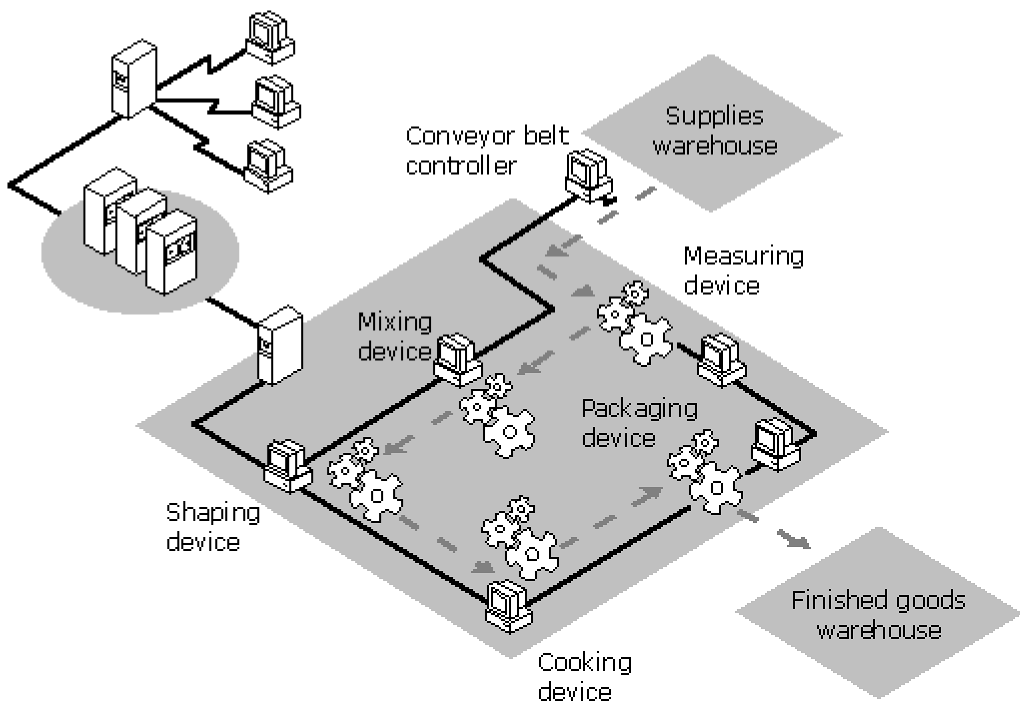
P


Supplies
warehouse
Conveyor belt
controller
Measuring
device
Mixing
device
Packaging
device
Shaping
device
Finished goods
warehouse
Cooking
device

PEC

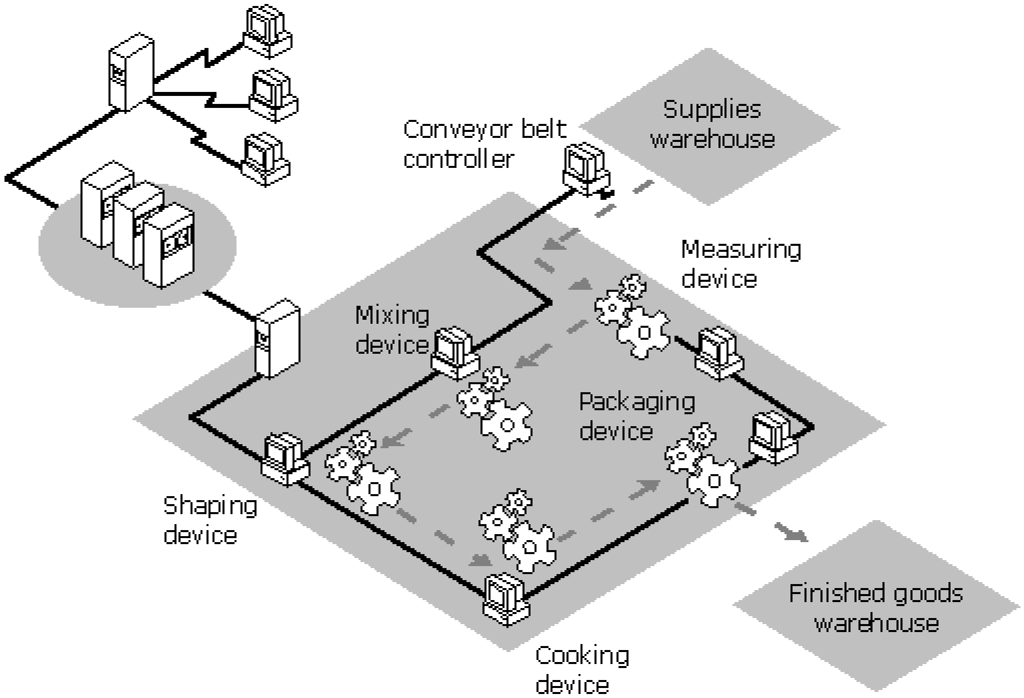


permission



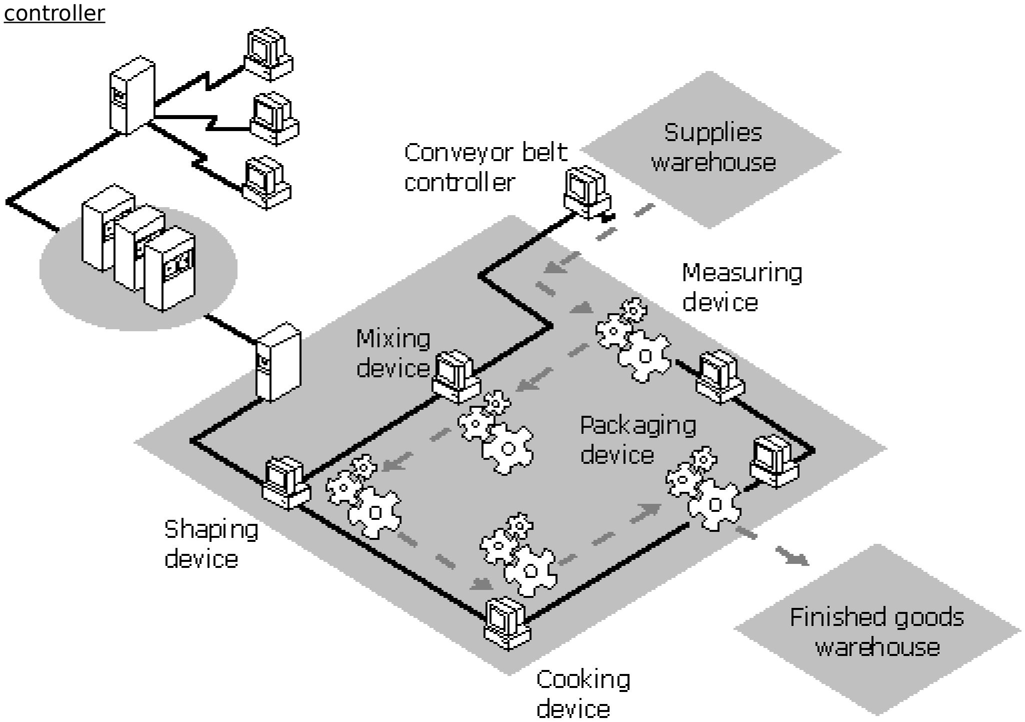
primary

enterprise controller

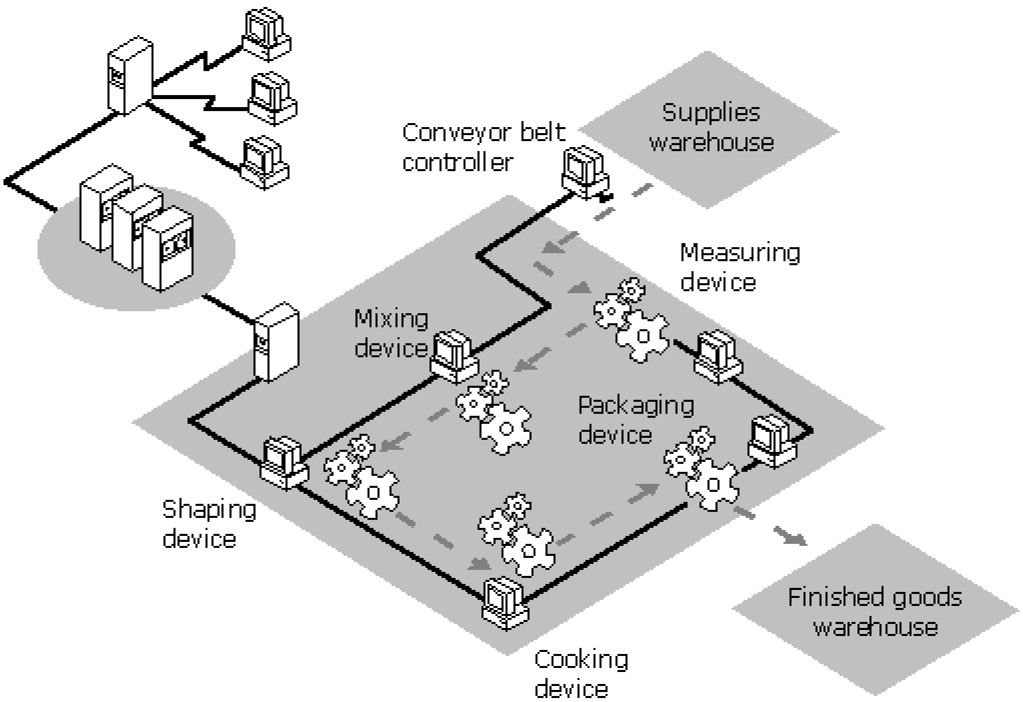


primary site

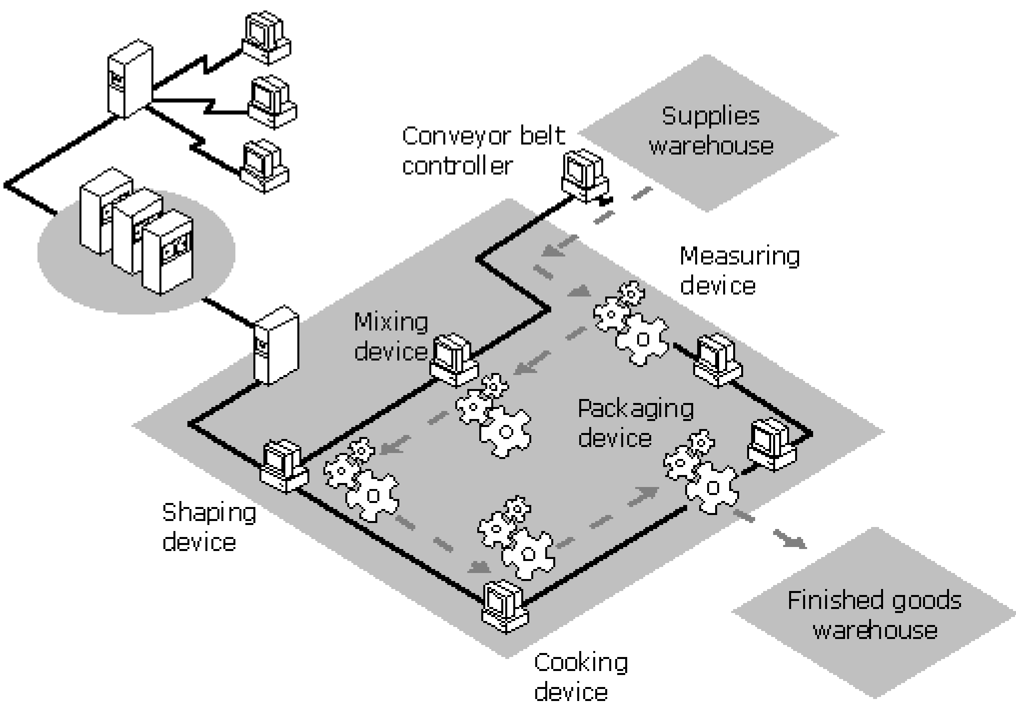
controller



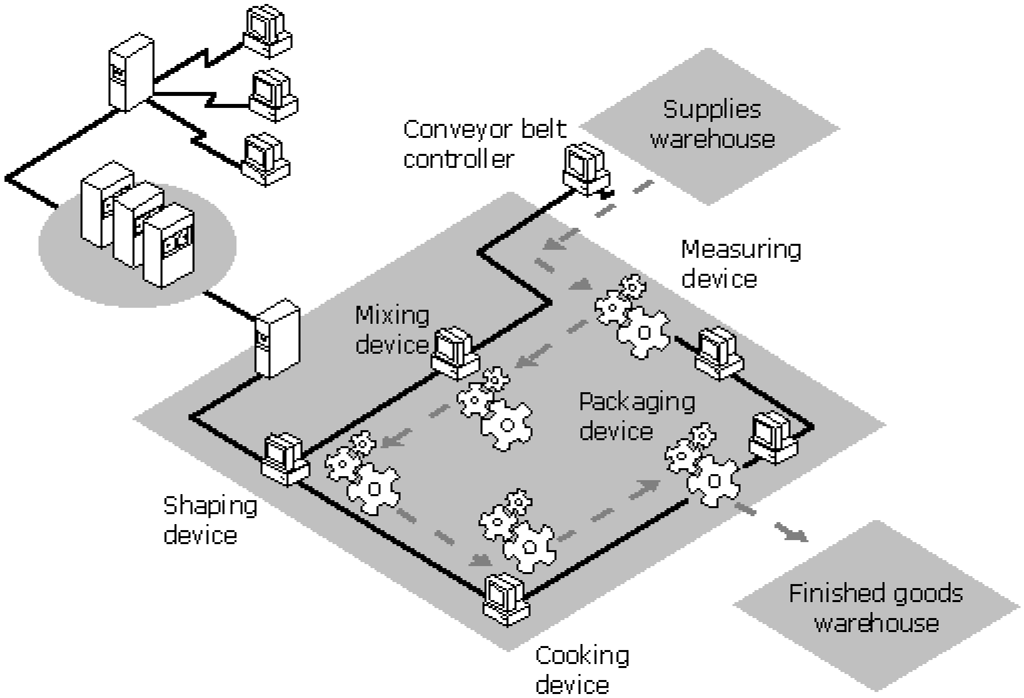
privacy level



private queues

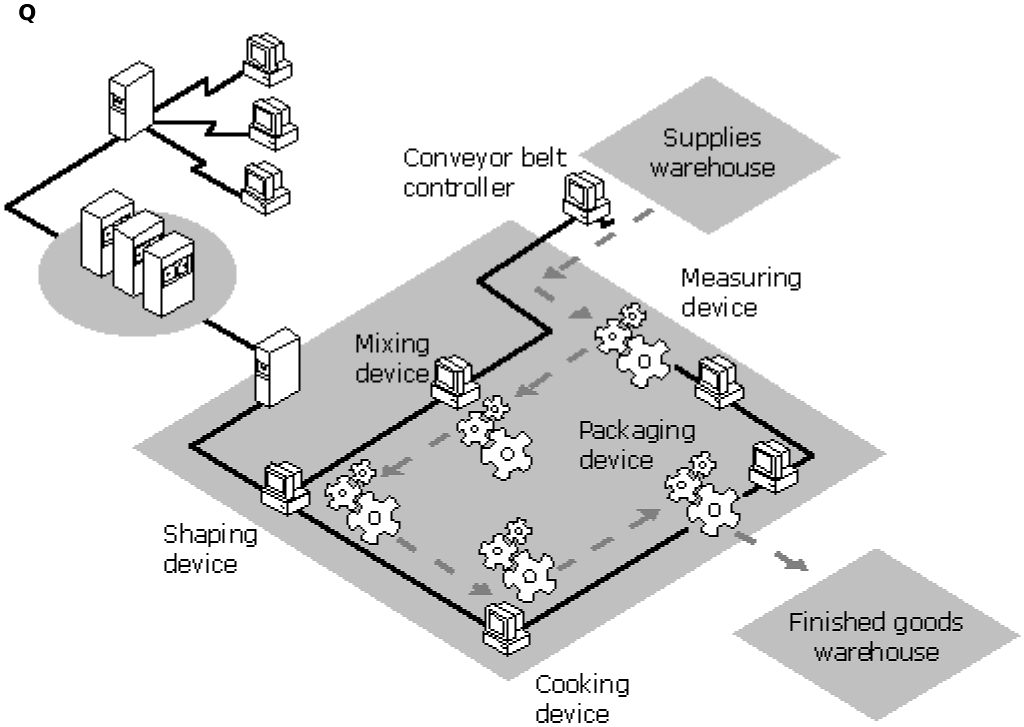


PSC

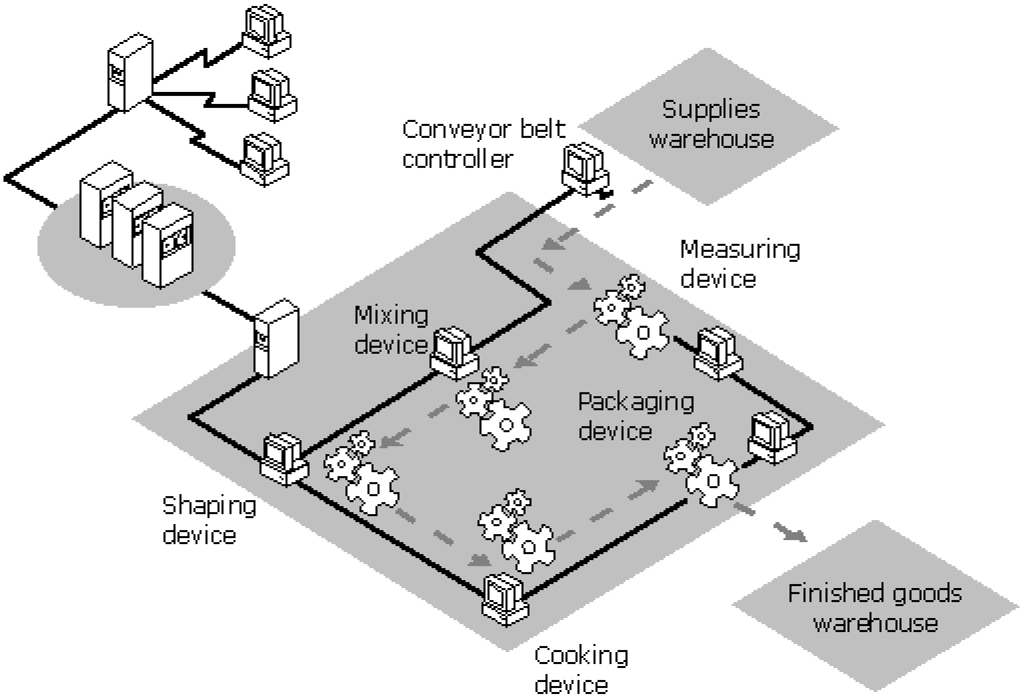


public queues

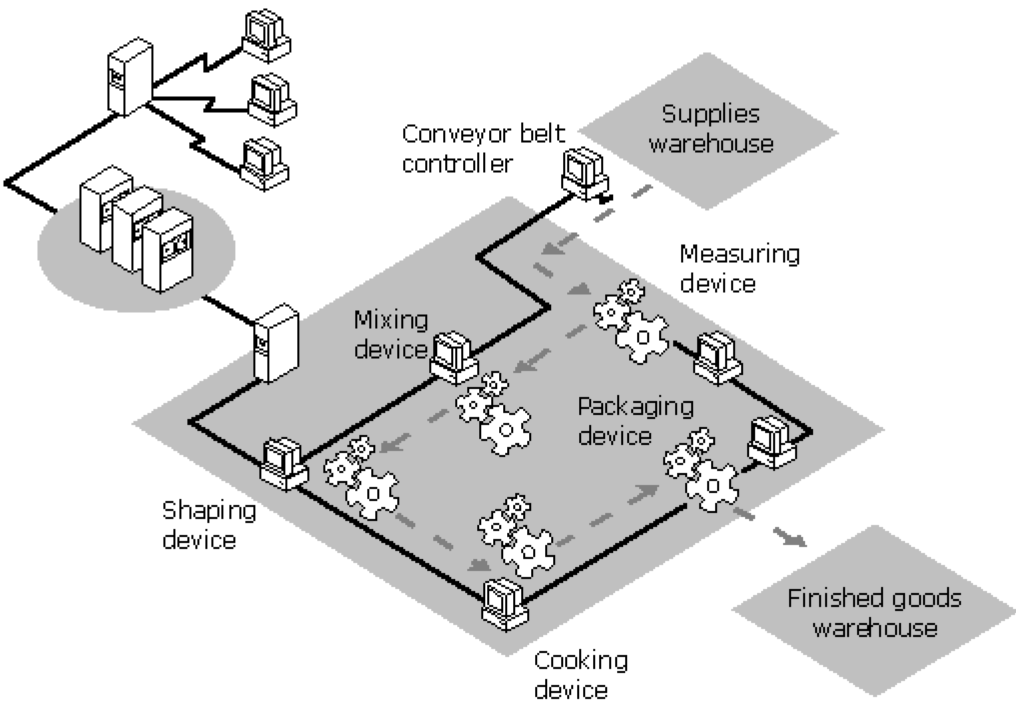
Q



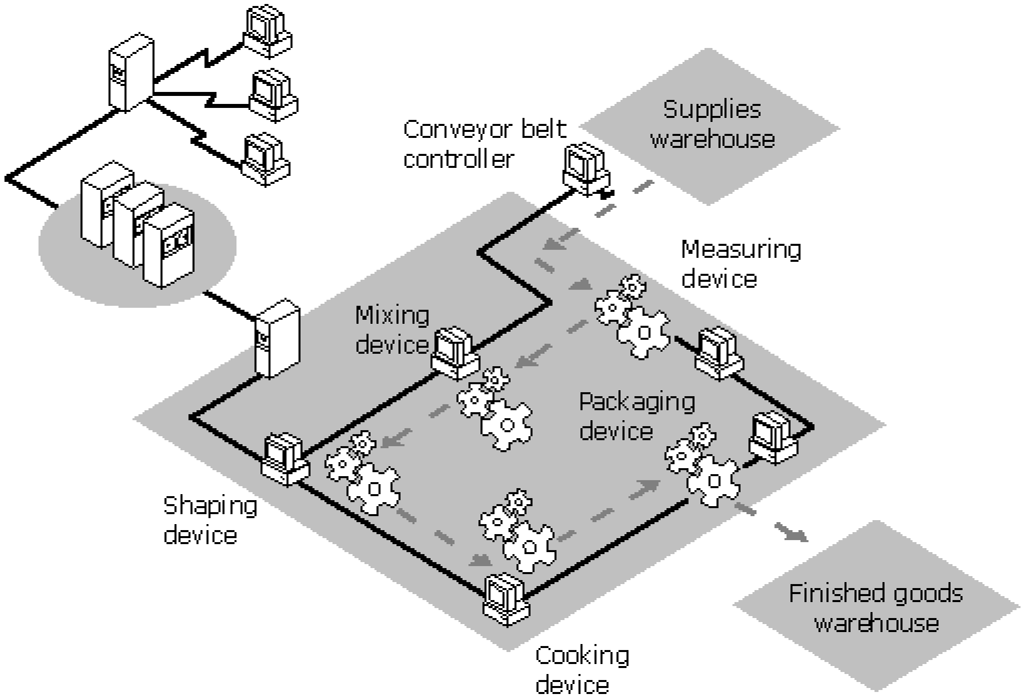
QM



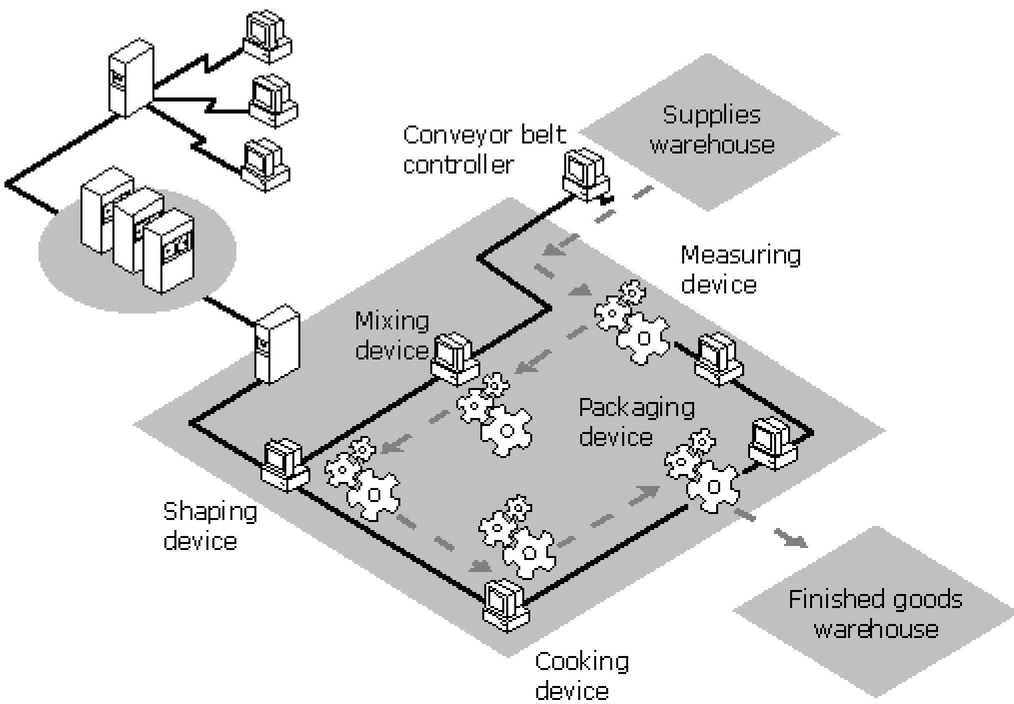
queue manager



queue pathname

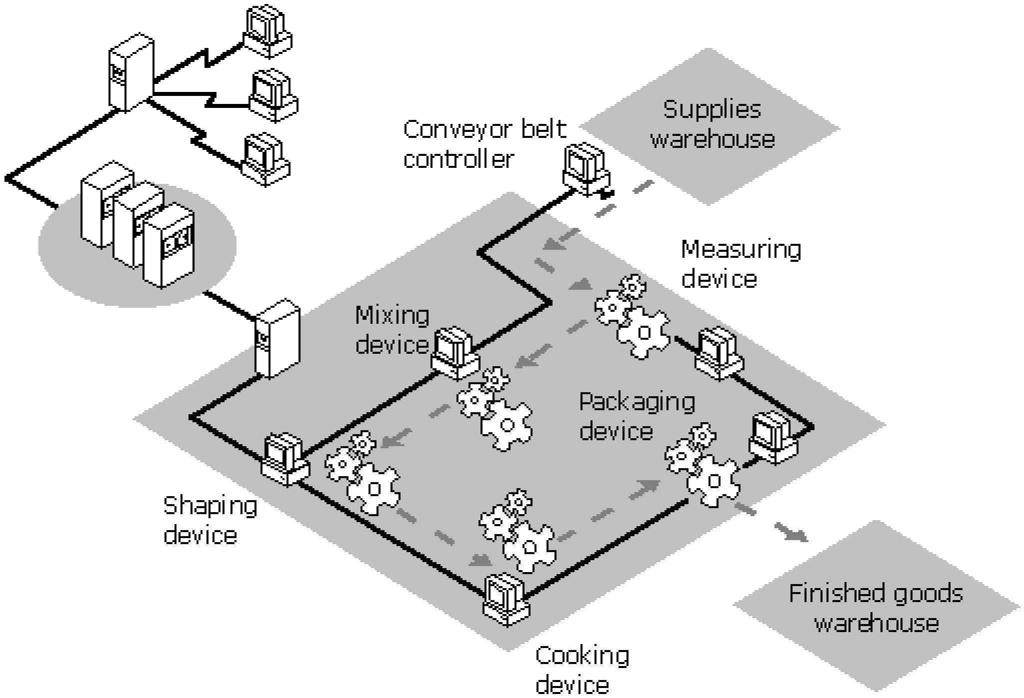


queue quota

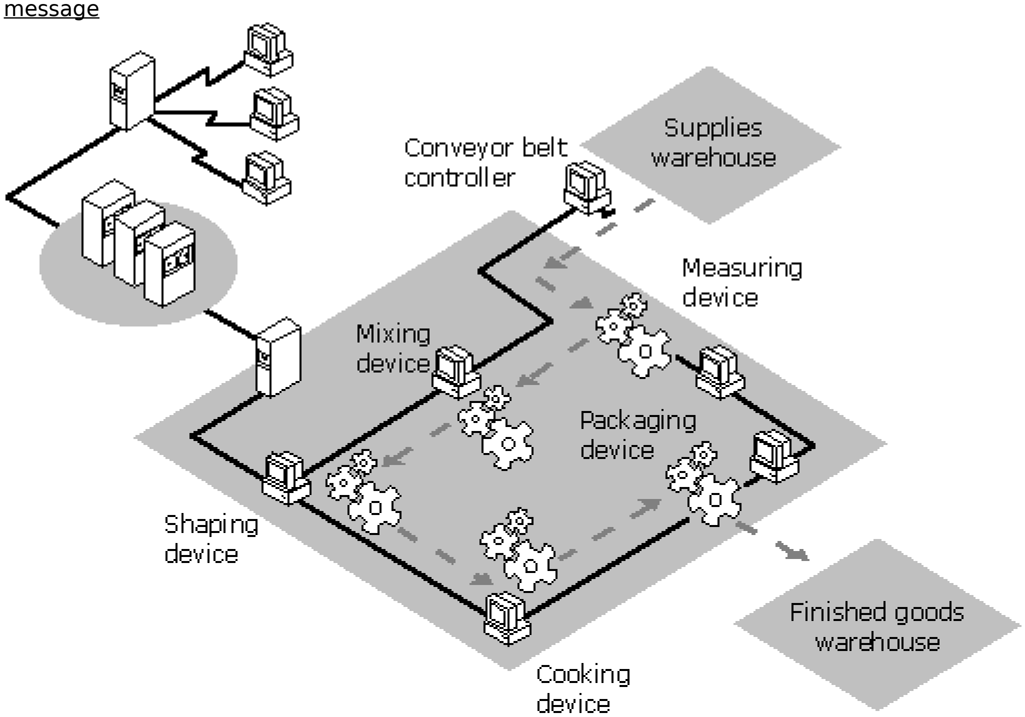


queue type

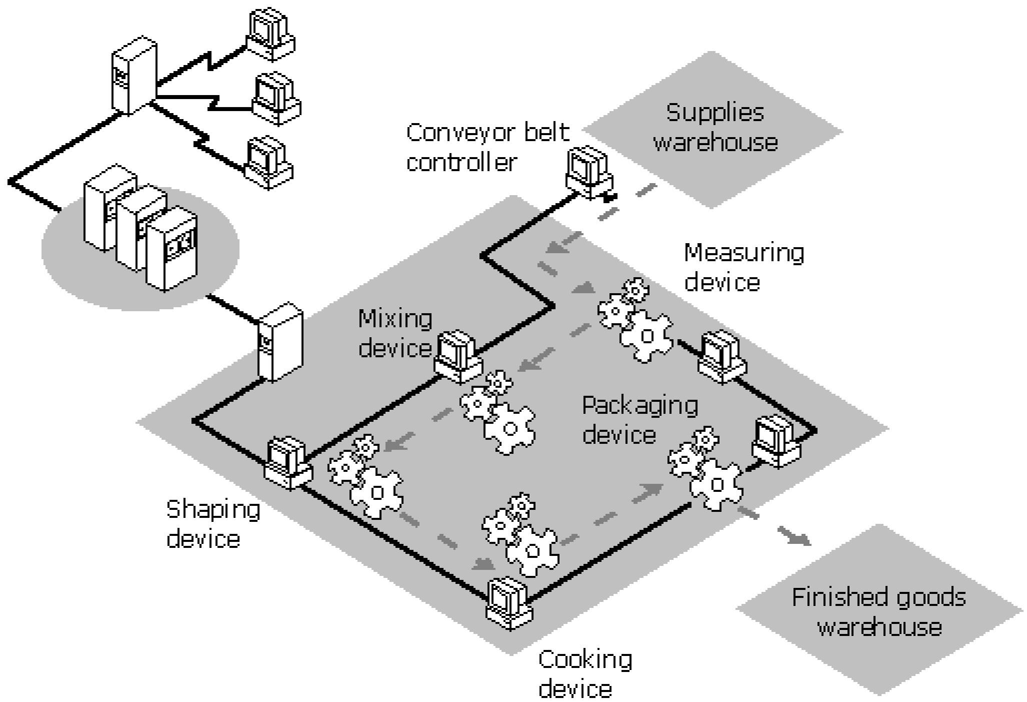
R



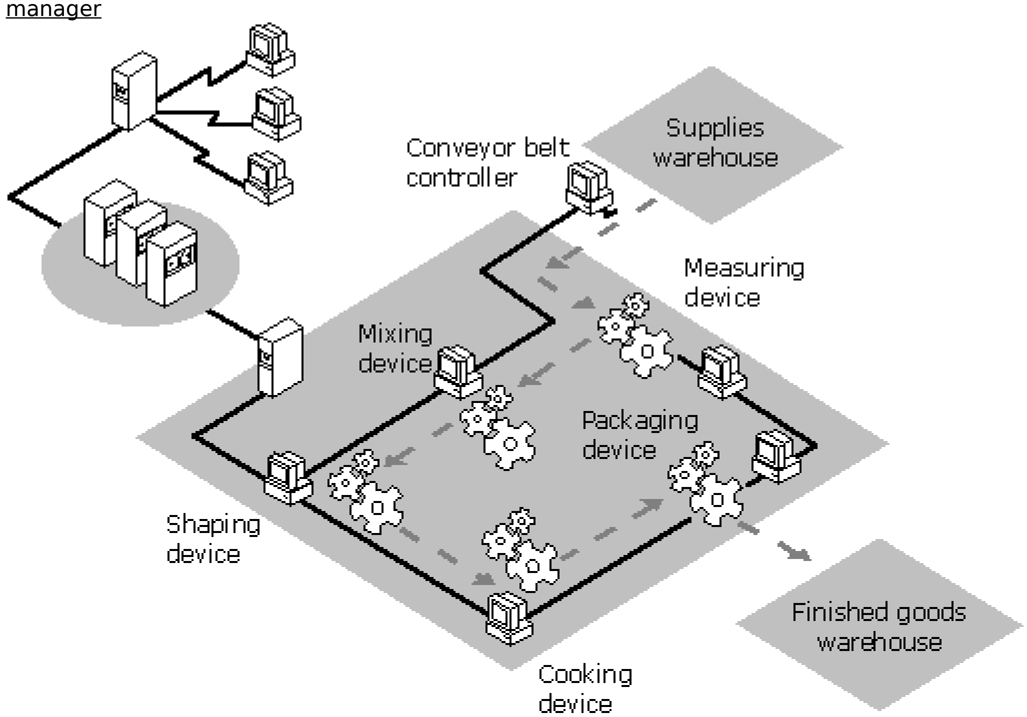
recoverable message



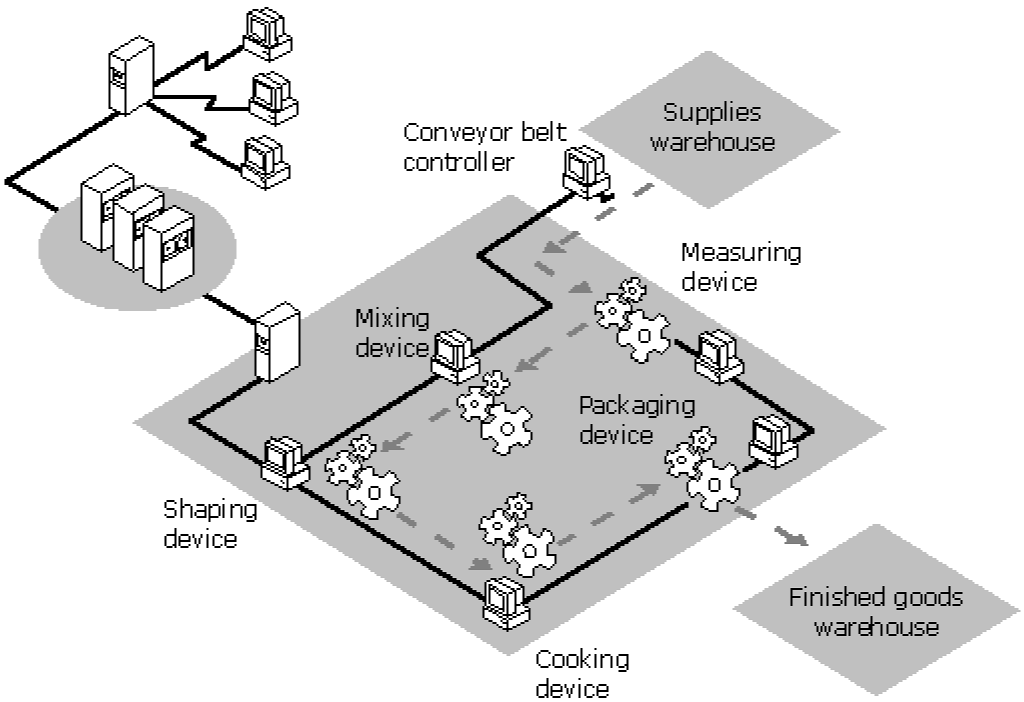
report queue



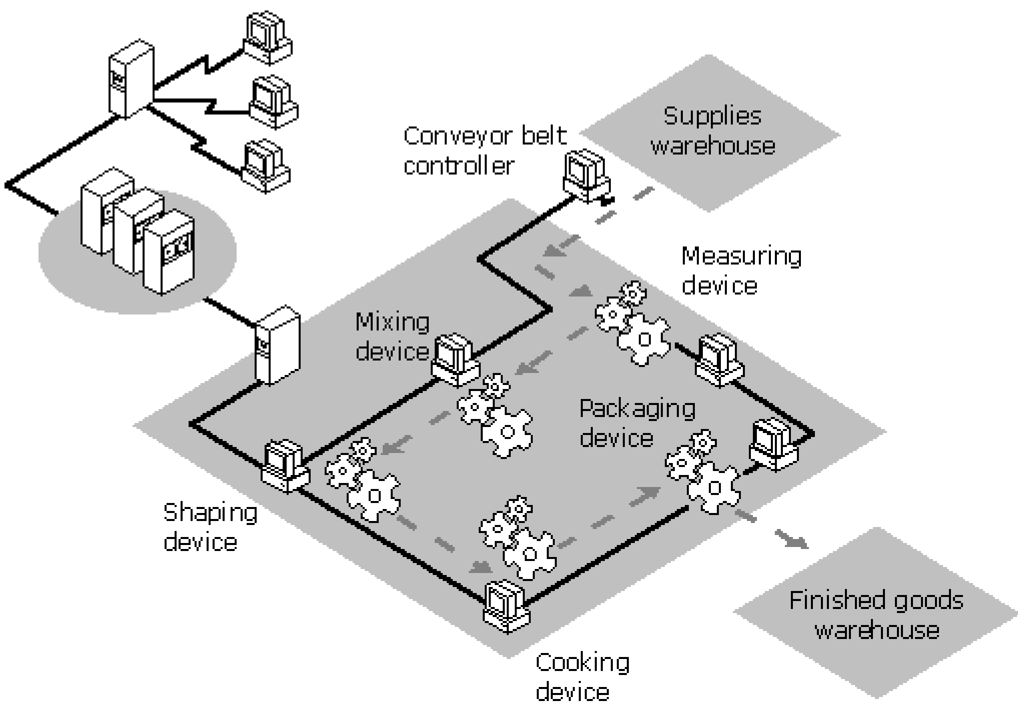
resource manager



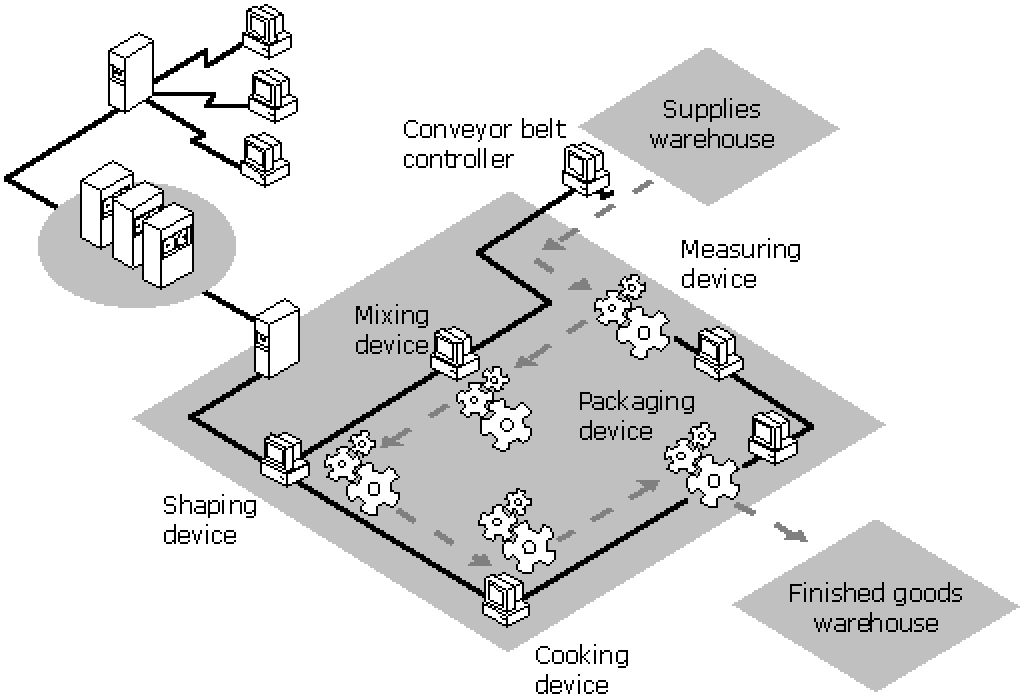
response queue



right

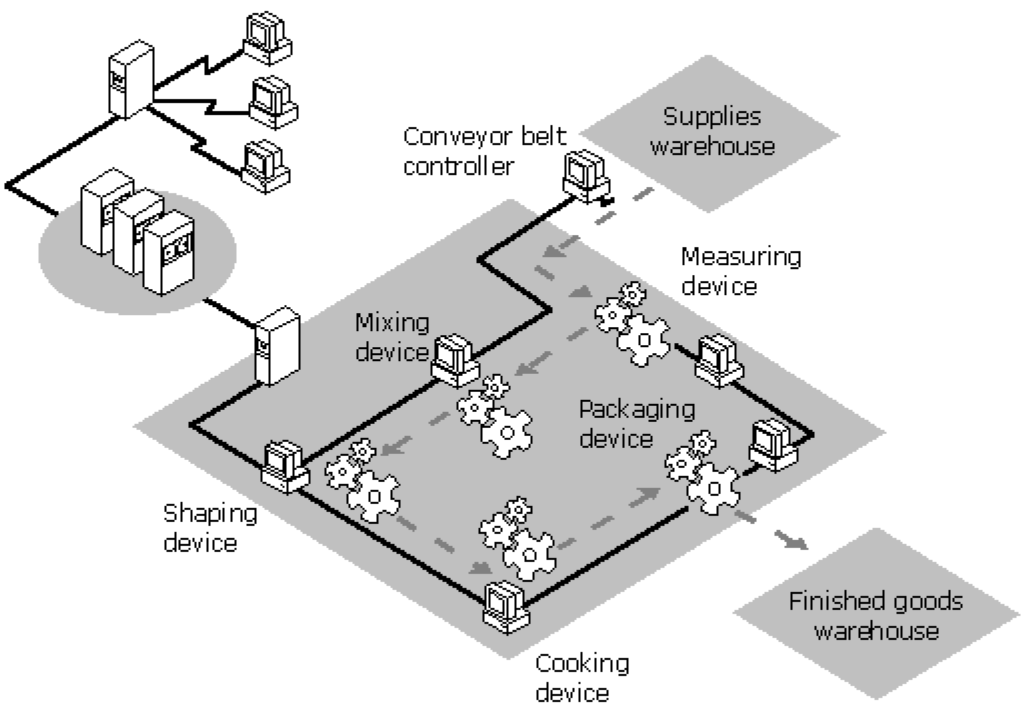


routing

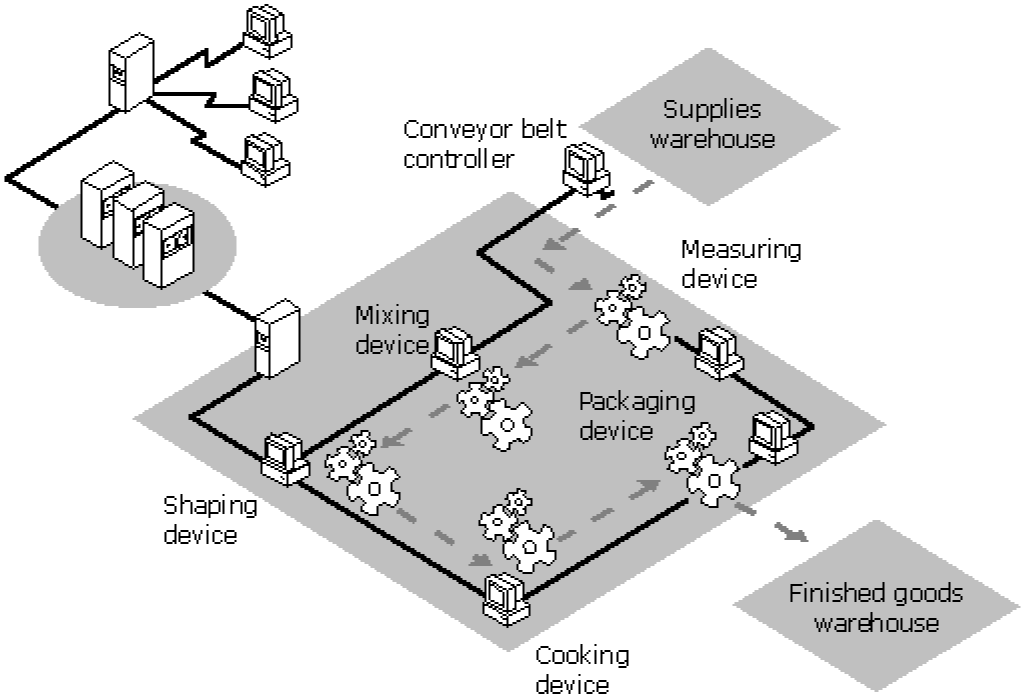


RSA

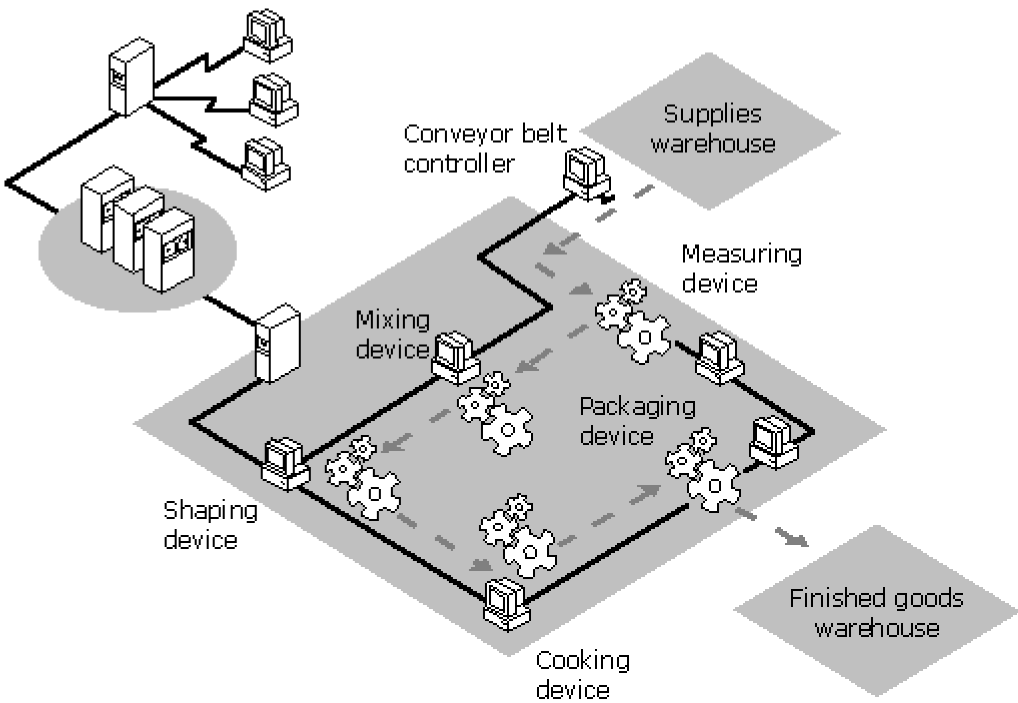
S



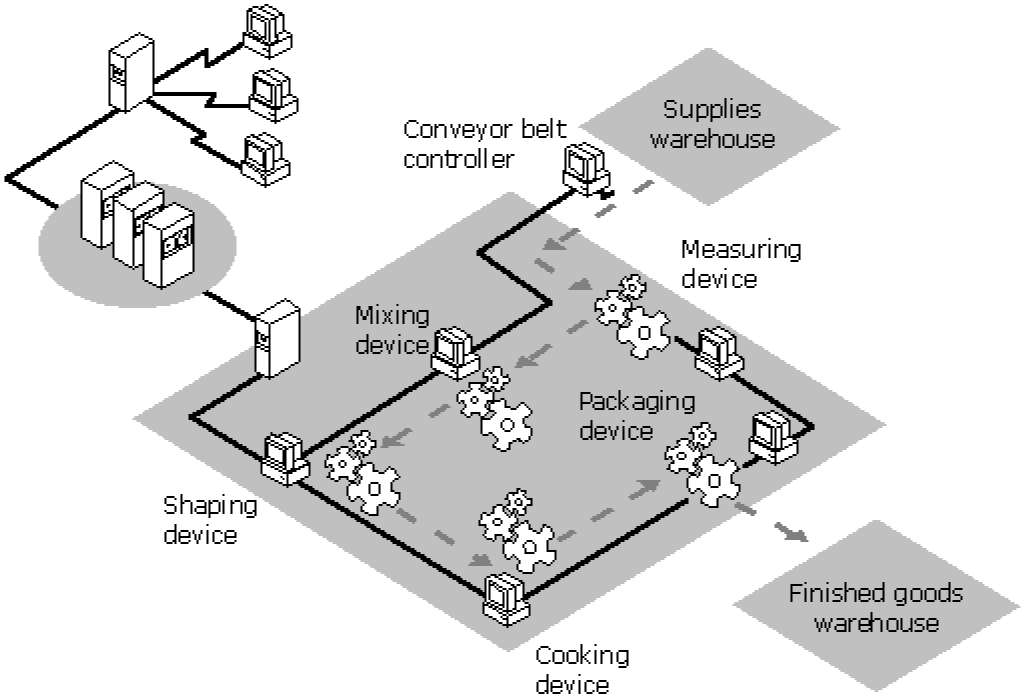
site pathname



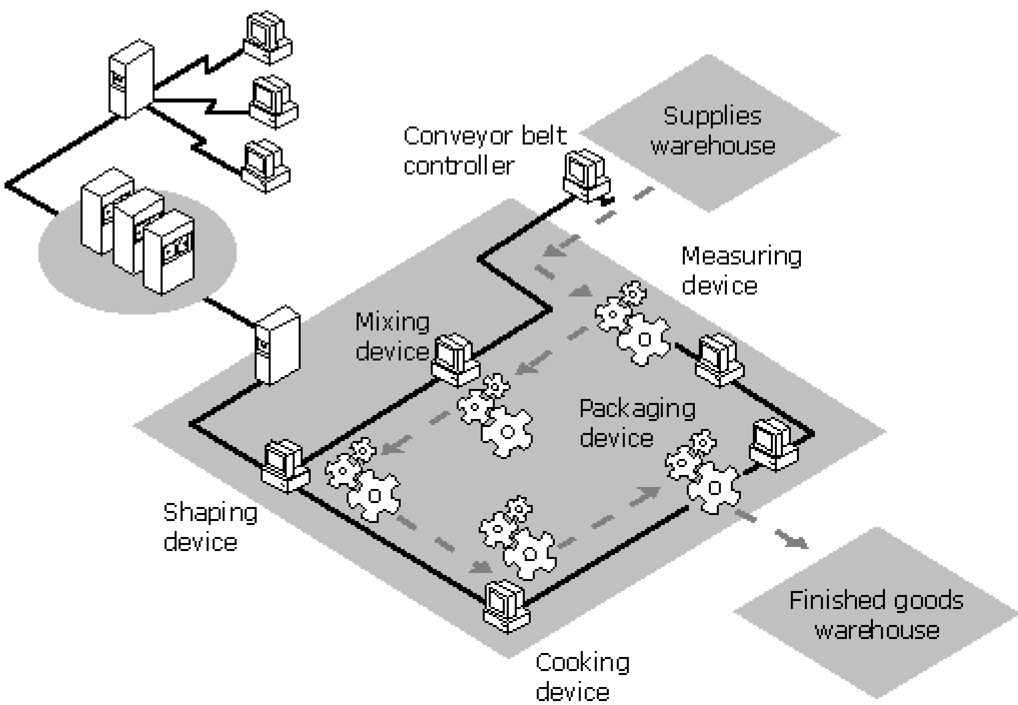
SDK



sender ID

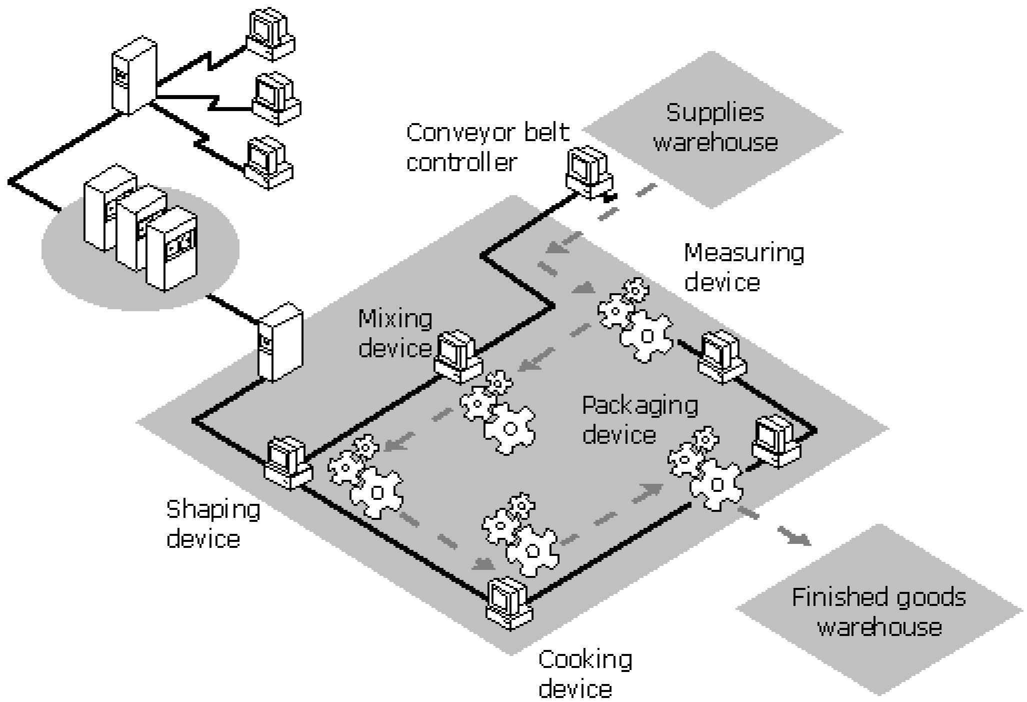


session

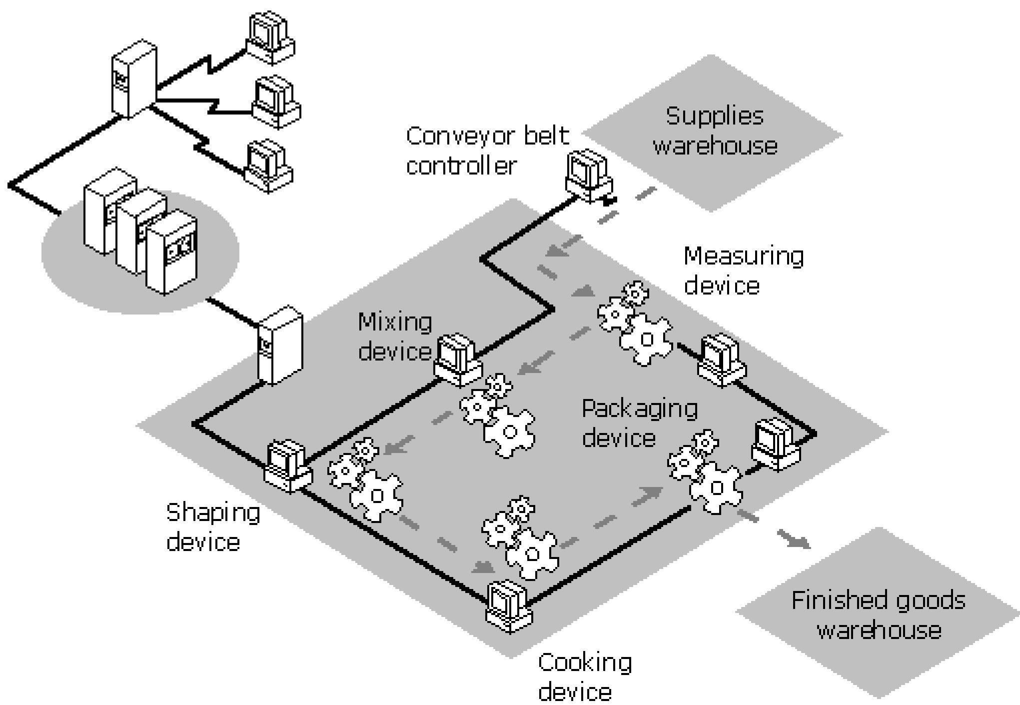


session

concentration

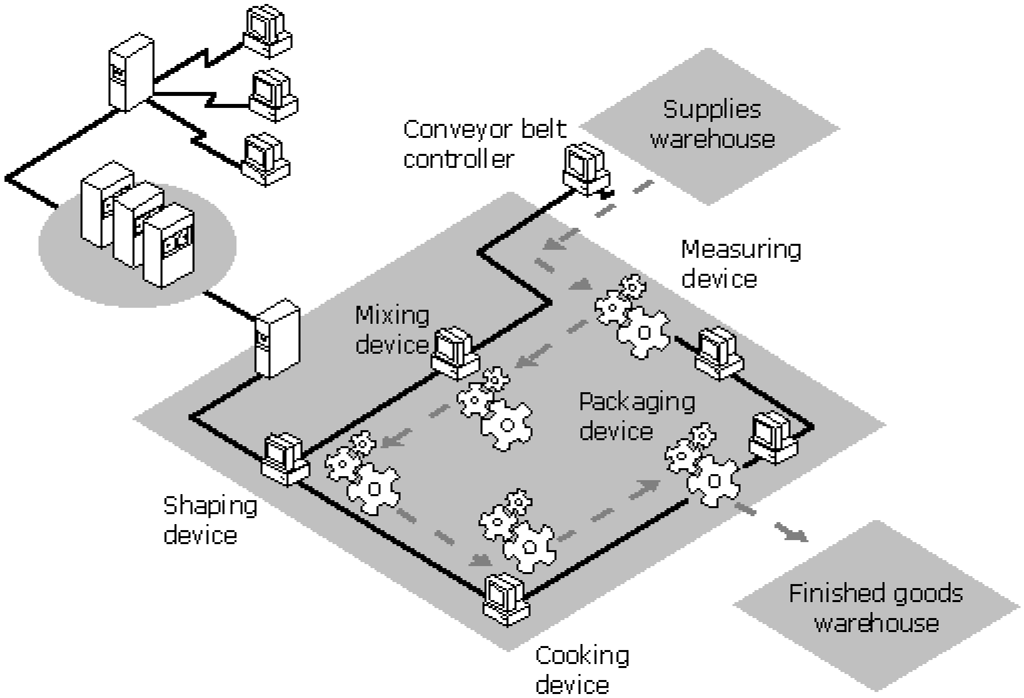


site

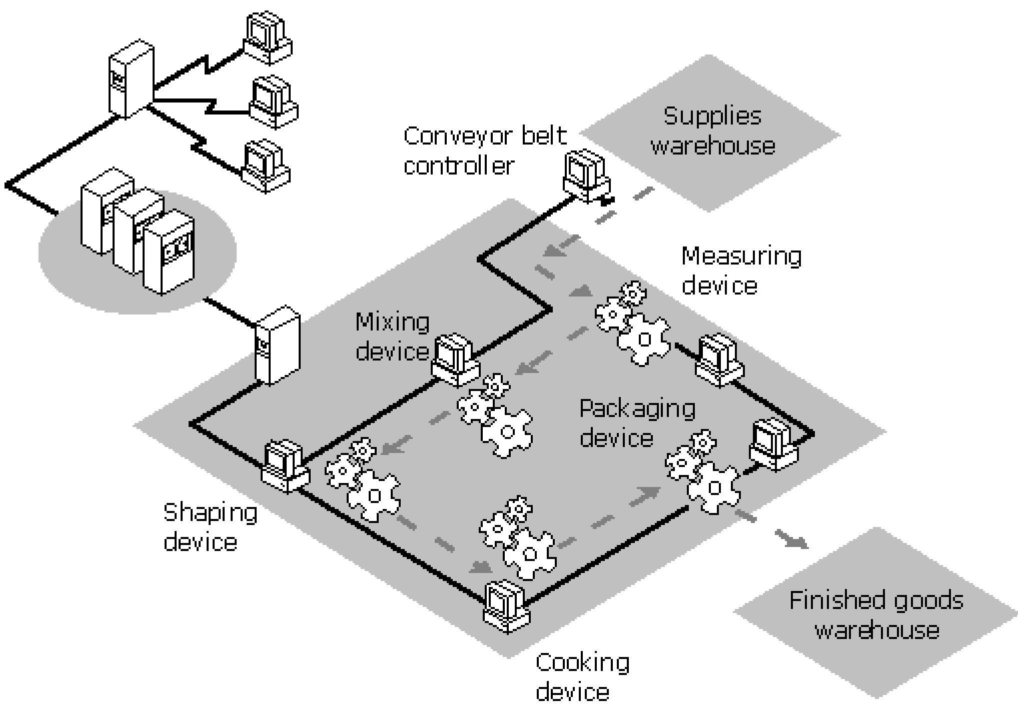


site controller

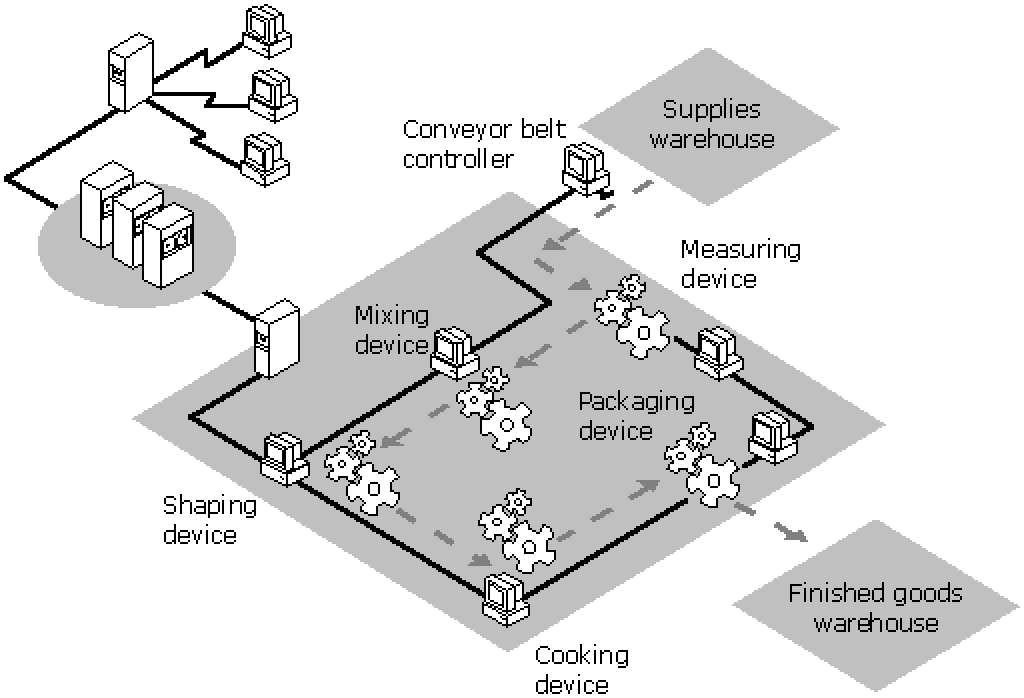
server



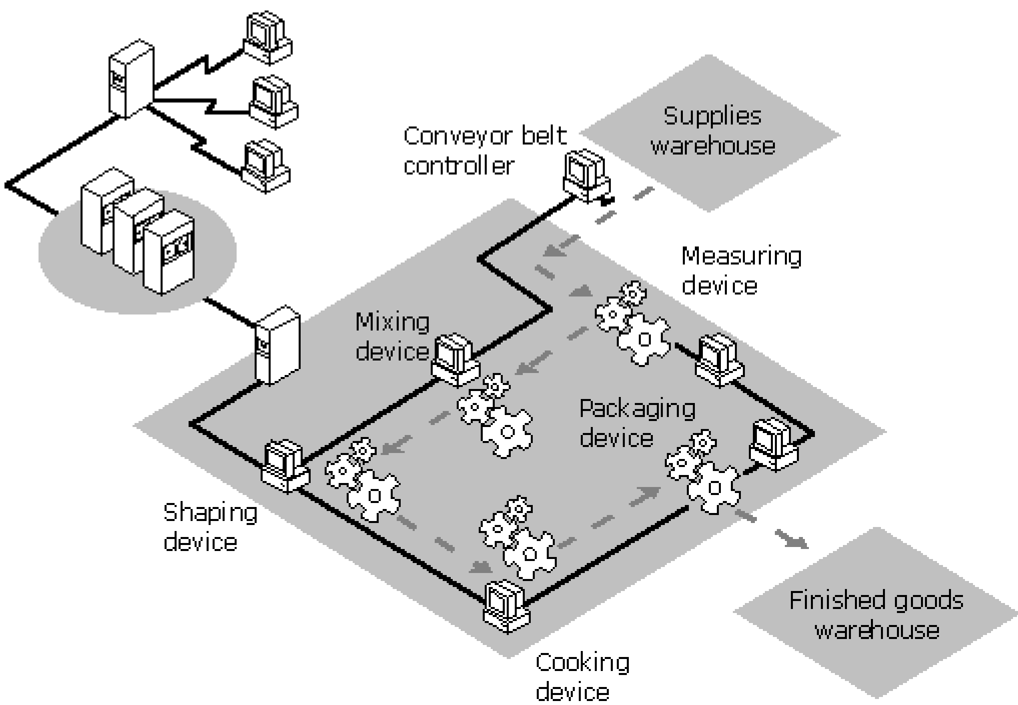
site gate



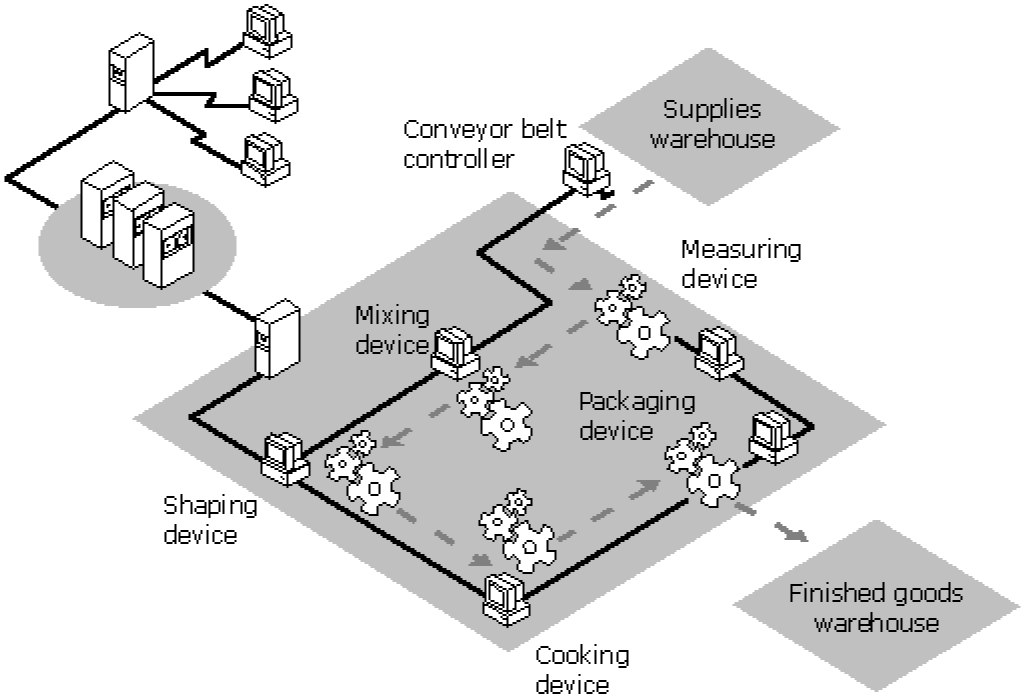
site link



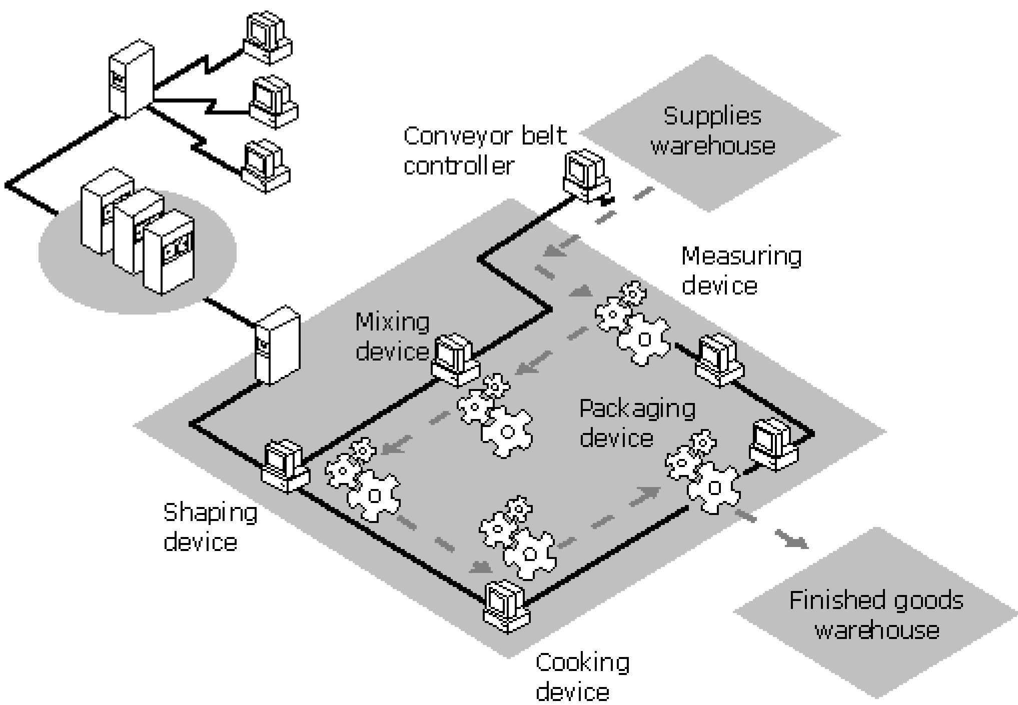
site link cost



source journaling

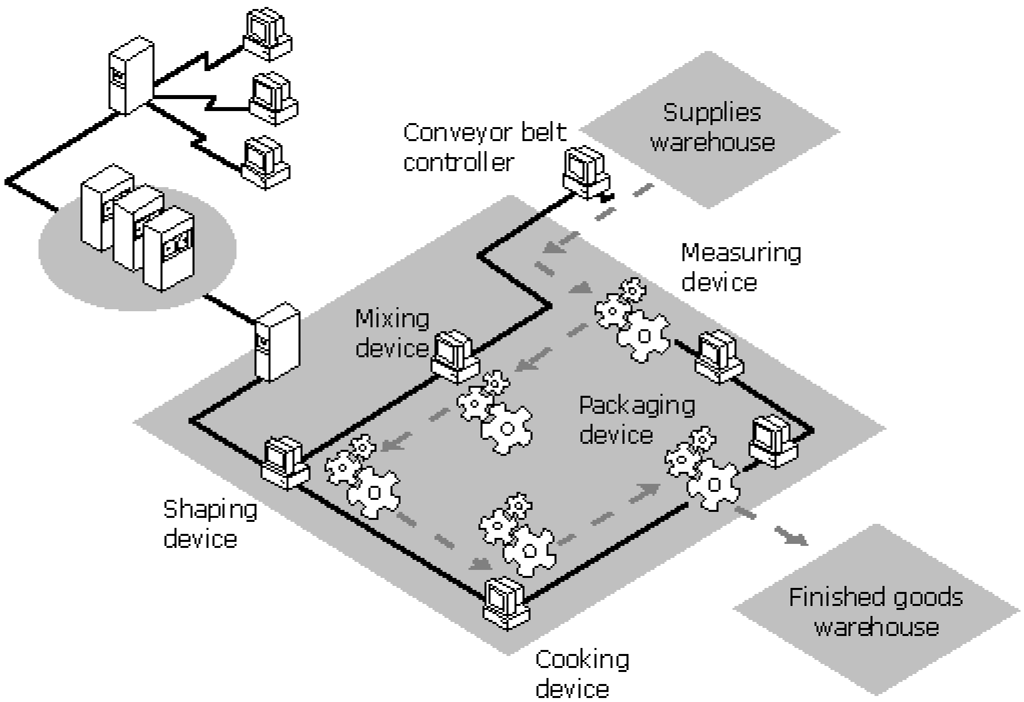


source computer

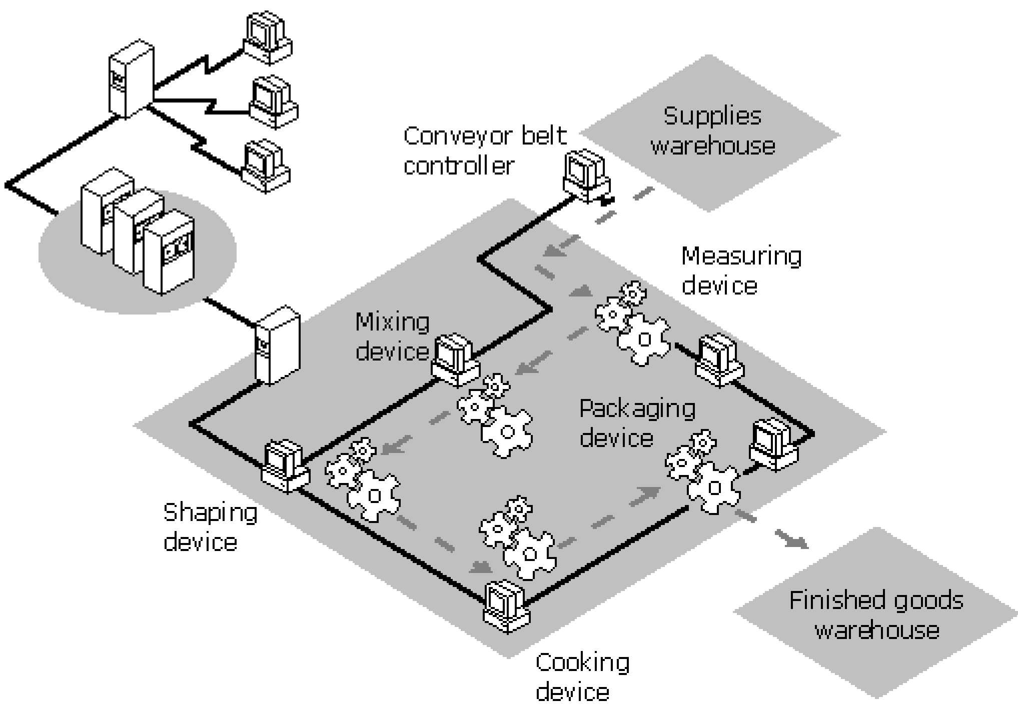


supporting server

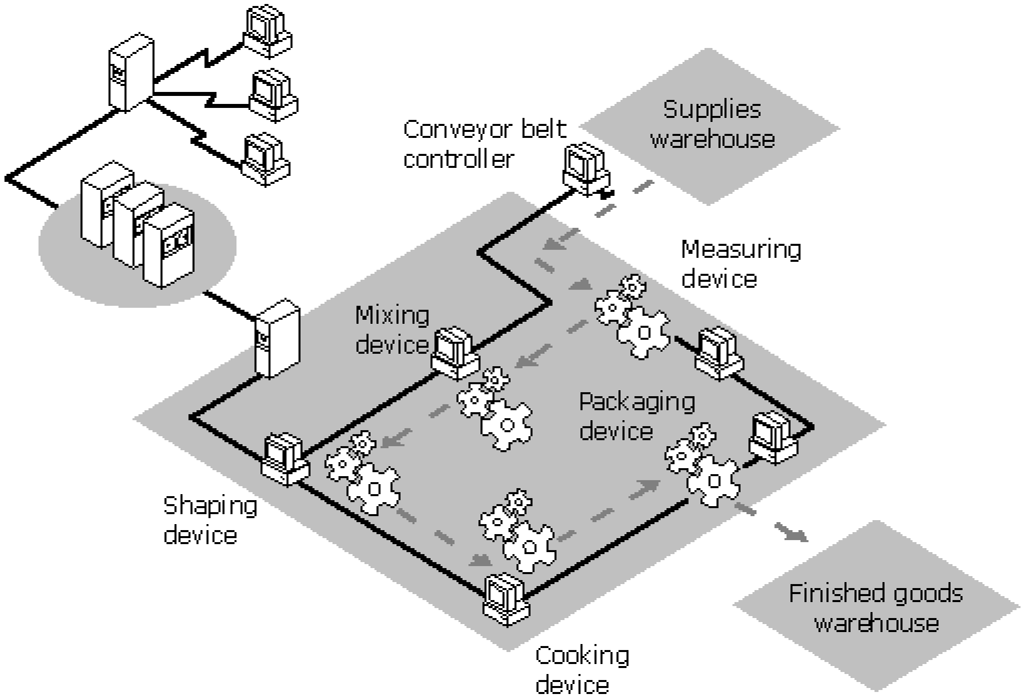
T



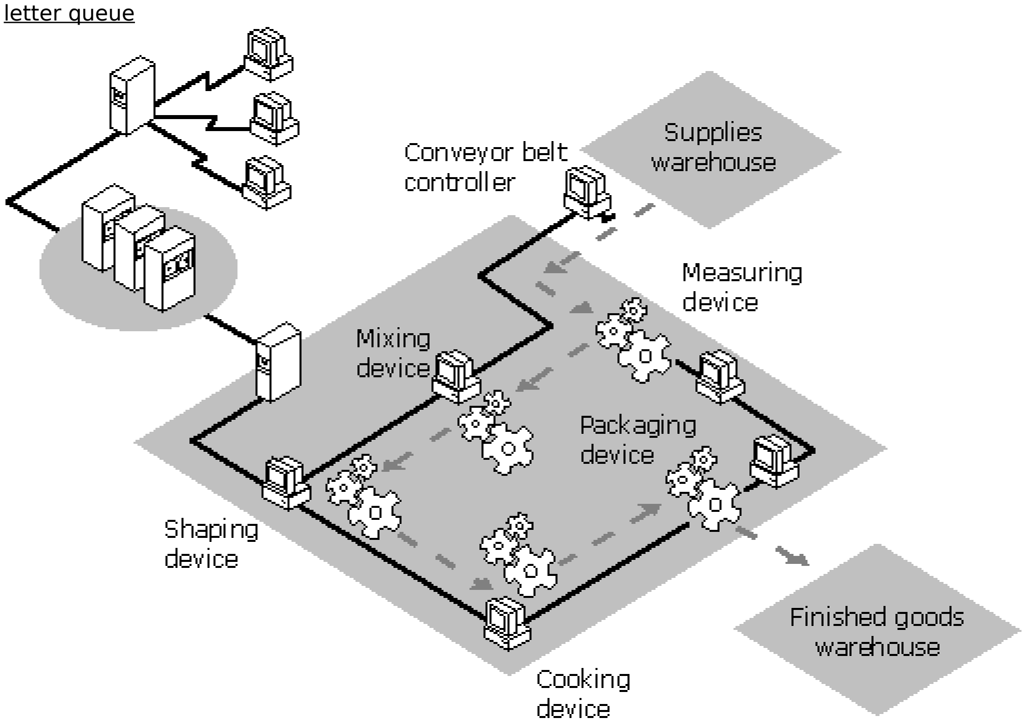
target journaling



transaction



transaction dead letter queue



type ID

U

There are no glossary entries for this section.

V

There are no glossary entries for this section.

W

There are no glossary entries for this section.

X

There are no glossary entries for this section.

Y

There are no glossary entries for this section.

Z

There are no glossary entries for this section.

ActiveX controls

You can use the ActiveX controls provided by MSMQ and Visual Basic (VB) or any other ActiveX container application (for example, Microsoft Access) to quickly and easily write MSMQ-based applications.

administration queue

MSMQ uses administration queues to pass acknowledgment messages. Acknowledgment messages, generated by MSMQ, are used by the application that sent the original message. These messages indicate that the original messages either arrived (a positive acknowledgment) or that an error occurred before the original message could be retrieved (a negative acknowledgment). A positive acknowledgment message might also indicate whether the original message was retrieved by the target application. A negative acknowledgment message might also indicate why the original message was not received or retrieved.

The application programmer determines whether to use acknowledgments. If acknowledgments are used, the application programmer must specify an administration queue in the source application and the type of acknowledgment messages to send to the queue.

API

Application programming interface.

authentication

Authentication (also called signing) is the process of verifying that the message sender is authentic (in other words, is not being impersonated by another user). MSMQ uses digital signatures (commonly called certificates) to provide message authentication.

Class 1 or Class 2 digital signatures must be obtained from a cryptographic service provider (CSP) such as AT&T or Verisign. The CSP does not have to provide full *RSA* support, only a signing. Any signing CSP can be used, as long as it is installed on both the sending and receiving computers. To use non-internal certificates the computer must have Microsoft® Internet Explorer (IE) 3.0 installed.

backup site controller

Sites do not require backup site controllers (BSCs). However, one or more BSCs can be installed at each site to provide load balancing and failure recovery, should the PSC or PEC fail. The BSC holds a read-only replica of the PSC or PEC MQIS database and also functions as an MSMQ routing server.

For information on the MQIS database and replication, see the *Microsoft Message Queue Server Administrator's Guide*, Chapter 1, Understanding MSMQ.

base priority

The base priority of a public queue specifies the queue’s priority. The base priority can be set by any MSMQ-based application with write permissions for the queue. The base priority can be set any time. The public queue priority can be set from -32768 to 32767 with the default priority of 0. Private queues do not support queue priority.

MSMQ routes and delivers messages based on a combination of queue priority and message priority. Messages are routed and delivered by queue priority first, and message priority second.

BSC

See *backup site controller*.

computer ID

A GUID created by MSMQ when the computer joins the enterprise.

computer pathname

The same as the computer name.

computer quota

Specifies the cumulative limit for message size on a computer, based on the total size of the messages. The computer quota can be set independently of the queue quota. When a computer quota is reached, messages can no longer be sent to the computer until one or more messages are removed from queues.

MSMQ enforces the computer limit no matter how many queues are opened, or the cumulative queue quotas. For example, if you specify a 10 MB limit for each of the six public queues on a computer, and a 50 MB total limit for a computer, MSMQ enforces the 50 MB computer limit even if no queue has reached its 10 MB queue quota limit. However, each queue quota still prevents any single queue from storing more than 10 MB of messages.

connector queue

Created on MSMQ connector servers. You can use the MSMQ connector server to exchange messages with computers that are not running MSMQ (called foreign computers). The user account running the MSMQ connector must have the Open Connector permission on the foreign CN to open a connector queue.

controller server

Refers to PECs, PSCs, and BSCs. *Site controller server* refers to just the enterprise PEC and any PSCs.

CN

Connected network. See *connected network* for more information.

connected network

A collection of computers where any two computers can communicate directly.

CSP

Cryptographic service provider. CSPs provide digital signatures for authenticating messages. For more information, see *authentication*.

dead letter queue

A queue in which MSMQ stores undeliverable messages. Each computer has a dead letter queue.

destination queue

Messages are sent from source computers to destination queues. As they traverse your MSMQ network, the messages may pass through intermediary store-and-forward servers.

encryption

The process of coding and decoding messages, ensuring they cannot be read or used by anyone not authorized to read or use the message. MSMQ supports encryption through the use of public and private keys.

The MSMQ public key implementation is based on the Microsoft CryptoAPI, and uses the Microsoft Base Cryptographic Provider version 1.0.

enterprise network

A collection of sites connected through slow and expensive links.

express message

MSMQ supports two delivery methods: express and recoverable. Choosing between express and recoverable delivery is a matter of trading performance and resource use for reliability and failure recovery. In general, express messages use fewer resources and are faster than recoverable messages. However, express messages cannot be recovered if the computer storing the memory-mapped message files fails. Recoverable messages use more resources and are slower than express messages, but can be recovered no matter which computer fails.

foreign computer

A computer that does not run MSMQ, but can exchange messages with MSMQ through an MSMQ connector server.

foreign CN

A CN that contains foreign computers (computers that do not run MSMQ) and an MSMQ connector server (a computer that handles messaging between your MSMQ enterprise and other message queuing systems).

format name

A representation of the queue used by the MSMQ API. Format names are used to specify a queue when making calls to several API functions.

GUID

A globally unique identifier (GUID) is a 128-bit (16-byte) integer that an algorithm creates. The algorithm uses several criteria, including the current date, time, and a computer identifier, to ensure that it is unique. The GUID format is {12345678-1234-1234-1234-123456789012}.

MSMQ uses GUIDs to represent the ID of many objects, including computers, queues, and messages.

hash algorithm

The hashing algorithm used in encrypting messages.

hop

A unit of measurement used in intra-site routing. Each MSMQ routing server that a message must pass through adds one to the number of hops the message travels. If the message travels directly between two computers that establish a session, the message travels only one hop.

in routing server

An MSMQ routing server that provides session concentration by acting as a gateway for all incoming messages for one or more independent clients. MSMQ independent clients can be configured to use InRSs, OutRSs, or both.

information server

See *controller server*.

information store

See *MQIS*.

InRS

See *in routing server*.

inter-site routing

The process of routing messages between sites.

intra-site routing

The process of routing messages within a site.

ISV

Independent software vendor.

journaling

The process of logging or storing messages in a queue. See *journal queue*.

journal queue

The process of logging or storing a copy of a message in a queue. Journal messages are stored in a queue called Journal.

Two types of journaling are available: source and target. Source journaling is the process of storing a copy of an outgoing message. It is configured on a message basis, and is therefore a property set by the sending application. When journaling is enabled for a message, a copy of the message is placed in the source journal queue on the source computer when the message is queued. In MSMQ Explorer, the source journal queue is displayed under the computer.

Target journaling is the process of storing a copy of incoming messages. It is configured on a queue basis. When target journaling is enabled, a copy of each incoming message is placed in the target journal queue when the message is removed (read) from the target queue. A target journal queue (simply called Journal) is created for each queue when the queue is created. In MSMQ Explorer, target journal queues are displayed under each public queue.

journal quota

Specifies the cumulative limit for messages in the journal queue. The limit is based on cumulative message size. When a journal queue’s quota is reached, messages are no longer journaled.

message

A unit of information or data sent between computers. The message can contain text or binary data as defined by the sending application. Messages can be up to 4 MB in size.

message class

Indicates the message source and its use. A message class can be normal, a positive or negative acknowledgment message, or a report message. This property is set by MSMQ or an MSMQ connector application. All messages created by MSMQ-based applications are normal messages.

message label

A string defined by the sending application. The label can be viewed from MSMQ Explorer and is never encrypted.

Microsoft Message Queue Service

The MSMQ component that provides the core MSMQ functionality. This service runs on all MSMQ independent clients and servers.

MS DTC

Microsoft Distributed Transaction Coordinator (MS DTC) is a transaction manager that coordinates transactions that span multiple resource managers. Work can be committed as an atomic transaction even if the work spans multiple resource managers, potentially on separate computers.

MQIS

MSMQ Information Store. The MQIS is a SQL 6.5 replicated database. For information on the MQIS database and replication, see the *Microsoft Message Queue Server Administrator’s Guide*, Chapter 1, Understanding MSMQ.

MSMQ

Microsoft Message Queue Server.

MSMQ dependent client

MSMQ dependent clients function much like MSMQ independent clients, however they cannot function without synchronous access to an MSMQ server (PEC, PSC, BSC, or MSMQ routing server).

MSMQ dependent clients can be installed on computers running Windows 95, and Intel-compatible computers running Windows NT Workstation or Windows NT Server. (Alpha and PowerPC computers running Windows NT are not supported).

MSMQ dependent clients require synchronous access to the supporting MSMQ server to perform all standard MSMQ functions, such as creating queues, sending messages, and receiving messages.

MSMQ connector server

MSMQ connector servers allow MSMQ-based applications to communicate with computers that use other messaging systems, called *foreign computers*. MSMQ connector servers use *foreign CNs* and *connector queues* to communicate with foreign computers. The Level 8 Systems MSMQ message queuing product is an example of an MSMQ connector server.

To develop an MSMQ connector server, you must obtain the MSMQ Connector Software Development Kit (SDK).

MSMQ RAS connectivity service

The MSMQ RAS connectivity service allows MSMQ independent clients to connect to your MSMQ network through a Windows NT Server remote access service (RAS) server. To reduce long distance charges and ensure independent clients always connect to the same site, you should configure your Windows NT RAS servers with the MSMQ RAS connectivity service in each site. While it is possible to configure MSMQ independent clients to connect to your MSMQ network through a RAS server in any site, this configuration does not provide the same benefits.

MSMQ routing server

Supports dynamic routing and intermediate store-and-forward message queuing. MSMQ routing servers allow computers that use different protocols to communicate. If configured to do so, MSMQ routing servers provide session concentration. Unlike BSCs, MSMQ routing servers do not hold a read-only replica of the PSC or PEC database.

MSMQ service

See *Microsoft Message Queue Service*.

MSMQ server

MSMQ uses four server types to control message queuing:

- Primary enterprise controller (PEC)
- Primary site controller (PSC)
- Backup site controller (BSC)
- MSMQ routing server

Any server can also function as a connector server.

MSMQ independent client

MSMQ independent client software can be installed on computers running Windows 95, Windows NT Workstation version 4.0 or later, or Windows NT Server version 4.0 or later. MSMQ independent clients can create and modify queues, and send and receive messages, just as MSMQ servers can. MSMQ independent clients can create queues and store messages on the local computer, without synchronous access to an MSMQ server. The primary difference between MSMQ independent clients and MSMQ servers is that independent clients do not have the intermediate store-and-forward capability of MSMQ servers, nor do they store information from the distributed MSMQ database.

In addition to the basic MSMQ files, you can install the MSMQ Software Development Kit (SDK) on MSMQ independent clients.

You can also install the MSMQ Explorer on MSMQ independent clients running under Windows NT Workstation or Server. You can use the MSMQ Explorer to administer your MSMQ enterprise remotely from computers running Windows NT Workstation.

non-repudiation

MSMQ authentication provides non-repudiation (because no user can sign a message with another user’s identity, no user can refute that he or she sent a message if it contains his or her signature).

out routing server

An MSMQ routing server that provides session concentration by acting as a gateway for all outgoing messages for one or more independent clients. MSMQ independent clients can be configured to use InRSs, OutRSs, or both.

OutRS

See *out routing server*.

PEC

See *primary enterprise controller*.

permission

A rule associated with an object (enterprise, site, CN, computer, or queue) to regulate which users can access the object and in what manner. Permissions apply to specific objects and are different from rights, which apply to the system as a whole.

primary enterprise controller

The primary enterprise controller (PEC) contains a master copy of the enterprise and site settings in the MQIS database and certification keys (used in authenticating messages). Administrators install one PEC on an MSMQ network. The PEC functions as a PSC for one site and also functions as an MSMQ routing server.

For information on the MQIS database and replication, see the *Microsoft Message Queue Server Administrator's Guide*, Chapter 1, Understanding MSMQ.

primary site controller

You install one primary site controller (PSC) for each additional site in your MSMQ network—the PEC functions as the site controller for the initial site you create. The PSC stores information about the site’s computers and queues in the MQIS, and also functions as an MSMQ routing server.

For information on the MQIS database and replication, see the *Microsoft Message Queue Server Administrator’s Guide*, Chapter 1, Understanding MSMQ.

privacy level

Specifies whether the queue accepts encrypted messages, unencrypted messages, or both. If the privacy level is None, the queue accepts only unencrypted messages. If the privacy level is Optional, the queue accepts both unencrypted and encrypted messages. If the privacy level is Body, the queue accepts only encrypted messages.

private queues

Private queues are not published in the MQIS, and therefore don’t add to the MQIS replication load. Private queues can be accessed only by applications that have access to the full format name of the queue. In the default view, MSMQ Explorer does not display private queues.

PSC

See *primary site controller*.

public queues

Published in the MQIS. All public queues are replicated throughout the enterprise, and can therefore be located by any computer within the enterprise.

QM

See *queue manager*.

queue manager

A component of the Microsoft Message Queue Service.

queue name

User friendly names created on the computer on which the queue resides.

queue pathname

Queues are referenced by the queue pathname. The queue pathname is a combination of the computer name where the queue resides and the queue name. For example, if you create a queue called Myqueue on a computer called Joeuser1, the queue pathname is Joeuser1\Myqueue.

queue quota

Specifies the cumulative limit for messages in public queues. The queue quota limits are based on size and can be set independently of the computer quota. When a queue quota is reached, messages can no longer be sent to the queue until one or more messages are removed from the queue and enough space is available for the message.

queue type

A GUID specified by the application that created the queue. Consistent use of queue type IDs can help MSMQ-based applications locate different types of queues. For example, if your application receives messages from a MAPI application, it might search for an input queue with a specific MAPI type-ID. Your MSMQ-based applications should use a different GUID for each queue type used within your enterprise.

recoverable message

MSMQ supports two delivery methods: express and recoverable. Choosing between express and recoverable delivery is a matter of trading performance and resource use for reliability and failure recovery. In general, express messages use fewer resources and are faster than recoverable messages. However, express messages cannot be recovered if the computer storing the memory-mapped message files fails. Recoverable messages use more resources and are slower than express messages, but can be recovered no matter which computer fails.

report queue

Contains MSMQ-generated report messages that track the route of your application messages as they move toward their target queues. A report message is generated each time a message passes through an MSMQ routing server. Report queues can be used when sending test messages and when tracking message routes for a specific application.

resource manager

A system service that manages durable data. Server applications use resource managers to maintain the durable state of the application, such as the record of inventory on hand, pending orders, and accounts receivable. The resource managers work in cooperation with the transaction manager to provide the application with a guarantee of atomicity and isolation (using the two-phase commit protocol).

For example, both MSMQ and Microsoft SQL Server are resource managers.

response queue

A message property that is set by the sending application and is used by the receiving application for replying to messages. For example, an application might send a response message to a response queue every time the application receives a message.

right

Authorizes a user to perform certain actions on the system. Rights apply to the system as a whole and are different from permissions, which apply to specific objects.

routing

MSMQ establishes a direct connection (a session) using the underlying protocol if possible. When a direct connection is not possible or not allowed MSMQ uses its own routing system. MSMQ routing occurs when one or more of the following conditions exist.

- A session cannot be established between the sender and the receiver (for example, the source and target computers do not share a common CN or the target computer is offline)
- In Routing Servers (InRSs) or Out Routing Servers (OutRSs) are defined for the sender or receiver
- Messages must travel between two sites, and one or both sites have a site gate defined

MSMQ servers make two assumptions about your MSMQ network: *Intra-site routing* is fast and inexpensive, while *inter-site routing* is slow and expensive.

RSA

The RSA public-key cipher was developed by (and named after) Ron Rivest, Adi Shamir, and Leonard Adleman in the late 1970's. This algorithm is very well known; you can read about its internal details in any book on cryptography.

RSA is used by many CSPs to encrypt/decrypt keys and to generate/verify digital signatures. This algorithm is used when operations are performed using either the key exchange or digital signature key pair. When using CryptoAPI, this algorithm cannot be used to encrypt bulk data.

RSA is a variable-key-length cipher. However, when using CryptoAPI with the Microsoft RSA Base Provider, the key length is hard-coded to 512 bits.

site pathname

The name of the site.

SDK

Acronym for Software Development Kit. The MSMQ SDK and related documentation are installed when you click **Development Workstation** or **Development Server** during Setup.

sender ID

Displays the security identifier (SID) of the user who was logged on at the source computer.

session

The time during which two computers maintain a connection and, usually, transfer information.

session concentration

Used to reduce sessions within a site and/or reduce sessions between sites. Unnecessary sessions can increase connection and bandwidth costs. MSMQ session concentration reduces these costs by funneling connections through MSMQ servers.

MSMQ supports two types of session concentration: intra-site and inter-site. Intra-site session concentration typically reduces network bandwidth use within a site. Inter-site session concentration typically reduces the number of sessions between sites.

By loading specific servers with more independent clients, you can manually load or tune your MSMQ network using session concentration. For example, all MSMQ independent clients in one department can be configured to send and receive all messages through a specific server or set of servers.

site

A physical collection of computers where communication between any two computers is fast and inexpensive. Sites usually parallel the physical location of the computers, however, every computer in a site does not necessarily have to be running the same protocol. Computers in the same site may not be able to directly communicate with each other.

site controller server

Refers to PECs and PSCs, usually in the context of installing MSMQ. *Controller server* refers to the enterprise PEC, all PSC, and all BSCs.

site gate

You can establish site gates to provide inter-site session concentration. If a site is configured to use a site gate, every MSMQ message sent between computers in different sites must be routed through the site gate. The routing topology beyond the source site is transparent to the computers within the site, with the exception of the site gate. Thus, the routing decisions of the computers in the source site are simplified. By default, sites do not use site gates.

The following are the requirements for site gates.

- The computer must belong to a site to be a site gate for the site
- The computer must be able to connect to each neighboring site

Because sites configured with site gates are dependent on the site gate for all inter-site transmission of MSMQ messages, it is preferable to assign more than one site gate to a site to provide load balancing and failure recovery.

site link

MSMQ measures inter-site routing based on costs assigned to links. You can assign link costs based on the delay of the link or the communications costs for the line (for example a dial-up line). If you have only two sites, choose any value above zero. If you have three or more sites, and the cost of routing between sites is more or less equal, use the same value for each site link. However, if you have three or more sites, and the cost of routing between sites is not equal, use site link costs to define the difference in the routing costs. A site link cost of zero indicates that the two sites are not connected. Site link costs can range from 1 to 999,999.

If a site is configured to use a site gate, the site gate must belong to the same CN as a server in the other site. If the other site also uses a site gate, the site gate must belong to the same CN as the site gate in the other site

site link cost

MSMQ calculates the cost of inter-site routing based on relative numbers that administrators assign to site links. These numbers, called *site link costs*, represent the cost of communication of that link. Site link costs can be between 0 and 999,999. A site link cost of 0 indicates that two sites are not connected.

source journaling

The process of storing a copy of an outgoing message. Source journaling is configured on a message basis, and is therefore a property set by the sending application. When source journaling is enabled for a message, a copy of the message is put in the source journal queue of the sending computer when the message arrives at the target queue. In MSMQ Explorer, the source journal queue is displayed under the computer.

source computer

The computer that sent the message.

supporting server

A PEC, PSC, BSC, or MSMQ routing server that supports one or more MSMQ dependent clients.

target journaling

The process of storing a copy of incoming messages. Target journaling is configured on a queue basis. When target journaling is enabled, a copy of each incoming message is placed in the target journal queue when the message is removed (read) from the target queue. A target journal queue (simply called Journal) is created for each queue when the queue is created. In MSMQ Explorer, target journal queues are displayed under each public queue.

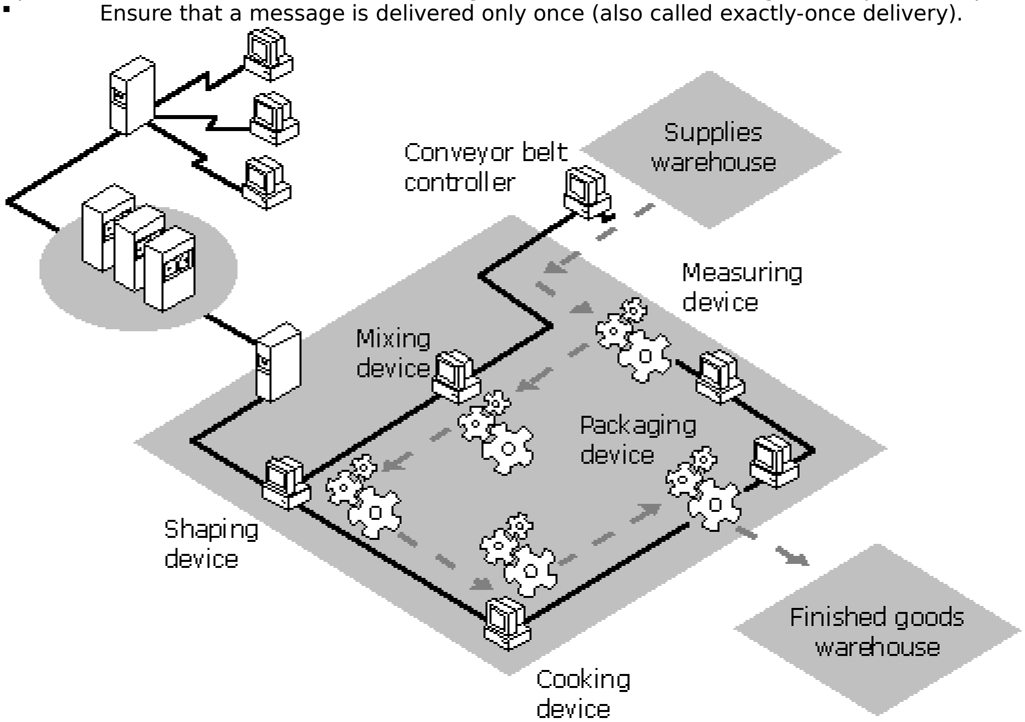
transaction

A transaction is a unit of work that is carried out as an atomic operation--that is, the operation succeeds or fails as a whole. The unit of work is usually the pairing of two or more operations. The transaction ensures that the MSMQ message operations either succeed or fail in conjunction with other OLE-transaction-compliant operations or other MSMQ operations. For example, an MSMQ-based application might send a message and update an SQL Server database in the same transaction. MS DTC ensures that both actions succeed, or neither are executed.

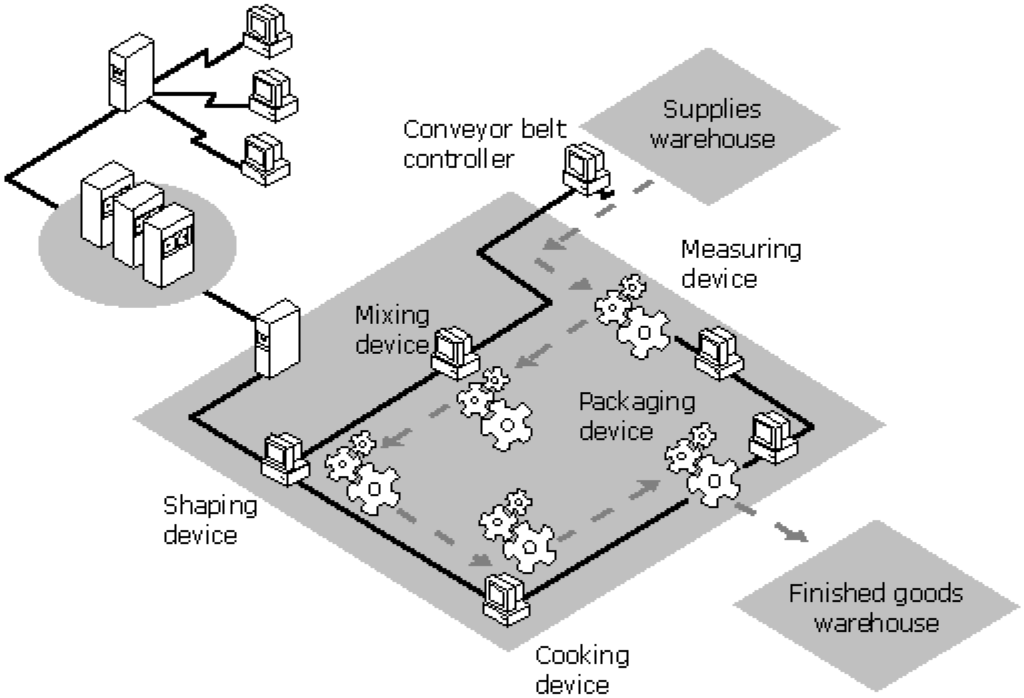
MSMQ can be used as a resource manager under the control of Microsoft Distributed Transaction Coordinator (MS DTC), and is therefore compatible with any XA-compliant resource manager.

Using the MSMQ transaction flag you can.

- Transact the sending or receiving of any message with any other transactional resource (for example, update an SQL database and send a message) and transact the sending or receipt of multiple messages.
- Ensure that a message is delivered only once (also called exactly-once delivery).



Ensure all messages sent from one computer to another are delivered in order (also called in order delivery).



Use positive or negative acknowledgments (ACKs and NACKs) to confirm messages reached or were successfully retrieved from the destination queue (also called end-to-end confirmation).

transaction dead letter queue

A transactional message that cannot reach the destination application is stored in the transaction dead letter queue on the source computer. Transaction dead letter queues are created for each independent client and server on your MSMQ network when MSMQ is installed on that computer. These queues are displayed as **Xact Dead Letter** under the computer in MSMQ Explorer.

type ID

See *queue type*.

XA-compliant

An XA-compliant resource manager uses the X/Open Distributed Transaction Processing XA interface to communicate with a transaction manager. XA-compliant transaction managers coordinate distributed transactions between XA-compliant resource managers. MS DTC is both an XA-compliant resource manager and an OLE transactions-compliant transaction manager. Because MSMQ is an OLE transactions-compliant resource manager, it can be XA-compliant through MS DTC.