

In Windows XP, the next version of MSMQ, Message Queuing 3.0, introduces several exciting new features designed to extend the application developer's set of tools to support new types of messaging-based applications.

Introduction

Message queue-based communication using Message Queuing (MSMQ) in the Windows operating system family already offers a rich feature set for applications. These include, but are not limited to, extremely fast inter-application communication, message delivery guarantees, sophisticated message and queue security, queue location transparency, rich programming language support for rapid application development, and Active Directory integration.

Message Queuing is available on all Microsoft platforms from Windows CE (3.0) to the Windows 2000 Server family and Windows XP Professional.

In Windows XP, the next version of MSMQ, Message Queuing 3.0, introduces several exciting new features that are designed to expand the application developer's set of tools to support new types of messaging-based applications. In particular, the two new areas that MSMQ 3.0 focuses on are messaging over the Internet and one-to-many messaging.

This white paper will help developers understand what the new features in MSMQ 3.0 are and how they can be used to extend existing applications or to build new kinds of applications.

Summary Of New Features

MSMQ 3.0 in Windows XP is a major feature release, in which new areas of functionality are introduced and current functionality is significantly enhanced in other existing areas.

Specifically, these include the following areas:

- ❖ Internet messaging
- ❖ One-to-many messaging model
- ❖ Message Queuing Triggers
- ❖ Management, deployment, and programming model enhancements

Internet Messaging

MSMQ 3.0 in Windows XP provides out-of-the-box support for Internet-deployed applications and for the B2B market. More specifically, MSMQ 3.0 adds the following enhancements:

- ❖ HTTP as an optional transport protocol
- ❖ An XML-based SOAP extension that defines a reliable end-to-end messaging protocol.
- ❖ Support for load balancing, web farms, and firewalls (in the Windows XP Server release).

HTTP Transport

MSMQ 3.0 lets you specify HTTP as the delivery protocol for messages. Natural extensions to the programming model allow the application developer to indicate which underlying protocol will be used to deliver messages. Of course, the proprietary, TCP-based, MSMQ protocol is still supported.

Sending applications can control the various options associated with HTTP, such as the HTTP proxy server to be used or the use of HTTPS for secured communication.

On the receiving end, MSMQ 3.0 is closely integrated with the Internet Information Server (IIS), which is also available on Windows XP. Applications can map specific URLs to MSMQ queues, and every message sent to that URL is automatically inserted into the applicable queue by MSMQ. An MSMQ application can then use standard MSMQ APIs to obtain those messages.

SOAP Extensions for Reliable Messaging

MSMQ 3.0 defines and implements an XML-based SOAP extension. This protocol enables message exchange with support for several qualities of services and kinds of message notifications.

This SOAP-based protocol supports the following features:

- ❖ Express messages
- ❖ End-to-end reliable messages
- ❖ End-to-end authentication
- ❖ Notifications of message delivery
- ❖ Message authentication and digital signatures.

From a programming point of view, every message sent via HTTP will automatically be formatted as a SOAP-based message. New properties, specific to SOAP, are available to developers for controlling SOAP formatting at a fine level. For instance, it is possible for a message sender to insert non-MSMQ headers into the SOAP envelope and have them retrieved and processed by the message receiver.

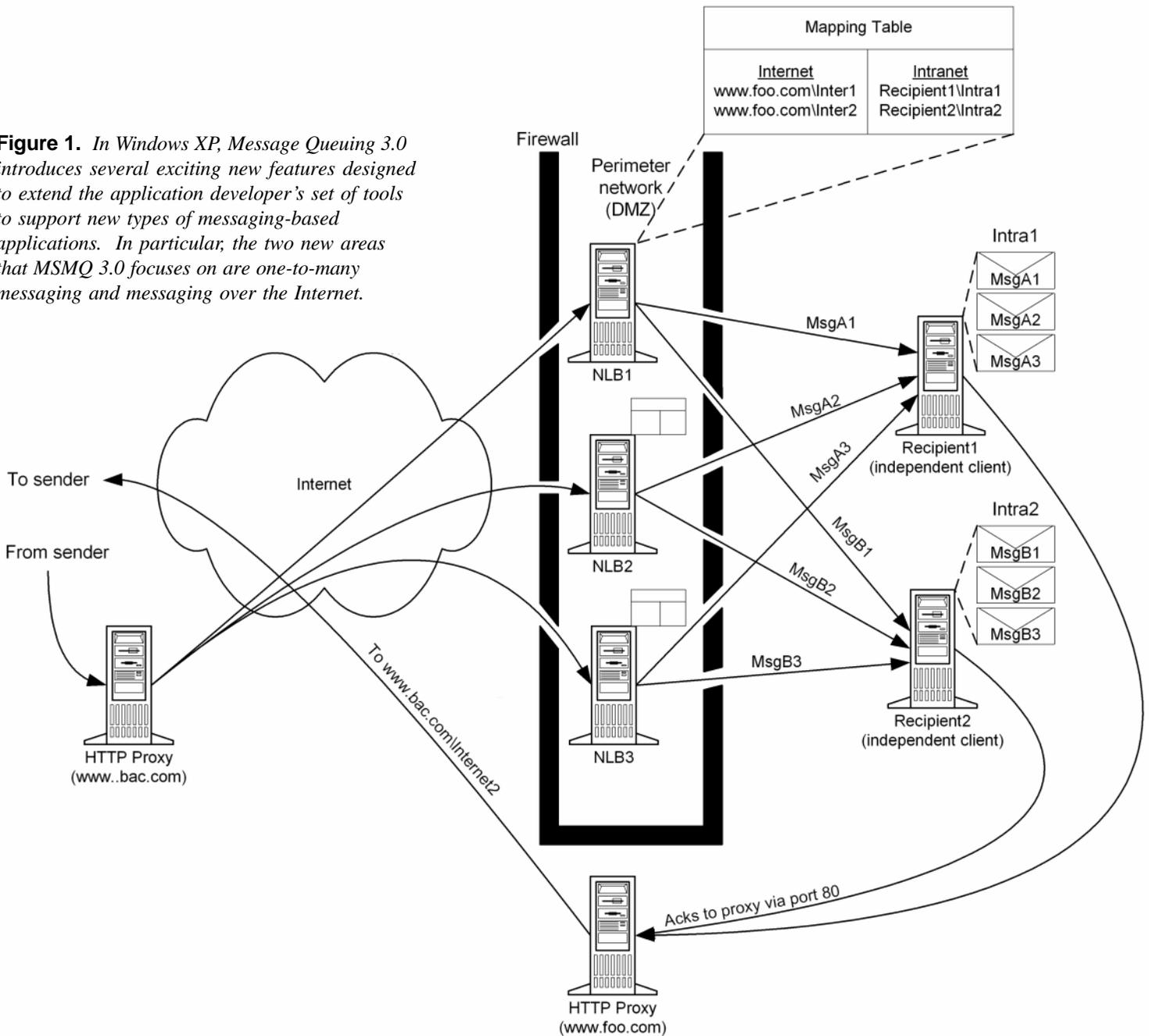
Firewall and Web-Farm Support

The use of HTTP as a transport enables MSMQ to leverage existing firewall support, without the need for MSMQ-specific firewall configurations. Wherever HTTP is available, MSMQ messages will get through.

In order to increase web server availability for scale, customers usually deploy a front-end web farm. A web farm is a collection of independent share-nothing servers, to which all HTTP incoming requests are directed. NLB (Microsoft Network Load Balancing), Cisco LocalDirector, NAT, and DNS round robin are some of the methods typically used to choose the server that will serve a particular HTTP incoming request. In such an environment, because two successive HTTP connections from the same source might be directed to two different servers, enabling end-to-end reliable messaging, where state has to be persisted, becomes complex.

Consider the following scenario: a sending MSMQ application sends two messages that must arrive in order at a specific URL (which is equivalent to an MSMQ queue). The first message sent in the first HTTP session might be directed to server A in the web farm, but the second message

Figure 1. In Windows XP, Message Queuing 3.0 introduces several exciting new features designed to extend the application developer's set of tools to support new types of messaging-based applications. In particular, the two new areas that MSMQ 3.0 focuses on are one-to-many messaging and messaging over the Internet.



might be sent in a different HTTP session- and might be directed to server B. In that case, the MSMQ service on server B will process the message but reject it because it does not know that the first message was delivered to server A.

To address this issue, MSMQ 3.0 introduces a new concept: the MSMQ "reverse message proxy" server. Reverse proxy functionality will usually be enabled on all the servers in the web farm. Each "reverse message proxy" in the web farm will maintain a mapping between the "external URL"

and "internal URL" of a queue. For every incoming message arriving at a web farm computer, MSMQ 3.0 will read the "external URL" address to which it was sent. If there is an entry in the map for that external URL, MSMQ 3.0 will forward the message to the "internal URL" associated with it. The "message proxy" itself is not considered the final destination of the message: in fact, the end-to-end reliable messaging protocol is not implemented at the "message proxy" level, but rather at the final destination of the message.

The "message proxy" also plays an important security role. It is positioned in the front-end of the enterprise, where all incoming HTTP requests are directed. If so configured, the "message proxy" will forward messages further into the enterprise as described above, but only those messages that meet certain security requirements. For instance, only messages arriving via an HTTPS session, or messages whose sender can be authenticated, or messages selected by another security criterion will be forwarded. This

ensures that only authorized messages are ever presented to corporate servers.

One-to-many Messaging Model

MSMQ 3.0 in Windows XP extends the regular MSMQ programming model from one-to-one to one-to-many. The traditional MSMQ model allows a single message to be sent to a single queue. However, the one-to-many messaging model enables clients to send the same message to multiple recipient queues. MSMQ 3.0 offers several ways to implement this model

- ❖ Real-time messaging multicast
- ❖ Distribution lists and multiple-element format names

MSMQ 3.0's one-to-many mechanisms provide the application developer with the fundamental tools required to build sophisticated publish/subscribe systems. These kinds of systems are useful for addressing reliable information dissemination scenarios.

Real-Time Messaging Multicast

MSMQ 3.0 provides a mechanism to send messages on top of IP multicast.

Specifically, MSMQ 3.0 uses the Pragmatic General Multicast (PGM) protocol (see <http://www.ietf.org/internet-drafts/draft-speakman-pgm-spec-06.txt>), which is becoming the de-facto enterprise standard for obtaining a degree of reliability on top of IP multicast.

In MSMQ 3.0, when an application sends a "real-time multicast" message, a single copy of the message is sent on the network, but this message can then be received and processed by numerous destination applications. This method is extremely efficient if the same message has to be delivered to a huge number (scale up to the millions) of receiving applications.

"Real-time multicast" messaging supports only an "at-most-once" quality of service. In particular, it does not support transactional sending. To send a "real-time multicast" message, the existing Send API has been extended so that the sender can specify a target IP multicast address. The receiver's programming model is not affected at all. An IP multicast address property can be attached to a queue, and subsequently every message

sent to that multicast address will be inserted into the queue by MSMQ 3.0. Applications continue to read messages from the queue using standard MSMQ APIs and methods.

From a security point of view, standard MSMQ authorization and authentication continue to be available, but MSMQ 3.0 does not provide specific encryption or decryption for real-time multicast message.

Distribution Lists and Multiple-Element Format Names

MSMQ 3.0 extends the sender's programming model to allow a sending application to send a single message to a list of destinations queues. That list can be constructed dynamically in the form of a multiple-element format name or can be published and kept in the Active Directory in the form of a distribution list.

Sending a message to a list of queues uses more network resource than "real-time message" multicast, since multiple messages are sent on the network. However, all qualities of service, including transactional sending and exactly-once-delivery are available.

Full security support, including authentication, authorization, and encryption is available.

Message Queuing Triggers

MSMQ 3.0 in Windows XP integrates a new trigger service that greatly simplifies the process of automatically invoking applications based on message arrival.

Message Queuing Triggers is a service that allows you to associate the arrival of incoming messages at a queue with functionality in a COM component or a stand-alone executable program. You can use Message Queuing Triggers to define business rules that can be invoked in response to such messages without any additional programming. Ordinarily, to provide such message-handling functionality, application developers must create a receiving application that invokes specific behavior on a per-message basis. By using Message Queuing Triggers, specific behavior can be invoked at the queue level. Thus, application developers no longer need to re-implement generic infrastructure code to provide such message-handling functionality.

Message Queuing Triggers is an integral part of MSMQ 3.0. Both an integrated installation procedure and integrated management functionality are provided.

Message Queuing Triggers Concepts

Message Queuing Triggers is defined in terms of a set of triggers.

In turn, each trigger is associated with a specific monitored queue on the local computer and is invoked for every message arriving at that queue. A trigger is associated with one or more rules. A rule, in turn, is defined in terms of one or more conditions and an action in the form of an executable file and an argument list or in the form of a COM component, a method, and an argument list. The actual arguments can be bound to specific message properties. The action is executed when all the conditions associated with the rule hold.

Management Deployment and Programming Enhancements

MSMQ 3.0 in Windows XP introduces significant infrastructure changes to ease deployment and installation and relaxes some existing limitations.

Easier Deployment

MSMQ 3.0 client installation no longer requires prior deployment of an MSMQ server on an Active Directory server.

Enhanced Functionality

MSMQ 3.0 message storage is now limited only by disk space. In particular, MSMQ 3.0 has relaxed the 2-Gb storage limit per machine. The theoretical capacity limit for persistent MSMQ 3.0 messages is now 1Tb (terabyte).

Likewise, MSMQ 3.0 introduces significant performance improvements that allow the MSMQ 3.0 service to quickly become available after system restart even in the presence of large numbers of persistent messages.

Programming Model Enhancements

MSMQ 3.0 exposes numerous enhancements to its programming model. Based on customer feedback and requests, vari-

ous improvements have been made that enrich the MSMQ developer's development environment.

- ❖ Queue browsing - a set of new functions have been added that enable the developer to easily traverse a queue bi-directionally and to retrieve a message based on its unique identification in O(1) time.
- ❖ Easy MSMQ service administration - a set of functions and methods have been added that enable MSMQ service monitoring and control in various scenarios.

Deprecated And Unsupported Features

Several existing features are being deprecated or removed in MSMQ 3.0. In particular features which are no longer relevant, useful or have been superseded.

- ❖ The MSMQ Exchange connector is no longer supported in Windows XP.
 - Of course, it continues to be available for previous versions of Windows (Windows NT 4.0 and Windows 2000).
- ❖ The IPX protocol is no longer supported by MSMQ 3.0 in Windows XP.
- ❖ The MSMQ service is not available in the Windows XP Personal version.
 - You can run DCOM-based applications on Windows XP Personal in order to access MSMQ.
- ❖ The MSMQ3.0 dependent client only supports MSMQ 2.0 level functionality.
 - This means that new MSMQ3.0 features like distribution lists are not available to MSMQ 3.0 dependent clients.

- The alternative is to deploy DCOM-based solutions instead.

Summary

As we can see, MSMQ 3.0 is a major feature release that extends traditional MSMQ messaging in two significant areas. MSMQ programmers can now leverage the vast reach of the Internet by using MSMQ 3.0's HTTP transport. Additionally, MSMQ programmers can easily build sophisticated and complex many-to-many messaging systems with the new multicast and distribution list abstractions.



CloverLink Systems, Inc. offers a full spectrum of Internet application development services. We specialize in analysis, design and development of Internet and Intranet based solutions, using leading-edge technologies backed by an experienced professional staff and a sound business strategy. We are dedicated to bringing your products and services to market on time and on budget!

Let us help you achieve your company's unique goals and vision by providing a development proposal to fit your needs.

Call today and put the CloverLink team to work for you.



Corporate Offices

CloverLink Systems, Inc.
1486 Sunshine Drive
Glendale, CA 91208

e-mail: Sales@CloverLink.com
<http://www.CloverLink.com>

800.378.8348