WebSphere® software

IBM WebSphere MQ, Version 7.0

Highlights

- ***Enhanced ease of use for publish-and-subscribe and JMS messaging***
- ***Enhanced usability with graphical configuration of publish-and-subscribe and JMS messaging through Eclipse-based WebSphere MQ Explorer***
- ***Enhanced publish-and-subscribe performance, increasing throughput by up to 20 percent***
- ***Enhanced JMS performance, increasing selectors by up to 250 percent and listener throughput by up to 45 percent with improved latency***
- ***Extended verbs and behaviors for MQI programming interface, improving developer productivity***
- ***Enhanced WebSphere MQ clients, increasing nonpersistent throughput up to 300 percent and increasing resilience and availability***
- ***Web 2.0 support to help create a richer user experience by bridging HTTP applications with AJAX and REST to the WebSphere MQ messaging backbone***

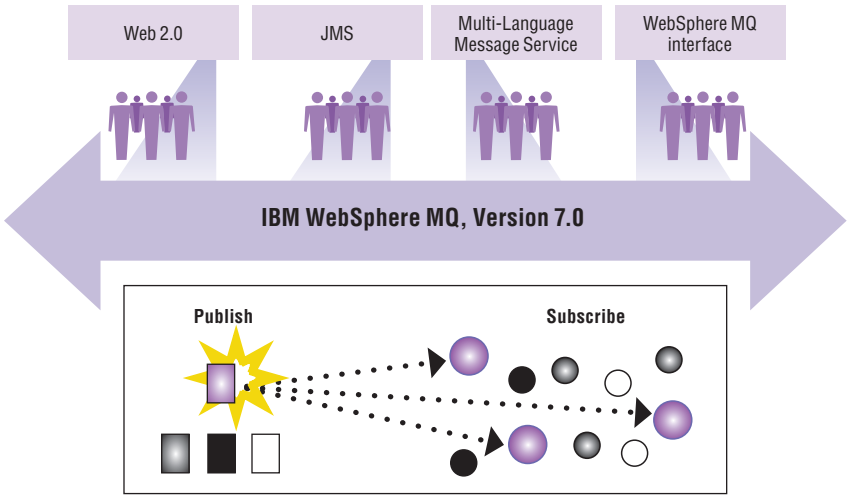

Figure 1. WebSphere MQ, Version 7.0 delivers enhanced ease of use and choice for developers with integrated publish-and-subscribe messaging.

IT departments today are challenged with keeping pace with ever-changing business demands. Today's IT infrastructure needs to be flexible to enable a rapid response to opportunities and pressures. A mixture of computing platforms and operating systems interconnected across networks can form a fragile, tangled web where IT resources become locked in maintenance mode, wrestling with network complexities and propagating changes across the mix of applications.

IBM WebSphere® MQ provides a messaging backbone that enables virtually any commercial IT system to be connected with a reliable, flexible transport. WebSphere MQ provides a messaging backbone for applications, Web services and Web 2.0 (see Figure 1). It enables you to connect the new with the now, so you can leverage new technologies to unlock the true potential of business data and applications residing in the core systems that run your business. It helps to ensure delivery so that data is not lost in transit and, as a result, helps to preserve the integrity of your IT systems. WebSphere MQ handles the complexities of communication protocols and dynamically distributes messaging workload across available resources.

Enhanced WebSphere MQ Explorer

WebSphere MQ Explorer enables remote configuration of WebSphere MQ from Linux® x86 and Microsoft® Windows® workstations. It does not require a local server or client, and it can be installed on workstations without charge.

WebSphere MQ Explorer Explorer now helps simplify management of multiple queue managers by enabling these to be grouped into separate views in the navigator. Test and production queue managers, for example, could be grouped and viewed separately. It is also easier to configure security settings. Channel exits and user IDs and passwords can be configured for each queue manager or globally for all queue managers in a group or workspace.

WebSphere MQ, Version 7.0 incorporates and extends WebSphere MQ Explorer plug-ins that were previously available in Fix Pack 6.0.2.0. These include graphical problem-discovery tools that provide one-click detection and diagnosis of messaging backbone configuration and support for graphical configuration of user permissions through WebSphere MQ Object Authority Manager (OAM).

Enhanced ease of use for publish-and-subscribe messaging

Event-driven service oriented architectures (SOAs) provide a responsive, flexible infrastructure that can enable quicker and easier changes to how applications are connected. WebSphere MQ, Version 7.0 provides the ideal transport layer for an event-driven SOA with integrated support for publish-and-subscribe messaging. The publish-and-subscribe capability provides a flexible, event-driven pattern for loosely coupling applications. It enables messages to be sent between applications without any prior knowledge of which applications need to receive those messages. The linkage between applications that publish and receive messages is not explicitly defined, so it does not need to be altered when changes occur to the applications that send and receive data. The pathway between these applications is determined dynamically by WebSphere MQ through the topics or keywords used to declare interest in a set of messages when subscribing or for tagging messages when publishing.

WebSphere MQ, Version 7.0 makes it easier than ever to use publish-and-subscribe services to increase the flexibility of messaging solutions. Publish-and-subscribe messaging is now fully integrated into the graphical WebSphere MQ Explorer tool, making it easier to use and configure (see Figure 2). This release integrates publish-and-subscribe services into the queue manager, eliminating the need to send publication requests to publication queues. With WebSphere MQ, Version 7.0, you can now publish and subscribe to topics directly in your applications. There is also no need to start a separate publish-and-subscribe component. Instead, publish-and-subscribe services are now automatically enabled as part of all queue managers.

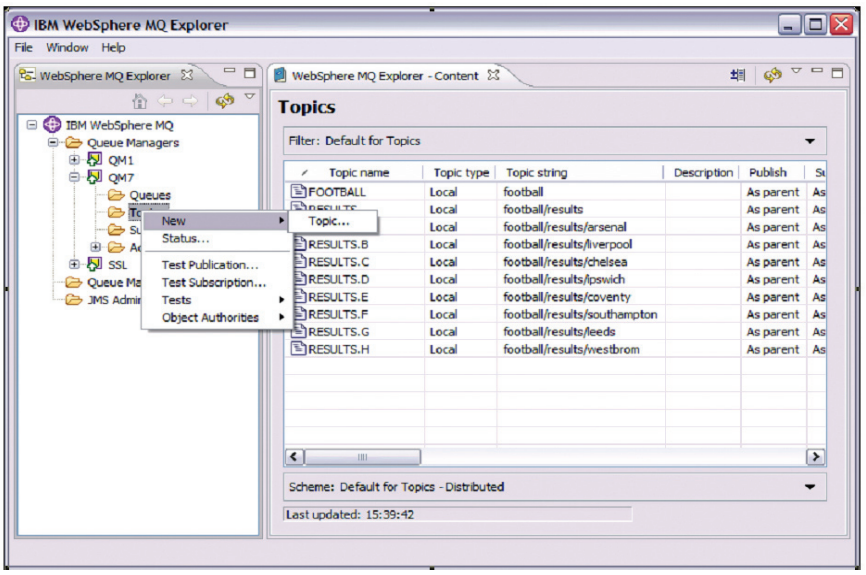

Figure 2. Publish-and-subscribe topics can be configured graphically using the Eclipse-based WebSphere MQ Explorer.

Configuration of publish-and-subscribe messaging is now fully integrated into the WebSphere MQ Explorer graphical tool. Topics can now be administered directly as first-class WebSphere MQ Explorer objects just like queues, simplifying administration and security management. Topics can be created using graphical wizards that can also generate corresponding Java™ Message Service (JMS) topics. Testing publish-and-subscribe messaging is now even easier, with built-in tools to send and receive test publications. Specific documented samples for publish-and-subscribe and point-to-point messaging are included.

With WebSphere MQ, Version 7.0, it is easier to see which applications are subscribed to topics in the same way that you can see which applications are using particular queues. Existing applications can be switched from using a point-to-point message pattern to using a publish-and-subscribe pattern without any code changes required. Administrators can create subscriptions on behalf of applications and redefine queues to have a topic assigned to them. Extensions in the Message Queuing Interface (MQI) make it easier for applications to take full advantage of publish-and-subscribe messaging. WebSphere MQ, Version 7.0 also now offers publish-and-subscribe statistics, such as the number of messages published on a topic.

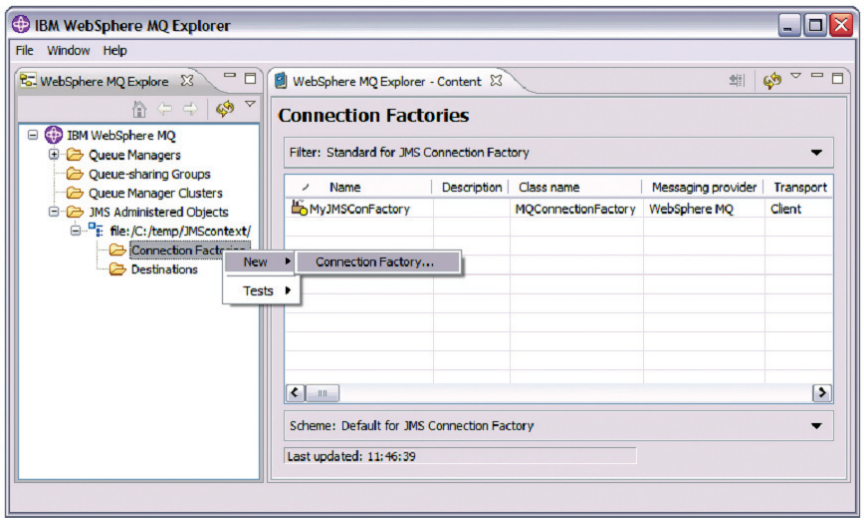

Figure 3. WebSphere MQ, Version 7.0 enables graphical configuration of JMS objects.

WebSphere MQ, Version 7.0 introduces a new *Topic object type* that can be used to define publish-and-subscribe topics and hierarchies of topics that enable a richer topic space. Topic objects are now automatically deleted after nondurable subscriptions, simplifying administration. Topic objects also support inheritance from parent topics, enabling topics to inherit attributes, such as security settings. This feature helps to reduce manual administration. Users do not need to define any topics in order to start using publish-and-subscribe messaging; default settings in the queue manager make it easy to get started.

WebSphere MQ, Version 7.0 helps ease management of publish-and-subscribe permissions. It restricts the use of topics by permission settings on the topic object following the existing WebSphere MQ Authorization Service security model.

Enhanced ease of use for JMS

JMS is an industry-standard Java programming interface for messaging products. WebSphere MQ, Version 7.0 delivers significant enhancements for JMS to improve ease of use and optimize performance. The previous release of WebSphere MQ already supported the latest iteration of the JMS standard, Version 1.1. So, even though the JMS interface remains the same in WebSphere MQ, Version 7.0, significant optimizations and enhancements have been introduced.

WebSphere MQ, Version 7.0 integrates JMS configuration into its graphical, Eclipse-based tool, WebSphere MQ Explorer, making it easier to design and deploy JMS solutions (see Figure 3). JMS objects, such as connection factories and destinations, are now displayed in WebSphere MQ Explorer along with WebSphere MQ objects, such as queues and channels. WebSphere MQ can remotely configure the entire WebSphere MQ network, it is now easier to explore and configure JMS messaging across the network.

Now WebSphere MQ Explorer makes it easier to see all the properties of JMS resources at a glance and update JMS object properties. It is easier to create JMS resources, such as connection factories, using step-by-step wizards. Creating a queue or topic can launch a wizard to help define a corresponding JMS destination at the same time. Developers can now customize views of JMS objects just as they can for other WebSphere MQ resources, such as filtering JMS objects that match selection criteria. Enhanced JMS trace facilities can help improve problem determination and serviceability. When problems are encountered that require the assistance of IBM support, improved diagnostic capabilities are provided to help resolve the problems rapidly. This includes First Failure Data Capture as well as enhanced trace control and formatting.

Enhanced publish-and-subscribe performance

WebSphere MQ, Version 7.0 optimizes the throughput of publish-and-subscribe messaging. This release integrates publish-and-subscribe services into the queue manager, removing the need to queue publications twice between the queue manager and the previously independent publish-and-subscribe component. By optimizing message logging for persistent publish-and-subscribe messaging, throughput can increase by up to 20 percent.[1] The resulting performance enhancement applies to all interfaces: JMS, MQI and Multi-Language Message Service (XMS).

WebSphere MQ, Version 7.0 enhances the scalability and availability of publish-and-subscribe messaging by providing publish-and-subscribe clusters that enable publications to be routed through a set of queue managers, eliminating single points of failure.

Enhanced JMS performance

WebSphere MQ, Version 7.0 delivers enhancements to optimize throughput and performance for JMS messaging.

JMS client applications can benefit from performance enhancements in WebSphere MQ, Version 7.0. Message read-ahead can increase nonpersistent JMS throughput by up to 300 percent.[1] JMS selector performance is also enhanced in this release by running selector matching on the server side, eliminating network latencies resulting from client-side selector matching. The performance of selectors depends heavily on the actual selector criteria, the header contents and the proportion of matching messages. With WebSphere MQ, Version 7.0, typical JMS selector scenarios can yield throughput increases of up to 250 percent.[1] WebSphere MQ, Version 7.0 optimizes JMS message listeners by eliminating polling and using asynchronous message delivery to monitor destinations. In this release, JMS message listener throughput can be increased by up to 45 percent with improvements in latency.[1]

Java Enterprise Edition (JEE) Application Servers can take advantage of these performance enhancements to improve message throughput for message-driven beans (MDBs) that are invoked whenever messages arrive through WebSphere MQ. By eliminating polling of the messaging provider, processor utilization of Application Servers can also be reduced.

Enhanced MQI

WebSphere MQ provides a rich programming interface, the Message Queuing Interface (MQI). The MQI interface is common across all supported platforms and provides a simple set of verbs that access the advanced features of WebSphere MQ. WebSphere MQ, Version 7.0 further enhances the MQI with new verbs and behaviors designed to increase ease of use for MQI developers.

WebSphere MQ, Version 7.0 introduces *message properties* to the MQI. This enables MQI users to customize message headers with user-defined data. New MQI verbs enable properties to be set (MQSETMP) and inquired (MQINQMP). This eliminates the need for applications to parse message headers to find message metadata. Message properties can be used to specify explicit relationships between messages, such as when messages are sent in reply to specific messages.

WebSphere MQ, Version 7.0 introduces a *callback* function to the MQI with a new verb: MQCB. This enables applications to register with the queue manager to be automatically notified whenever messages or publications arrive for their consumption. This eliminates the need for client applications to continuously poll queue managers and helps simplify administration, free up network bandwidth, reduce lag time from message arrival to delivery, and lower server- and client-processor utilization. Both MQI and JMS clients can benefit from callback. The JMS onMessage method has been reimplemented to take full advantage of callback, helping eliminate the need for internal polling and helping JMS clients achieve improvements in higher throughput and reduced latency.

WebSphere MQ, Version 7.0 enhances support for developers using the MQI for publish-and-subscribe messaging. A new MQI verb, MQSUB, enables applications to register subscriptions. Another new verb, MQSUBRQ, enables new subscribers to receive the most recent retained publication that was sent on a topic. Existing MQI verbs have new options for publish-and-subscribe messaging. MQOPEN can be used to access a topic, MQCLOSE to end durable subscriptions, and MQPUT and MQGET to publish and to receive subscriptions.

WebSphere MQ, Version 7.0 introduces support for *selectors* to the MQI, enabling applications to select messages from queues based on the values of message properties or message headers. Queries constructed in Standard Query Language (SQL92) can now be used to retrieve a filtered set of messages by matching the criteria in the message properties. The MQOPEN and MQSUB verbs can be used in conjunction with selectors to retrieve a series of messages matching the required criteria. Queries are now run within the queue manager to optimize performance for clients. Selectors eliminate the need for applications to browse through a queue, comparing these with their selection criteria and ignoring those that do not match.

Now, only messages matching the selection criteria are delivered to the application in the first place. This can improve performance and lower network bandwidth use by eliminating the need to send messages to clients that are not wanted and would merely be discarded or ignored. Filtering messages based on the content body — rather than just the header and properties — is enabled by IBM WebSphere Message Broker or IBM WebSphere Enterprise Service Bus.

Client enhancements

WebSphere MQ, Version 7.0 introduces a new quality of service to help optimize performance for client applications that require nonpersistent delivery of a stream of messages.

WebSphere MQ, Version 7.0 now enables servers to stream messages to clients so that messages can arrive buffered on the client machine, even before the client requests them. This *message read-ahead* function enables WebSphere MQ, Version 7.0 to preemptively dispatch messages that it expects clients to request. In addition, WebSphere MQ, Version 7.0 servers can regulate the flow of messages to clients.

Message read-ahead can significantly increase throughput of nonpersistent messaging by up to 300 percent.[1] Because messages on the client side are held in client memory and no longer queued, this feature applies to nonpersistent messaging scenarios only. If the WebSphere MQ server needs to send persistent messages to the client, WebSphere MQ automatically reverts back to its regular quality of service to handle those messages. Enabling WebSphere MQ, Version 7.0 clients for message read-ahead requires no changes to existing applications, only reconfiguration of WebSphere MQ. WebSphere MQ, Version 7.0 clients are also required for this function.

WebSphere MQ, Version 7.0 clients use full-duplex protocols for TCPI/IP, enabling more-effective heartbeating monitoring to increase availability by providing faster detection of connection failures and orphaned server-connection channels.

WebSphere MQ, Version 7.0 helps simplify administration of large numbers of client connections by providing the ability to share TCP/IP sockets. Connection sharing, or *multiplexing*, can make it easier to see the connection status of multiple clients, at a glance. By sharing sockets, multiplexed connections can also enhance scalability and reduce the time needed to establish connections, increasing overall throughput, especially for SSL connections.

WebSphere MQ, Version 7.0 introduces a new feature that can enable client applications to continue doing useful work after putting messages on queues. Rather than wait for response codes to be sent back to the client, *asynchronously putting messages* eliminates the need for clients to wait for responses back from the queue manager each time a message is put on a queue. Return codes can be requested later if required, using a new verb, MQSTAT, which provides the last asynchronous return code.

Now client applications can choose not to wait and can be preparing to send the next message or doing some other useful work instead of pausing to synchronize with a WebSphere MQ server. For applications that do not require response codes, this new feature can significantly increase performance.

Delivering value to Web 2.0

Web 2.0 promises a compelling new way of creating user interfaces using new Web technologies, such as Asynchronous JavaScript and XML (AJAX), and simple techniques, such as Representational State Transfer (REST). Compelling and valuable user interfaces require more than just an exciting design and cool widgets. Access to meaningful, up-to-date business data is key to delivering real value to Web 2.0 users.

WebSphere MQ, Version 7.0 makes the connection between real business data in core applications and newer Web 2.0 applications, unlocking the value of that data and making it easier to present it to Web 2.0 users (see Figure 4). WebSphere MQ, Version 7.0 delivers a *bridge for HTTP* (previously available as IBM SupportPac™ MA0Y) that links AJAX applications to the WebSphere MQ backbone using a RESTful programming model. Web 2.0 developers do not require WebSphere MQ knowledge or skills to connect their new applications to core business systems. Both point-to-point and publish-and-subscribe messaging is supported and accessed through uniform resource identifiers (URIs) that map to WebSphere MQ queues and topics. REST verbs GET, POST and DELETE are mapped to MQGET and MQPUT calls to queues or topics.

Sample applications help accelerate the development of Web 2. 0 solutions and help demonstrate the value of linking core business applications with Web 2.0. Because client applications do not require installation or configuration of WebSphere MQ client code, the bridge for HTTP can also be used where zero client footprint is preferred, and it can help simplify administration for large communities of applications that require simple access to WebSphere MQ.

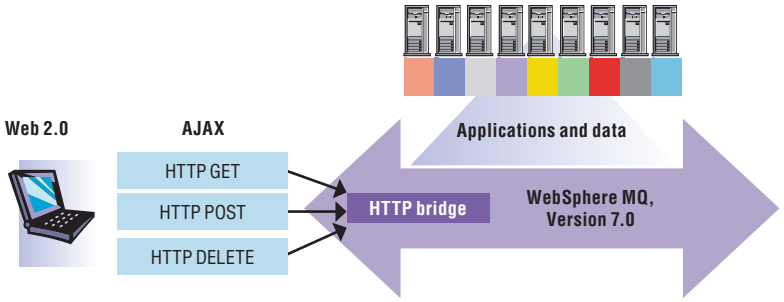

Figure 4. WebSphere MQ, Version 7.0 connects Web 2.0 with core enterprise systems, enabling a richer user experience and unlocking the value of business data.

Standards support

WebSphere MQ provides extensive support for industry and technical standards. WebSphere MQ, Version 7.0 enhances support for the industry-standard Java programming interface, JMS. It extends the JMS model to other programming languages through XMS, which provides interfaces equivalent to JMS but in more programming languages, such as C, C++ and C#. WebSphere MQ, Version 7.0 provides a bridge to HTTP networks, enabling rapid connectivity between Web 2.0 AJAX applications and core enterprise systems. This HTTP bridge offers a RESTful interface to simplify the experience for Web 2.0 developers. For tight integration with .NET environments, it provides a managed .NET client for Windows Communications Framework (WCF). WebSphere MQ enables SOAP messages to flow across its reliable transport, offering a higher transport quality of service for Web services than HTTP.

The service definition published for WebSphere MQ provides an Internationalized Resource Identifier (IRI) mapping for referencing topics and queues, and a Web Service Definition Language (WSDL) binding specification to represent WebSphere MQ applications, including the connections to the application, the queues or topics it uses, its message-exchange pattern (request-response or one-way) and the quality of service and message formats it uses. This service definition enables connected applications to appear as services in an SOA when accessed through WebSphere MQ. WebSphere MQ provides a JCA interface so that JEE Application Servers can substitute WebSphere MQ as their JMS providers and use the JCA interface to access the JMS services of WebSphere MQ. WebSphere MQ Explorer is based on the open source Eclipse platform, providing an extensible graphical user interface that shares the common look and feel of the IBM software portfolio and can be customized with user-defined Eclipse plug-ins. Security support is provided for both messages and WebSphere MQ Explorer connections through SSL. Certain releases of WebSphere MQ have been evaluated for Common Criteria to assurance level EAL4+.

Universal messaging backbone

In an SOA, an enterprise service bus (ESB) provides an integration layer that mediates, transforms and enriches data as it is transported between service components. The transport layer that underpins an ESB is a messaging backbone that enables ESBs to move data between both service and non-service assets. As a key member of the WebSphere software portfolio, WebSphere MQ delivers a universal messaging backbone that can connect virtually any commercial IT system and can help you take the first step to SOA.

WebSphere MQ enables SOAP interactions to flow over its messaging backbone between Web service requesters and providers. Heritage and batch applications that are enabled as Web services can also benefit from using WebSphere MQ in its asynchronous mode as a buffering mechanism to regulate the flow of requests made to these systems. WebSphere MQ is an excellent transport for adding reliability and traceability to service interactions, providing a scalable, reliable and resilient backbone that enables business-critical SOA. The WebSphere MQ service definition enables connected applications that are not service enabled to be seen as services by an SOA when accessed through WebSphere MQ.

For more information

To learn more about how IBM WebSphere MQ, Version 7.0 provides the messaging backbone for your SOA, and to find out how you can integrate your investments and reach your business and IT goals, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/webspheremq

IBM WebSphere MQ Version 7.0 at a glance

WebSphere MQ, Version 7.0 is supported on over 80 platform configurations by IBM and IBM Business Partners. WebSphere MQ maintains compatibility with the corresponding prior releases. For the latest information about supported platforms, visit: **ibm.com**/webspheremq/requirements

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
03-08

[1] Preliminary results observed on prerelease-level code. For the latest performance information, enter *performance report* in the search tool at **ibm.com**/webspheremq/support.

WSD10000-USEN-00