Enterprise service bus solutions
White paper

WebSphere® software

WebSphere Message Broker: Delivering business value through an advanced enterprise service bus.

December 2005

Contents

2 *Introduction*
2 *The integration challenge*
3 *Providing universal connectivity*
5 *Rich transformation and mediation capabilities help you make the most of your data*
8 *Flexible deployment and management for your advanced ESB*
9 *WebSphere Message Broker topologies*
11 *Delivering value through performance*
12 *Powering your event-driven architecture with complex-event processing*
13 *What is the real value to your business?*
14 *Conclusion*
15 *For more information*

Introduction

Business integration, along with the IT integration resources required to support it, has become an essential focus for companies across industries. These organizations want to update existing technology assets—and realize new value from what were previously disconnected and isolated applications. By using a flexible solution to integrate their most critical assets efficiently, businesses can help optimize their operations today and help ensure that their infrastructures are poised to take advantage of new technologies as they evolve.

This white paper focuses on the value that IBM WebSphere® Message Broker provides for businesses that want to streamline and optimize their business processes through an advanced enterprise service bus (ESB) solution. It discusses the key benefits delivered by an advanced ESB, such as universal connectivity and transformation, flexible deployment and management, and capabilities supporting an event-driven architecture. You can use this information to help understand how WebSphere Message Broker software serves as a foundational component from which you can build your service oriented architecture (SOA) today.

The integration challenge

The diagram in Figure 1 might remind you of how your organization's IT assets are connected—and it is, in fact, the actual topology of a major U.S. corporation's IT infrastructure. With new applications come new interfaces to build and maintain, which in turn create a barrier to future modifications to your IT infrastructure. As new standards emerge, integration developers must spend increasing amounts of time to develop support for and test against those standards internally. The result is a sea of connectivity as developers connect IT applications manually from point to point to solve today's problem without an eye to future requirements.

"The speed at which we can implement SOA is at least 30 percent faster using IBM WebSphere Message Broker, Version 6 as the underlying ESB compared to implementing SOA using tools from competitive vendors in the marketplace. By using IBM WebSphere Adapter solutions (technology and application adapters), the process of implementing business integration solutions has been greatly enhanced — resulting in reduced total cost of ownership [TCO] and a better return on investment [ROI]. Based on our experience, we are glad to say that the upgrade path to Version 6 from any previous versions of WebSphere Message Broker is painless and effortless. This is indeed a genuine way to protect any existing investments in IBM integration infrastructure."

– Prasad Lokam,
CEO and chief architect,
Miracle Software Systems, Inc.

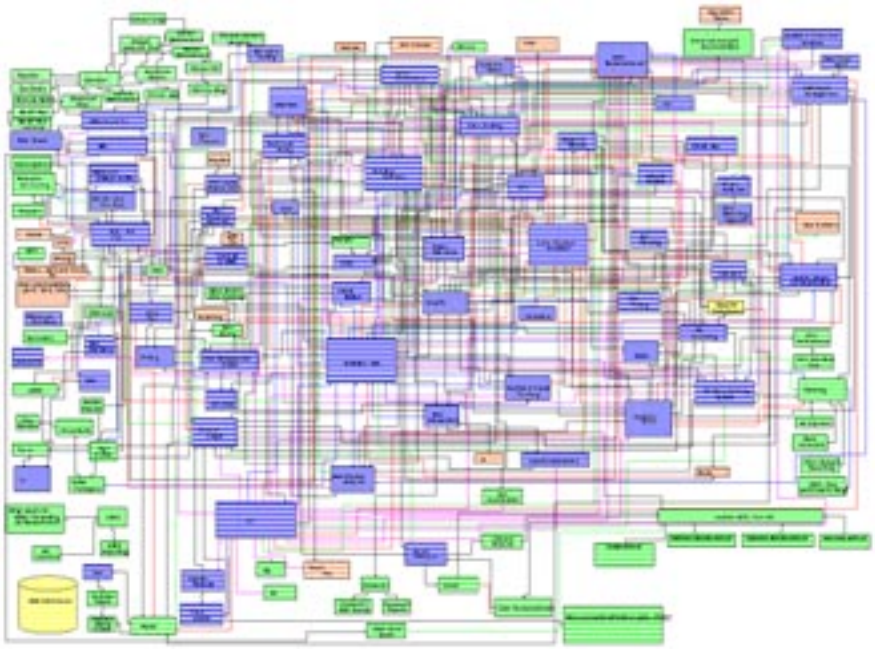


Figure 1. Topology of a major U.S. corporation's IT infrastructure

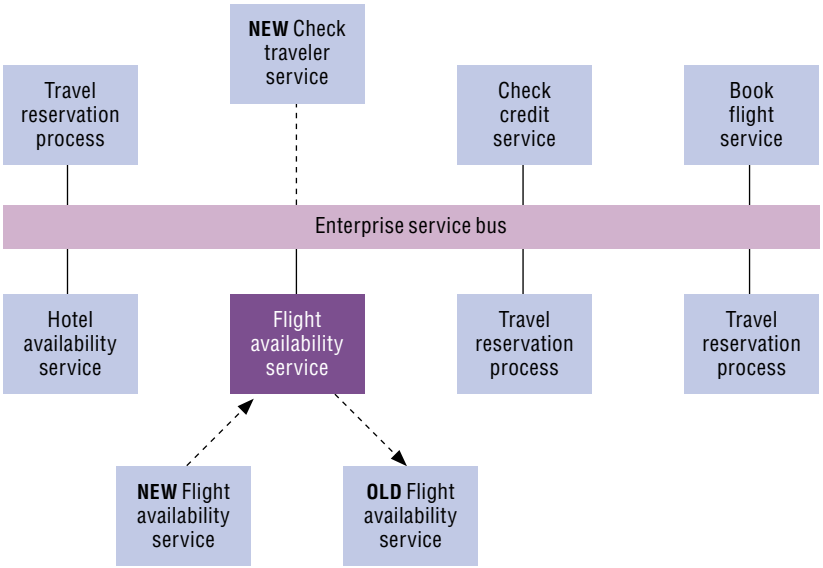
To illustrate this point further, consider a simpler example where an IT organization has 10 applications it needs to connect. The result is that the organization must develop and manage 90 (N*N-1) separate interfaces. Now suppose that same organization needs to add a new application or service: the result is 20 additional interfaces that must be built and maintained. The complexity that results from manual, point-to-point integration also increases bottom-line cost in terms of development time and expense—and in opportunities lost as customers and partners move to competitors with more effective solutions.

Providing universal connectivity

To reduce this cost and complexity, you need to optimize your integration solution by reducing the number of interfaces, as well as increasing the flexibility of managing those connections. By implementing an advanced ESB, you can help eliminate many point-to-point connections and help reduce the time required to develop, test and implement new services in your

IT organization as they become available.

WebSphere Message Broker is designed to address this need by providing universal connectivity for your integration needs. By enabling you to abstract your data away from protocol and transport specifics, an advanced ESB allows you to separate information distribution from the actual business logic that governs that distribution. This capability, in turn, makes the data centrally available within the ESB, and offers the potential to develop new, value-add services that can use your critical data. Through this abstraction, you can gain the flexibility to introduce new services gradually (as shown in Figure 2) without having to change existing application code to realize the benefits of those new services. Last, through the rich protocol- and message-format



conversion services of advanced ESB technology, you can easily add new information feeds to your systems as the need arises.

Figure 2. An advanced ESB lets you add new services gradually.

However, providing universal connectivity is just the first step in achieving the value offered by an advanced ESB. Because it acts as a central hub for your

enterprise messaging backbone, WebSphere Message Broker software can also provide a rich set of transformation and mediation capabilities.

Rich transformation and mediation capabilities help you make the most of your data

To provide smooth integration, processing, and management of the disparate information sources within today's IT infrastructures, IBM has developed a new way to present messages for processing. This method involves generating a logical message model that is built from the physical representation of the data. All manipulation within WebSphere Message Broker is defined by references to this logical message model. This way of defining messages means that a message flow implemented within WebSphere Message Broker can process messages, regardless of the data format in much the same way that relational database applications are not concerned about the physical storage of information. WebSphere Message Broker then becomes responsible for mediating between the abstract or logical model of the message data and the on-wire representation.

Managing diversity

The advantage WebSphere Message Broker brings by modeling all these messages is the ability to rise above the message-format detail. Whether it is a tag-delimited SWIFT or EDIFACT message, a custom-record format closely mapping a C or COBOL data structure, or XML, you can manage this diversity in a consistent, format-independent way.

Along with diverse data formats, many IT organizations have varying requirements in terms of how they process messages. WebSphere Message Broker is designed to provide a high level of control through different interaction patterns, including one-way messaging, request-response, aggregation and publish-subscribe.

WebSphere Message Broker enables point-to-point applications to use a request-reply or client-server model, or they can broadcast a message to many target applications using distribution lists. Other applications might require one-way send-and-forget or datagram traffic to exchange information with known partners. Within WebSphere Message Broker, each application is aware of the identity of the one or more applications with which it is communicating. You can create and configure message flows to process both send-and-forget and request-reply messages, and deploy them to WebSphere Message Broker.

An alternative message-processing model, *aggregation*, is the generation and fan-out of related requests derived from a single input message and the fan-in of the corresponding replies to produce a single aggregated reply message. The aggregation nodes within WebSphere Message Broker help improve response time because slow requests can be performed in parallel and do not have to follow each other sequentially. If the subtasks can be processed independently, and do not have to be handled as part of a single unit of work, they can be processed by separate message flows to help improve speed and efficiency. Through the various message-processing models supported by WebSphere Message Broker, you can gain a high level of flexibility and the ability to fine-tune your diverse messaging infrastructure to map to your specific business requirements.

Determining the right quality of service for reliability and efficiency

Many businesses have multiple business processes with differing requirements for quality of service. To accommodate these requirements, WebSphere Message Broker provides a range of delivery options. For companies that consider data integrity to be paramount, WebSphere Message Broker can provide the transactional support to help ensure this data integrity. For example, by using IBM WebSphere MQ and persistent messages, WebSphere Message Broker can enable the assured delivery of messages, once and once only. And when you update databases, two-phase-commit processing available with WebSphere Message Broker helps ensure that updates to WebSphere MQ messages and database resources occur together, or neither takes place.

For situations in which speed is most important, and assured delivery or transactional integrity is secondary, you can use WebSphere MQ nonpersistent messages, or for very high message rates, you can use the WebSphere MQ real-time transport protocol, together with the real-time optimized node to

support publish-and-subscribe processing.

Transactional benefits

Related to the ability to select the appropriate quality of service is the fact that WebSphere Message Broker message flows provide vital processing and data-manipulation capabilities, and as a result, are fully transactional. A message flow either completes all or none of its processing successfully. However, if required, individual nodes can elect to perform operations outside of the message- flow transaction (such as an audit).

Content-based routing supports your business needs

Message-processing nodes perform the real work of handling the messages within WebSphere Message Broker, which incorporates a set of built-in nodes that provide basic out-of-the-box functionality as well as the foundational functions for retrieving input messages or data, and routing output messages or data. This kind of message handling enables processing decisions to be based on the content of the message (either application data or information carried in the message headers), manipulating or transforming the information content of the message, and augmenting the content of the message with data extracted from a relational database system or capturing information content into a relational database system. As a result, WebSphere Message Broker enables you to automatically route data within your organization to the recipients that need it most, regardless of the original data format.

Publish and subscribe

The publish-and-subscribe function embedded in WebSphere Message Broker takes content-based routing capabilities a step further by enabling interested consumers to register their interest in particular sets of information with WebSphere Message Broker. Because the broker is handling a structured information flow, you can apply classification schemes to the information structure to provide more value. These schemes enable you to identify certain common key data items within a message, such as a topic- or subject-based publish-and-subscribe model. In this model, each inbound message is associated with a user-specified topic or subject. Users subscribe to and request delivery of information by specifying a topic or set of topics. Users can classify information by defining regular expressions or blocks of data across their entire message content. These kinds of classification schemes enable highly customizable control and management of information flow

through WebSphere Message Broker.

The primary advantage of WebSphere Message Broker is that it supports two approaches within a single, consistent framework. Both subscription filters (specified by an information consumer) and routing filters (defined by configuration of a message flow) — are expressed using the same language, based on the industry-standard Structured Query Language 3 (SQL3). Moreover, both approaches can be used together within a single message flow, which enables the construction of both simple and complex routing patterns and algorithms.

Flexible deployment and management for your advanced ESB

To improve your organization's flexibility and responsiveness, WebSphere Message Broker is designed to provide a variety of deployment options to match your business needs. WebSphere Message Broker includes four components:

IBM WebSphere Message Broker Toolkit

Microsoft® Windows® and Linux® systems use a broker-specific Eclipse perspective to develop message flows by assembling nodes to route and transform messages. Message flows, message definitions and any other associated files are packaged into deployment containers called *broker archive (BAR) files*. An administration perspective enables operations staff to deploy BAR files to any broker within the administrative domain. Administrators can view and control the full operational state of each broker from this perspective.

The broker

The broker is the run time where deployed flows operate within containers called *execution groups*, which appear as address spaces for IBM z/OS® implementations or operating-system processes for other platforms. Execution groups provide an excellent opportunity for vertical scaling and isolation through the ability to use multiple task-control blocks (TCBs) on z/OS or multiple threads on other platforms, as well as the ability to clone address spaces or processes. An individual execution group becomes capable of using multiple processors offering enhanced scalability when multiple copies of that same execution group are run.

Configuration manager

The configuration manager is the single management and administration component for a collection of brokers in a domain. The configuration manager controls all operational actions through access-control lists. It also monitors broker state, and holds topology information related to deployments and interbroker connectivity. All user toolkits are connected to a configuration manager.

User name server

The user name server component is used in publish-subscribe networks to determine the set of users and groups either from the operating system or through a user-defined program or file. These values are sent to both the configuration manager and the broker for subsequent administrative and runtime processing.

WebSphere Message Broker topologies

WebSphere Message Broker Toolkit runs on Linux and Windows systems under the Eclipse environment using IBM Rational® Application Developer software. All other components reside on the user's platform of choice, and a broker domain should contain the appropriate mix of platforms to meet the message-processing needs of connected applications. Brokers can be deployed either individually as stand-alone processing engines or in a connected bus to create highly available and scalable architectures.

Whether you use a hub, bus or arbitrary graph topology is a decision you should make based on your architectural needs rather than on functional characteristics. You can arrange brokers in any topology necessary to meet the needs of your enterprise. You can also deploy message flows and their associated artifacts to one, many or all of the brokers within a topology. Applications connected to a broker node are able to interoperate with other applications connected to any other broker node within the topology using any protocol or message-format combination. Figure 3 shows how brokers can be arbitrarily connected together to create a topology that meets the scalability and availability needs required.

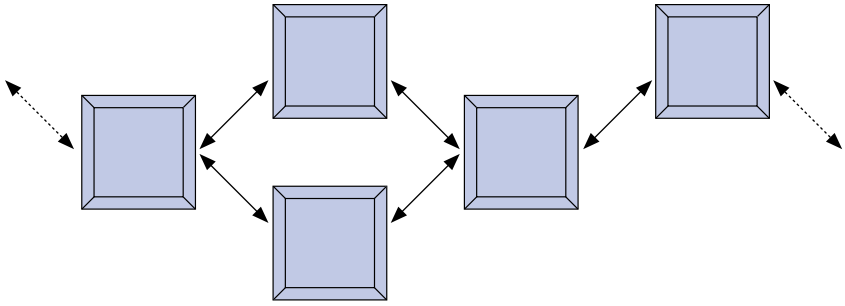


Figure 3. Creating a broker topology

You can also choose to create heterogeneous combinations of brokers on different operating-system platforms. The key requirement behind this need is to be able to link together brokers operating on the appropriate platforms to meet your business needs, and as a result, enable any and all applications throughout your enterprise to be connected. In many cases, this task can involve a combination of z/OS and other platforms, including IBM AIX®, Hewlett Packard (HP), Sun Solaris operating environment, Linux and Windows.

Coordinated reporting and systems

WebSphere Message Broker enables you to collect message-flow accounting and statistics data for an active broker at any time. You can select the granularity of the data that you want to collect by specifying the appropriate parameters in the associated command, and you can view the parameters in force for a broker, an execution group or an individual message flow. Along with providing deeper insight into how your solution is performing, the accounting and statistics features provide a robust tool for enabling chargebacks in a shared-services environment.

Delivering value through performance

WebSphere Message Broker offers a high-performance transformation and routing engine. And with the release of Version 6.0, IBM has taken significant steps to help reduce the cost of owning a broker by further enhancing performance and resource consumption.

A comprehensive range of tests involving different messaging styles, protocols and message formats were compared between WebSphere Message Broker, Version 6.0 and Version 5.0, and results have shown that in customer-oriented end-to-end messaging scenarios, messaging throughput increased by 50 percent with WebSphere Message Broker, Version 6.0. These improvements have been achieved without external product-interface changes, meaning that users migrating from previous releases can see an improvement for existing message flows without having to change their integration logic.

These performance improvements have arisen from work in the following areas:

- *Industry-based (such as SWIFT) and record-based (COBOL and C) parsers have been improved by a factor of up to four.*
- *Implementation has been thoroughly analyzed and improved by an average factor of two.*
- *Locking and scalability enhancements help reduce processing cost and wait times in multiprocessing environments.*
- *WebSphere Message Broker enables you to use a WebSphere MQ technology-based implementation for intermediate data storage rather than having to rely on a database to provide up to four times improvement in throughput.*

Extensive samples library to get you started quickly

WebSphere Message Broker, Version 6.0 includes a rich library of prebuilt samples that contain examples of the most common types of processing and enable you to get started quickly and save development time. The samples include use of key WebSphere Message Broker features such as aggregation, message transformation, use of WebSphere MQ queues for intermediate data storage and parsing techniques for managing larger messages.

Powering your event-driven architecture with complex-event processing

Businesses across industries rely on the generation and processing of events to be successful. An event might be part of a business process, such as issuing a trade order, landing an aircraft in a specific flight pattern, reading data using a sensor or monitoring information gathered about IT infrastructure performance. However, most IT infrastructures cannot effectively manage the high volume of events that flow through their enterprise operations in real time each day. Also, most IT infrastructures are incapable of identifying and maintaining relationships among discrete events from various groups within and outside the enterprise and correlating those events with meaningful historical context.

New market forces exist today that have introduced the need for more sophisticated processing of events, such as demonstrating compliance with regulations and the drive for expense reduction in back-office costs, which has boosted the demand for more efficient automation such as automated exception handling. WebSphere Message Broker includes rich complex-event-processing capabilities that enable you to analyze clouds of events at the application, enterprise or worldwide level, and perform validation, enrichment, transformation and intelligent routing based on a set of business rules.

To understand the value of complex-event processing, consider the following example. A large banking firm wanted to offer rich features to its customers to differentiate itself from its competitors. Using WebSphere Message Broker, the bank built an alert system that could receive information from various sources and in a variety of formats, and applied business functionality to that information. Now, the bank uses mail and short message service (SMS) to send alerts to users from its proprietary applications. As a result, customers can receive up-to-the-minute information from internal bank applications on their mobile phones. WebSphere Message Broker filters the data, transforms it and detects interesting situations (complex events), such as when the customer account balance has exceeded a certain threshold. However, the alert is issued only once when the account crosses the threshold. The next report is issued when the threshold is regained and then crossed again.

Xerox enhances productivity with IBM ESB solution and SOA

Copying has made Xerox a household name — but the widespread duplication of efforts to custom-code new business applications for its many product divisions created a bottleneck that hampered productivity. Developing new applications from scratch was wasting resources, especially because many applications shared common back-end databases and enterprise resource planning (ERP) and customer relationship management (CRM) systems.

To centralize its programming efforts and bring costs under control, Xerox created the Integration Competency Center. This group set to work to build an IT architecture that would enable them to reuse coding assets and take advantage of a common infrastructure to integrate a large number of applications. After evaluating the feasibility of using an ESB solution to connect disparate software assets, Xerox chose WebSphere Message Broker to be the foundation of their solution based on its full failover capabilities using message-oriented, event-driven and Web services.

With its new WebSphere Message Broker ESB solution, Xerox has experienced a 100 percent payback of its investment in only 24 months — and estimates that it is saving US$720000 annually in the cost of making changes to its applications. And application changes take just 25 percent of the time they took previously.

Many computer-based systems enable a single event to be processed, such as making a withdrawal from an automatic teller machine (ATM), a railroad train passing through a detection point or an order being placed for clothing on the Internet. Although many business-rules engines on the market can enable complex processes following the occurrence of a single event, few facilities are available to handle the correlation of multiple events, such as all the events surrounding an Internet order (including the order, shipping, customer receipt, payment and inventory management). Instead, these events are usually handled in an ad hoc manner.

What is the real value to your business?

Now that this white paper has discussed some of the key attributes and benefits of the advanced ESB capabilities of WebSphere Message Broker, the next step is to help you understand the actual value WebSphere Message Broker can deliver for your business.

Helping to reduce costs

By enabling you to use your existing core IT assets and build a solution with an eye toward integrating future applications and services with that core infrastructure, WebSphere Message Broker can help you to drive down costs. You do not need to write an XML parser or spend time pulling information from the Internet. Instead, you can focus on maximizing the use of data across your business rather than being consumed with data transport and distribution issues.

By providing an advanced ESB, WebSphere Message Broker helps reduce development and operational costs through the use of a consistent approach within your company rather than individual application teams adopting their own approach. WebSphere Message Broker also promotes reuse of existing code such as Java™ and Extensible Stylesheet Language Transformations (XSLT) assets, as well as database stored procedures, helping drive increased profitability for your existing and future business processes.

Integration initiatives speed information exchange with customers, suppliers and banks

An IBM client in the chemicals and petroleum industry needed to consistently deliver timely communication to customers and partners, and to meet contractual commitments to restore customer confidence and avoid having to pay substantial fines. Implementing a WebSphere software solution that includes WebSphere Message Broker enabled the company to migrate 95 existing business processes and automate 150 processes, helping to increase business efficiency and individual employee productivity.

The company was also able to increase the number of monthly transactions by 84 percent (from 4500 to 8300) — with an associated 400 percent increase in dollar volume from US$50 million to US$200 million. Back-end applications integrate smoothly with interenterprise business processes. And the chemicals and petroleum company can deliver the right information to the right people, at the right time, in a reliable, security-rich environment.

WebSphere Message Broker also helps drive down administration costs through use of a hub-and-spoke implementation, enabling you to centralize control. As WebSphere Message Broker continues to evolve and provide support for new standards, your business can adapt more quickly and with less development expense than would be required to implement new technologies internally.

Attracting new customers and helping to increase revenue from existing customers
Because WebSphere Message Broker enables you to build new services or improve existing services quickly and with minimal impact to your existing IT infrastructure, your business can become more flexible and adaptive to your customers' and your partners' needs.

Conclusion
As new technologies emerge, and new standards become available, your business needs to build an IT infrastructure that is flexible and adaptable, so that you can respond to dynamic changes in your customers' needs and your competitors' offerings. As thousands of customers around the world have realized, WebSphere Message Broker plays a key role in providing the advanced ESB capabilities necessary to maximize the value of your core IT infrastructure and optimize the use of your critical business information across your entire value chain.

WebSphere Message Broker not only provides the capability to interface existing systems, but also provides a facility in which new services and applications can be rapidly developed to take advantage of new business opportunities. The advanced ESB capabilities offered by WebSphere Message Broker give you the tools to help reduce your operating costs and drive new business with your existing and prospective customers.

For more information

To learn more about IBM WebSphere Message Broker, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/websphere/wbimessagebroker

To join the Global WebSphere Community, visit:

www.websphere.org

© Copyright IBM Corporation 2005

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

Produced in the United States of America
12-05
All Rights Reserved

AIX, IBM, the IBM logo, the On Demand Business logo, Rational, WebSphere and z/OS are trademarks of International Business Machines in the United States, other countries or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Linux is a trademark of Linus Torvalds in the United States, other countries or both.

Other company, product or service names may be trademarks or service marks of others.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

G224-7549-00