



WebSphere software

**Delivering new business value to
the enterprise on a Java 2 Platform,
Enterprise Edition (J2EE) technology
and Web services base.**

Contents

- 2 Introduction**
- 2 What are Enterprises Services?**
- 5 Enterprise Services and the WebSphere software family**
- 5 An Enterprise Services scenario**
- 6 Business rule beans**
- 7 Extended messaging technology**
- 9 Internationalization**
- 10 Shared work areas**
- 12 Bidirectional CORBA connectivity**
- 13 C++ CORBA software development kit (SDK)**
- 14 ActiveX bridge**
- 15 Full integration with WebSphere Studio Application Developer, Integration Edition**
- 16 Summary**

Introduction

With the advent of Java™ 2 Platform, Enterprise Edition (J2EE) application servers, companies are beginning to enjoy the benefits of an industry-standard programming model that offers new levels of portability, flexibility, reuse and responsiveness. Complementing J2EE-based technology, the emerging set of Web services standards arm companies with even greater business flexibility to compete in a changing world. Simply put, Web services and J2EE-based technologies are good for business. The most sophisticated e-businesses, however, need to extend these industry standards to meet important enterprise requirements.

In response to the needs of the most demanding customers, IBM now delivers new Enterprise Services that build upon Web services and J2EE technology-based standards. This white paper explains the philosophy behind Enterprise Services. It also discusses the solutions they can provide through a fictional business scenario based on a hypothetical, representative enterprise customer. Each Enterprise Service is reviewed at a high level in terms of the business and technical benefits that it provides. All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

What are Enterprise Services?

Enterprise Services add significant value to the base application server. They plug into and augment IBM WebSphere® Application Server, Advanced Edition. Enterprise Services is not a separate runtime but instead builds directly on top of the core WebSphere Application Server runtime environment. They are one of three elements within WebSphere Application Server, Enterprise Edition, Version 4.1—the other two elements are the WebSphere Application Server, Advanced Edition and IBM TXSeries® (the separate IBM transaction processing monitor runtime for exceptionally high performance and throughput requirements).

Enterprise Services enhances the capabilities of Web services and J2EE-based technologies along two important dimensions:

- *Enterprise-class functionality*
- *Enterprise-class qualities of service*

The enterprise-class functionality dimension delivers Web services and J2EE technology-based programming model extensions that are focused on integration and adaptability. Enterprise-class functional requirements include:

- *Integration of externally administered business rules that encapsulate variable business practices and promote consistent policy*
- *Integration between messaging- and component-based paradigms, including a blending of synchronous and asynchronous communication patterns (both inbound and outbound); mappings between message elements and component attributes; and declarative transactional support*
- *Global integration that accommodates different cultural conventions and accounts for different geographical boundaries*
- *Application integration through transparent access to shared information across a distributed computing environment*
- *Application integration through Common Object Request Broker Architecture (CORBA) support, including bidirectional connectivity to most Object Request Brokers (ORBs) from multiple vendors and the ability to leverage non-Java language code (Results may vary by vendor and product, and in some instances, limited details may affect results.)*
- *Component model integration, including client- and server-based access to Enterprise JavaBeans (EJB) components from Microsoft® ActiveX environments*
- *Integration of existing Enterprise Information Systems (EIS) using an integrated development environment to create visual flows and application adapters*

The enterprise-class qualities-of-service dimension emphasizes desirable operation and deployment characteristics of an application server, such as availability, recoverability, maintainability, scalability, serviceability and manageability. Collectively, these characteristics bring the very highest qualities of service to a distributed computing platform for e-business. They define a leading-edge, best-of-breed infrastructure for mission-critical computing.

WebSphere Application Server, Advanced Edition is the first application server to integrate Web services with J2EE-based technology. It provides exceptional functionality, performance and scale.

IBM also offers WebSphere Application Server, Enterprise Edition to deliver even greater levels of scale and manageability for forward-thinking e-businesses. The high-end application server includes IBM innovations related to business process and application integration.

Enterprise-class qualities-of-service requirements include:

- Capabilities for massive scale across heterogeneous environments to support an ever-growing number of Internet clients, including a burgeoning set of pervasive computing and wireless devices
- High availability through extreme fault tolerance and nondisruptive remote systems management, providing an exceptionally satisfying end-user experience and supporting global business hours
- Enterprise-class manageability that facilitates highly dynamic system monitoring, tuning and maintenance

Enterprise Services included with WebSphere Application Server, Enterprise Edition, Version 4.1 deliver on the enterprise-class functionality requirements discussed in the preceding list, and these requirements are the focus of this white paper. Subsequent releases of this product will address additional enterprise-class qualities-of-service requirements.

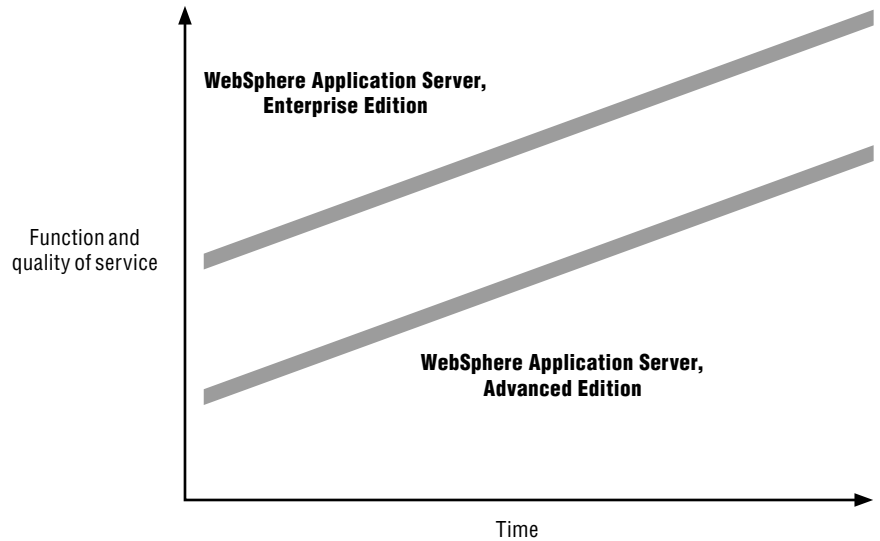


Figure 1. Relationship between WebSphere Application Server, Advanced Edition and WebSphere Application Server, Enterprise Edition

Enterprise Services and the WebSphere software family

Figure 1 depicts the general relationship between WebSphere Application Server, Advanced Edition and WebSphere Application Server, Enterprise Edition. Both editions provide increasing functionality and ever-higher qualities of service over time. As J2EE platform and Web services standards evolve, functionality and qualities-of-service levels once reserved for WebSphere Application Server, Enterprise Edition will be incorporated into WebSphere Application Server, Advanced Edition. However, at any given point in time, WebSphere Application Server, Enterprise Edition will offer the most functionality and deliver higher qualities-of-service levels.

An Enterprise Services scenario

The following hypothetical scenario demonstrates how sophisticated e-businesses can benefit from Enterprise Services. At Dynamic Corporation, the company's rapidly changing business requirements have strained the ability of information technology (IT)—the engine of the company—to keep up with the changes needed for the business to stay competitive. There is a gap between the business strategy and what IT can deliver quickly. Problems include the need for integration of different systems to work together within the company and with other companies, innovation in deploying and adapting to new business models and agility in meeting the time frames demanded by competitive pressures.

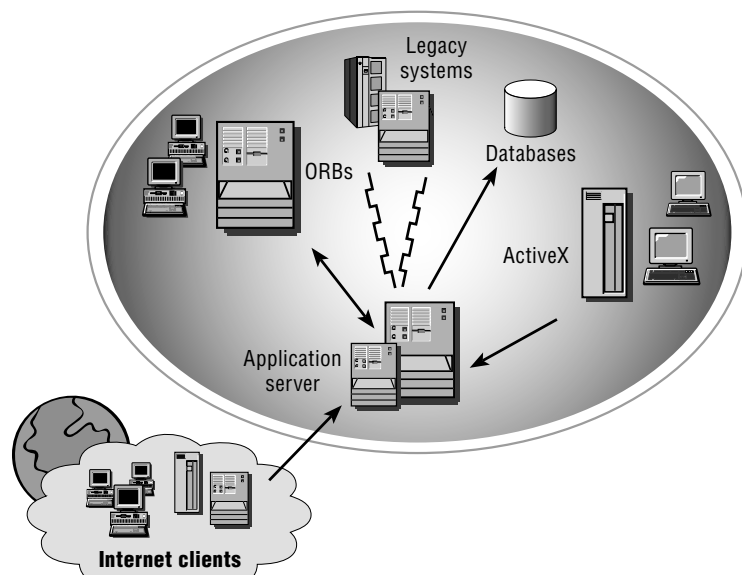


Figure 2. The application server as a business logic hub

Like a growing number of today's large enterprises, Dynamic Corporation has made a strategic commitment to J2EE-based technology as an important step toward meeting these challenges. In an architecture centered on J2EE-based technology, you can view the application server as a distributed *business logic hub*. Figure 2 illustrates a simplified view of an e-business architecture, placing special emphasis on the application server dimension. The remaining sections of this white paper outline specific enterprise requirements faced by Dynamic Corporation's IT professionals. In each section, you will find a description of the issues surrounding the requirement, along with a solution that utilizes one of the Enterprise Services that IBM delivers with WebSphere Application Server, Enterprise Edition.

Business rule beans

Dynamic Corporation competes in a rapidly changing business environment. As a result, the company must frequently change business processes to meet evolving customer needs and to comply with changing government regulations. Dynamic Corporation needs the ability to model its business logic in the form of reusable and externally manageable business rules. The company can address this requirement through an Enterprise Service that IBM delivers called business rule beans.

Business rule beans provide a way to externalize and apply parameters to potentially pervasive and variable business practices, maintain rules administratively and change business behavior without exposing any code. This means that you can change business behavior by maintaining your business rules instead of changing your application code. Business rule beans enable developers to separate reusable abstractions within a business domain – and base these abstractions on policies, regulations or any other business practice that tends to change over time. Most rules engines are inference engines. This is not the case with business rule beans, which are actually a lightweight framework. The framework includes a number of basic business rules, while application developers can build others. You can write rules to derive values, classify situations based on input values and execute scripts. Business analysts can then manage rules externally through the sophisticated capabilities for rule modification offered as part of business rule beans technology.

This technology is different from the functionality in the Versata Logic Server for WebSphere, which captures and executes Declarative Business Logic for large, J2EE-based application systems. Declarative is a new perspective where the desired result is specified in business-oriented terms associated with procedural approaches (*what* rather than the technology-oriented *how* results, such as architecture, design, coding, testing, debugging and maintenance). Declarative logic is executable through engines in the Versata Logic Server for WebSphere that operate within the context of WebSphere Application Server. This approach enables many improvements in time to market and maintenance, dramatically reduces the complexity and risk of large projects and enables nontechnical staff to participate effectively with IT staff to deliver business systems.

Business rule beans, on the other hand, are built from a Java technology-based perspective. Business rule beans provide a useful facility for programmers to achieve reuse. Versata rules represent the application, whereas business rule beans supplement the application. Business rule beans also tend to be parameter-driven—behavior can be dynamically defined through externally maintained data values, which is less common in Versata. Because Versata logic is extensible because it can invoke Java methods, and because business rule beans are Java technology-based, it is effective to use Versata and business rule beans together to form a powerful solution for your e-business.

Extended messaging technology

Another requirement of Dynamic Corporation concerns the seamless blending of component and messaging technologies. The architecture, shown previously in Figure 2, highlights the need for inbound and outbound messaging flows to link external applications and systems with the application server acting as the business logic hub. For example, Dynamic Corporation wants to link an existing J2EE-based Web application for order entry, with an existing customer relationship management (CRM) application. Specifically, the company wants to design a system that is capable of presenting orders to the Web application from a variety of sources, one of which is this CRM system. The company feels that an asynchronous interface is most appropriate for the CRM system because, in many instances, customers are on the telephone where sluggish responses are not well tolerated. An asynchronous link to the order-entry application would allow customer requests to be acknowledged and posted immediately with the compute-intensive fulfillment processing taking place at some later time.

IBM addresses this requirement through an Enterprise Service that includes message beans and Java Message Service (JMS) listener technology. This technology represents an earlier version of the J2EE 1.3 message-driven beans and is virtually identical to the functionality outlined in the J2EE-based specification. The initial support for component and messaging integration is depicted in Figure 3. A JMS listener running in the background pulls messages arriving in an inbound queue into the application server. Message content is then dispatched against a general-purpose message bean from which specific business logic can be invoked. The general-purpose bean is actually general for a given destination queue. That is, different destination queues will have different general-purpose beans. This Enterprise Service solves the common problem of managing the integration between inbound messages and Enterprise JavaBeans technology-based operations.

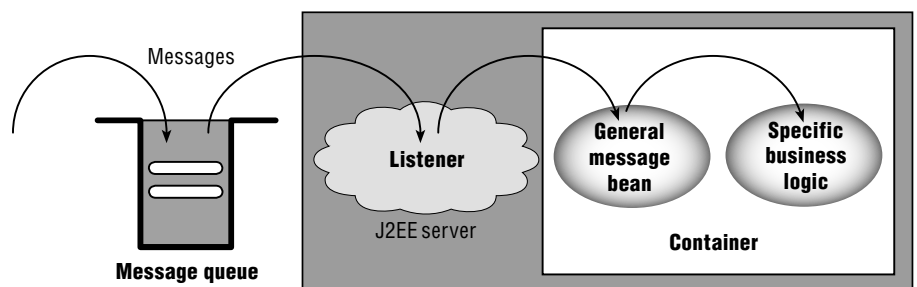


Figure 3. JMS listener and message bean support

Transactional support is also provided for the retrieval of a message from a queue. This means that as you update resources during subsequent application server processing, you can roll back any failures that are encountered, along with the original message. This helps ensure that the system manages message handling and subsequent processing automatically and with high integrity. The transactional solution represented here builds on the JMS/XA support already included in WebSphere Application Server, Advanced Edition. Message beans and JMS listener technology do not provide a complete answer because developers must be proficient in JMS programming. A subsequent release of this Enterprise Service is planned to deliver even broader solutions that:

- *Eliminate the need for JMS programming expertise*
- *Support outbound (not just inbound) messaging requirements*
- *Provide robust code generation and tooling support*
- *Facilitate and manage mappings between messages and components*

Internationalization

Dynamic Corporation wants to efficiently build a single Web application, capable of conducting transactions in several countries around the world. The company wants to run the application from a centralized location. As a result, Dynamic Corporation requires an application server that recognizes different client time zones and different client locales. Specifically, objects that return or use a client's local time must be aware of the time zone with respect to that client. Moreover, Dynamic Corporation wants to express dates and times correctly for clients according to the cultural conventions of each locale. Dynamic Corporation also desires that currency and decimal points be displayed correctly because the representation of these differ across national boundaries. Finally, the company wants product lists to display correctly as sorting rules vary by language. These concepts are illustrated in Figure 4.

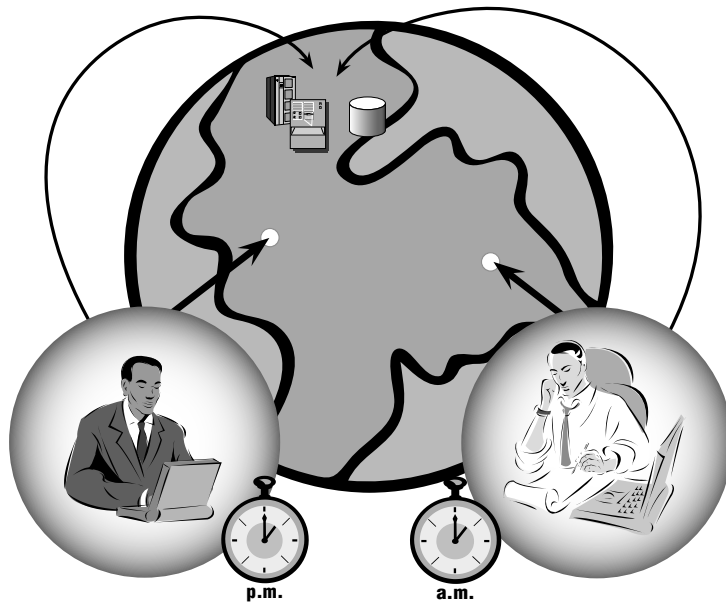


Figure 4. Internationalization requirements

IBM nonintrusively delivers on these requirements through the Enterprise Service for internationalization that provides simple application program interfaces (APIs) that server-side business objects can use to extract propagated information for localizing relevant computations. The fact that these are delivered nonintrusively is significant. Opening up method signatures to pass along client locale and time zone information is error-prone and tedious, and is especially burdensome for already

deployed applications. As is true of Enterprise Services in general, this particular Enterprise Service provides an example of IBM ability to build upon established standards. In this case, IBM addresses end-to-end internationalization issues. In fact, IBM has submitted a Java Specification Request (JSR) that addresses the enterprise requirement for internationalization.

How does this Enterprise Service relate to the WebSphere Translation Server? WebSphere Translation Server is compatible with language translation of Hypertext Markup Language (HTML), plain text and IBM Lotus® Notes® files. The Internationalization Enterprise Service does not do any translation. It only identifies locale and time zone information, which means that a program using internationalization Enterprise Service could apply appropriate translation on its own, if necessary. It could actually invoke WebSphere Translation Server to do this.

Shared work areas

Dynamic Corporation has distributed applications that comprise many parts, including servlets and JavaServer Pages (JSP) pages, as well as EJB-based technology. In some cases, the company has thick Java clients that provide direct support for end-user interactions. In designing new applications, Dynamic Corporation wants the capability to have different parts of the application record information of general interest to the application at large and to have efficient access to this shared information. Examples of potential shared information include identities, profiles and qualities-of-service descriptors for Dynamic Corporation customers.

IBM delivers the ability to transparently access shared information as an Enterprise Service, referred to as shared work areas. Think of work areas as virtual scratch pads that enable you to read and write arbitrary information, such as a customer's profile. By automatically sending the customer profile along the wire with the flow of execution, all elements of the distributed application implicitly obtain access to potentially useful information without the need to open up their interfaces explicitly. This can result in cleaner and more maintainable interfaces. It also should provide a higher-performing alternative than issuing remote calls or using persistent storage as the mechanism for information sharing. As with internationalization, IBM has submitted the shared work area service as a JSR.

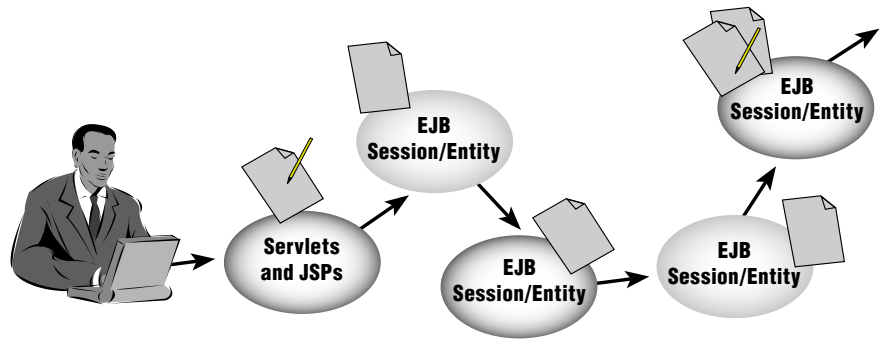


Figure 5. Transparent access to shared information through work areas

Figure 5 graphically illustrates how a shared work area might be used in practice. A servlet initially writes some information to a work area. This information flows along the thread of execution to an EJB module, which then reads the information and passes control to a second EJB component that chooses to ignore the work area. A third EJB component receives control, and it reads the work area again. It is significant that work areas flow implicitly because:

- *The second EJB module in this example is not required to receive, transmit or even know about the work area information. The work area does not affect its interface. None of the EJB components is required to adapt its external interfaces to account for the existence of the work area. This promotes flexibility and keeps the interfaces clean.*
- *Work areas are always immediately accessible and highly efficient – there is no need for any calls to a shared database, directory or remote object. This reduces bottlenecks and promotes high performance.*

In the last part of Figure 5, yet another EJB component chooses to write additional information into a work area. It actually pushes information to the work area that effectively overrides the previous value. Other parts of the distributed application then process the new value, and the original value does not resurface until control returns to the point at which the new value was written.

Bidirectional CORBA connectivity

Like many other companies, Dynamic Corporation already has existing investments in CORBA. Dynamic Corporation needs to be able to invoke its CORBA applications from servlets and EJB components (calling *in* to CORBA), as well as invoke EJB tags from the same CORBA applications (calling *out* from CORBA). Bidirectional CORBA connectivity delivers another valuable service function that provides broad support for CORBA and EJB interoperability. This enables customers with existing CORBA investments to incorporate new business value based on the J2EE technology-based model while continuing to leverage their existing systems.

CORBA connectivity support consists of numerous samples and documentation (including restrictions). The following CORBA environments are supported to varying degrees: VisiBroker C++, Version 3.3.3; VisiBroker Java, Version 3.4; VisiBroker C++/Java, Version 4.0/4.1; Orbix C++, Version 3.0.1 (except Sun Solaris); Orbix C++, Version 3.0.2 (Sun Solaris operating system only); Orbix Web, Version 3.2 and Orbix2000 C++/Java, Version 1.2. Initially supported platforms include IBM AIX®, Version 4.3.3; Microsoft Windows NT®, Version 4.0; Microsoft Windows® 2000 and Sun Solaris operating system, Version 2.7.

Calls from either servlets or EJB components to CORBA are especially interesting. You can issue these calls using either *coexistence mode* or *interoperability mode*. With coexistence mode, you invoke distributed objects using a third-party ORB running in the WebSphere software platform environment. This enables use of the third-party ORB bootstrapping protocol and vendor-specific APIs. With interoperability mode – which is the preferred mode – you invoke distributed objects using the ORB included with WebSphere Application Server. This enables the propagation of service contexts, such as in-progress transactions. Because these service contexts are potentially usable by the third-party ORB receiving the calls, you may, for example, be able to include processing performed by the third-party ORB within the scope of a WebSphere software-initiated transaction. In addition, because interoperability mode uses the embedded WebSphere Application Server ORB, you automatically enjoy the performance, stability, scalability and interoperability IBM provides in its standards-compliant implementation of the Internet Inter-ORB Protocol (IIOP).

C++ CORBA software development kit (SDK)

Dynamic Corporation has either built or acquired many valuable libraries of C and C++ code over the years. Dynamic Corporation would like to integrate these assets with the J2EE technology-based environment in a standards-compliant way. How can the company do this? One technique is to leverage CORBA technology. As an Enterprise Service, IBM delivers a C++ CORBA software development kit (SDK) that customers can use for building a lightweight CORBA server to help leverage, transform or adapt existing C and C++ programs. IBM delivers this CORBA solution as an Enterprise Service so that customers do not need to purchase a CORBA solution from other vendors. Customers can also choose to replace an existing CORBA server with the planned solution from IBM. (IBM obtained CORBA 2.1 certification for WebSphere Application Server, Enterprise Edition, Version 4.0, and support for later versions of CORBA is possible for future releases.)

Figure 6 illustrates how you can use an IBM-based CORBA server behind or in front of the J2EE-based server. Customers can also leverage the SDK to build a C++ CORBA client, without needing to create a CORBA server at all. This provides a simple and effective way to connect a C++ desktop application to the J2EE server, for example.

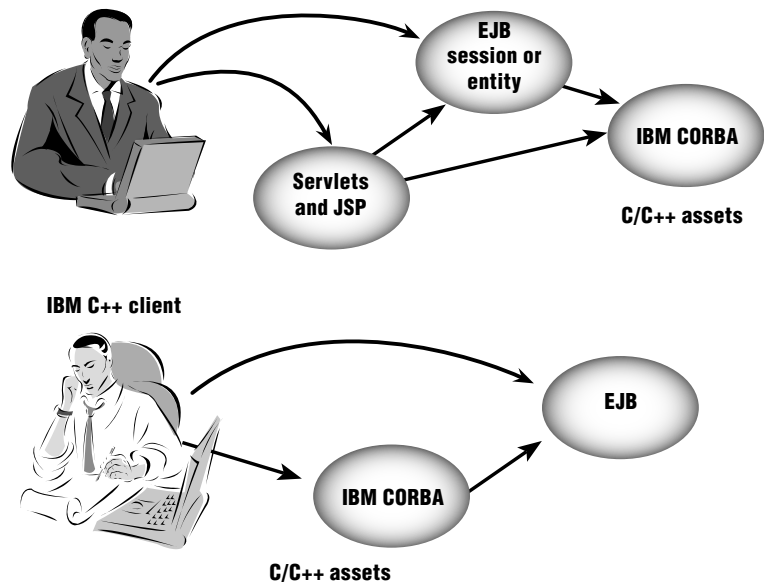


Figure 6. Using IBM-supplied C++ CORBA technology with J2EE-based technology

Accessing J2EE services from a C++ ORB requires that both standard and application-specific Java classes exposed through EJB interfaces be duplicated in the C++ runtime. To facilitate Enterprise development, IBM's C++ CORBA SDK includes a value-type library that defines IBM C++ CORBA classes corresponding to a number of standard Java classes. These include Boolean, Character, Integer, Float, Vector, List, and others. This value-type library is not compatible with ORBs from other vendors. Consistent with the strategic emphasis on J2EE-based technology at IBM, the C++ CORBA solution provides a basic CORBA environment. It can bootstrap into the J2EE-based name space, and it can invoke J2EE technology-based transactions. However, it does not ship with its own naming and transaction services. For this, a C++ CORBA client or server relies on the J2EE technology-based environment as a service provider. The C++ CORBA technology will first be provided on Sun Solaris operating system (Forte, Version 6.0 C++), IBM AIX (VisualAge[®] for C++, Version 5.0), and Microsoft Windows NT and Microsoft Windows 2000 (Microsoft Visual C++, Version 6.0).

ActiveX bridge

Looking back at the Dynamic Corporation topology, a requirement exists for J2EE-based connectivity from ActiveX (in fact, from both ActiveX clients and ActiveX servers). Like many other companies, Dynamic Corporation has made a considerable investment in COM-based technologies, such as Visual Basic, Visual C++ and Active Server Pages (ASP). It is important that newly developed EJB modules be made accessible from the world of ActiveX.

IBM delivers support for ActiveX integration with J2EE-based platform through an ActiveX bridge. The bridge makes Java-based classes dynamically available through ActiveX interfaces. This idea is depicted in Figure 7.

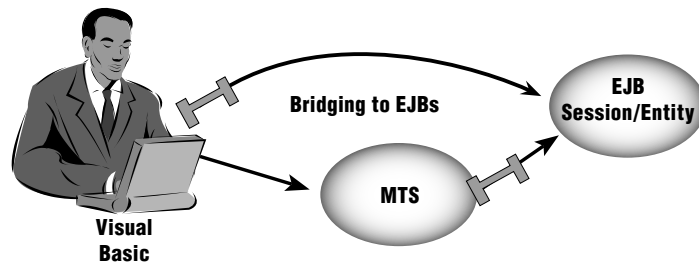
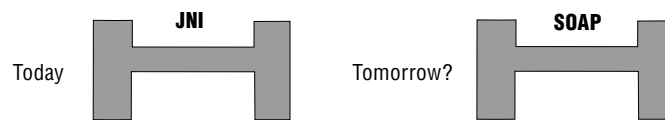


Figure 7. The ActiveX bridge provides a generic ActiveX-to-Java-based classes bridging solution.

Both the Visual Basic client desktop application and Microsoft COM-based server (MTS) utilize the IBM-supplied bridge to communicate with a WebSphere software EJB module. ActiveX automation interfaces are available by means of Visual Basic, VB scripting and so on. The bridge presently consists of a C++ DLL that calls some Java-based code using Java Native Interface (JNI). The Java technology-based code invokes the services of either a J2EE technology-based client container (preferred on thick desktops) or a Java proxy (required on the Microsoft server), which in turn make a call to a Java ORB. The ActiveX bridge is available on both Windows 2000 and Windows NT, although this is in one direction only (as shown in Figure 6). Through the bridge, an ActiveX environment can also use J2EE APIs, such as Java Naming and Directory Interface (JNDI), Java Database Connectivity (JDBC), JMS and JavaMail. The bridge supports both workload management and security.



These capabilities constitute higher qualities-of-service than is currently available using Simple Object Access Protocol (SOAP), the language-neutral wire format and a core element of the services-oriented architecture known as Web services. However, because SOAP may evolve to become an attractive alternative transport, the current JNI-based bridge is flexible to enable the possible substitution of a SOAP-based implementation in the future.

Full integration with WebSphere Studio Application Developer, Integration Edition

Dynamic Corporation has made the commitment to move their business into the future by utilizing J2EE-based technology. Through the years, Dynamic Corporation has made significant investments in non-J2EE-based technologies and is now faced with the dilemma of integrating these investments with the Java technology-based environment. WebSphere software helps address these needs with IBM WebSphere Application Server, Enterprise Edition and IBM WebSphere Studio Application Developer, Integration Edition that together can help leverage and protect existing IT assets. WebSphere Studio Application Developer, Integration Edition is a development environment that fully supports WebSphere Application Server, Enterprise Edition.

The Intelligent Corporation is an international enterprise that plans to build a globally accessible Web-based order-fulfillment system. The corporation wants to reward its largest customers with higher levels of end-to-end service. Intelligent expects to accept requests from browsers and call centers that batch requests asynchronously. The company will determine shipment dates through dynamic scheduling algorithms that reflect both regional policies and multiple time zones. As part of the fulfillment process, Intelligent wants to query and update a number of heterogeneous systems with integrity (although some of these systems are not subject to normal transactional coordination). In a few cases, it must also link libraries of C and C++ code into the solution. And finally, Intelligent must interface an existing set of front-end applications with the fulfillment system (and most of these applications have been written using Microsoft technologies).

What application server should Intelligent Corporation choose? IBM WebSphere Application Server, Enterprise Edition can deliver the Enterprise Services needed to solve the demands of the most complex e-businesses.

One of the quickest, most cost-effective ways to e-business is to reuse existing applications. WebSphere Studio Application Developer, Integration Edition and WebSphere Application Server, Enterprise Edition enable Java developers to build and deploy adapters quickly and easily to provide access to enterprise resources from a Java application. Application adapters quickly and easily extend enterprise applications to e-business – helping leverage current resources to build new applications. WebSphere Studio Application Developer, Integration Edition helps reduce the risk, complexity and cost of creating and customizing application adapters and helps eliminate the need for extensive coding. It provides enterprise Java developers and application integration specialists with powerful graphical tools for the rapid creation of application adapters. Because applications are often customized, the tool also enables fast customization of the adapters, generally without the need for advanced programming skills.

WebSphere Application Server, Enterprise Edition and WebSphere Studio Application Developer, Integration Edition also offer support for service choreography through the creation of visual flows. For example, developers could create basic service definitions through adapter tooling that visually connects Java technology-based applications to EIS. These basic services could be choreographed into composed services that perform higher-level business activities. Visually linking these interactions can make it easier for developers to create applications and to preserve the flow structure of the application when underlying service implementations change over time. This service choreography provides an interactive technology to allow businesses to be more efficient and productive by automating workflow.

Summary

With the release of WebSphere Application Server, Enterprise Edition, Version 4.1, IBM continues its tradition of providing high-quality service to enterprise customers. WebSphere Application Server, Enterprise Edition is supported by an evolving set of Enterprise Services that delivers the functionality and qualities of service that competitive businesses need to solve the most demanding business and technical problems. Customers who invest in the WebSphere Application Server, Enterprise Edition will buy middleware that extends the core WebSphere runtime and delivers leading-edge functionality and higher-quality service.

As a result, sophisticated customers with high-end needs will be able to integrate, innovate and adapt quickly in response to relentless competitive forces. WebSphere Application Server, Enterprise Edition customers will leverage their existing investments and build upon a reliable infrastructure that positions them for leadership and growth.

For more information

For more information about WebSphere Application Server, Enterprise Edition, contact your local IBM representative or visit ibm.com/websphere



© Copyright IBM Corporation 2002

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

Produced in the United States of America
07-02
All Rights Reserved

AIX, the e-business logo, IBM, the IBM logo, TXSeries, VisualAge and WebSphere are trademarks or registered trademarks of International Business Machines in the United States, other countries or both.

Lotus and Lotus Notes are trademarks or registered trademarks of Lotus Development Corporation and/or IBM Corporation in the United States, other countries or both.

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

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

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