



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



---

<b>Contents</b>	
<b>2</b>	<b><i>Introduction</i></b>
<b>2</b>	<b><i>What is an Enterprise Service?</i></b>
<b>4</b>	<b><i>Enterprise Services and the WebSphere software family</i></b>
<b>5</b>	<b><i>An Enterprise Services scenario</i></b>
<b>7</b>	<b><i>Business rule beans</i></b>
<b>8</b>	<b><i>Message beans and JMS listener</i></b>
<b>9</b>	<b><i>Internationalization</i></b>
<b>11</b>	<b><i>Shared work areas</i></b>
<b>13</b>	<b><i>Bidirectional CORBA connectivity</i></b>
<b>14</b>	<b><i>C++ CORBA software development kit (SDK)</i></b>
<b>15</b>	<b><i>ActiveX bridge</i></b>
<b>16</b>	<b><i>So what's next?</i></b>
<b>17</b>	<b><i>Summary</i></b>
<b>17</b>	<b><i>For more information</i></b>

## **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 technology, the emerging set of Web services standards arm companies with even greater business flexibility to compete in a changing world. Simply put, J2EE and Web services 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 intends to deliver new Enterprise Services that build upon J2EE and Web services standards. This white paper begins by articulating the philosophy behind Enterprise Services and relates the major solution themes that Enterprise Services address. An examination of some of these planned Enterprise Services follows in the context of a fictitious yet 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 is an Enterprise Service?**

Enterprise Services are intended to add significant value to the base application server. They will plug into and augment IBM WebSphere® Application Server, Advanced Edition. Enterprise Services will not be a separate runtime but instead will build directly on top of the core WebSphere Application Server runtime environment. They will be one of three elements within WebSphere Application Server, Enterprise Edition, Version 4.0 – the other two elements will be the WebSphere Application Server, Advanced Edition, itself, and IBM TXSeries® (the separate IBM transaction processing monitor runtime for exceptionally high performance and throughput requirements).

Enterprise Services will enhance the capabilities of J2EE and Web services technologies along two important dimensions:

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

The enterprise-class functionality dimension will deliver J2EE and Web services programming model extensions 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 Object Request Brokers (ORBs) from multiple vendors and the ability to leverage native code (that is, programming assets not written in the Java language)*
- *Component model integration, including client- and server-based access to Enterprise JavaBeans™ (EJB™) from Microsoft® ActiveX® environments*

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

Enterprise-class quality-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 supporting global business hours*
- *Enterprise-class manageability that facilitates highly dynamic system monitoring, tuning and maintenance*
- *Ability to optimize application performance to meet the needs of the most sophisticated e-businesses*

Enterprise Services included as part of WebSphere Application Server, Enterprise Edition, Version 4.0 intend to deliver on only the previously listed enterprise-class functionality requirements and therefore is the focus of this white paper. Subsequent releases of WebSphere Application Server, Enterprise Edition intend to address the enterprise-class qualities-of-service requirements.

#### **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 will provide increasing functionality and ever-higher qualities of service over time. As J2EE and Web services standards evolve, functionality and quality-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 quality-of-service levels.

*WebSphere Application Server, Advanced Edition is the first application server to integrate Web services with J2EE 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 sophisticated e-businesses. The high-end application server includes IBM innovations related to business process and application integration.*

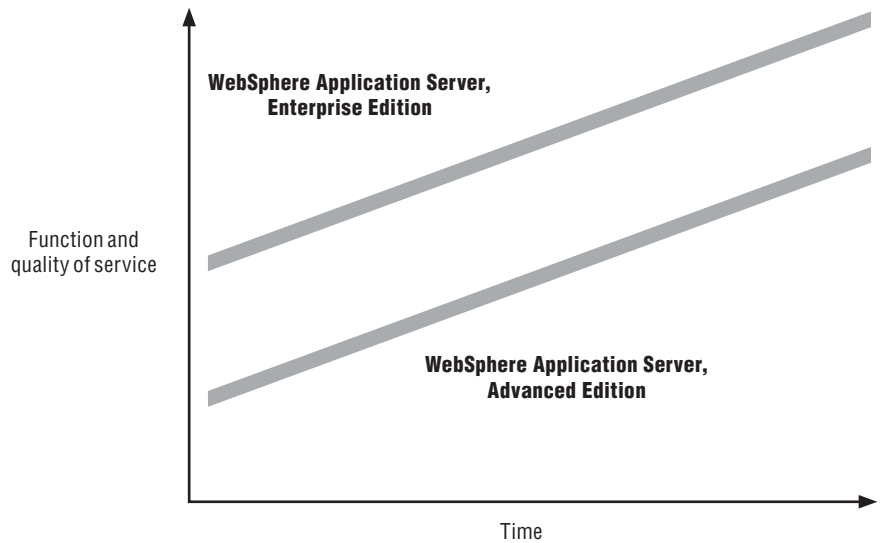


Figure 1. Relationship between WebSphere Application Server, Advanced Edition and WebSphere Application Server, Enterprise Edition

#### **An Enterprise Services scenario**

The following fictitious yet representative example 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 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: 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 scales demanded by competitive pressures.

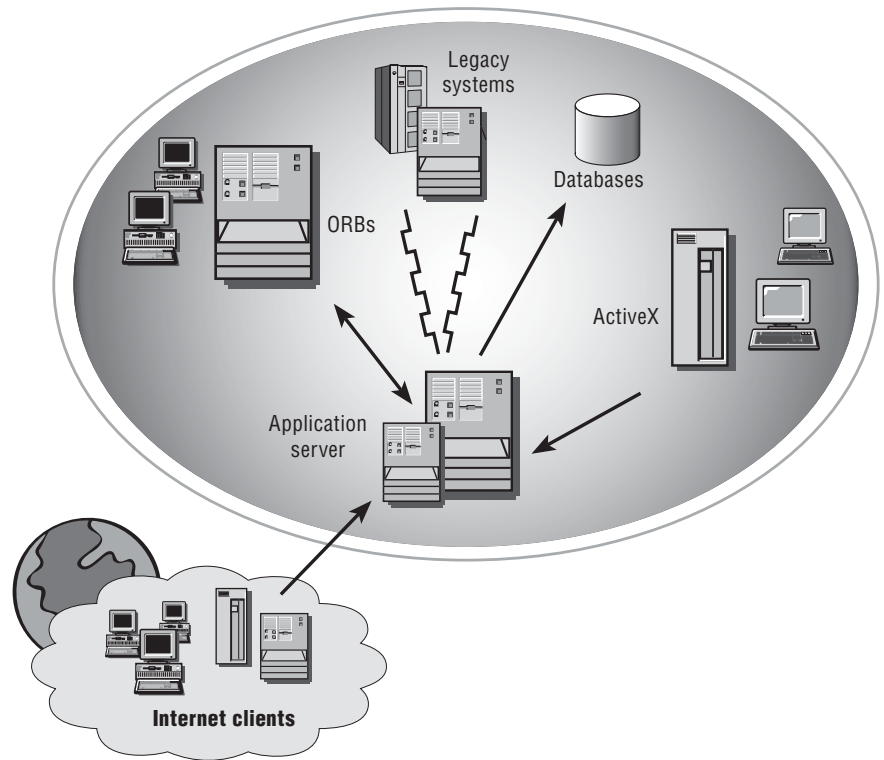


Figure 2. The application server as a business logic hub

Like a growing number of today's large enterprises, Dynamic Corporation has made a strategic commitment to J2EE technology as an important step towards meeting these challenges. In an architecture centered on J2EE 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 which utilizes one of the Enterprise Services that IBM intends to deliver this year 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 IBM plans to deliver called business rule beans.

Business rule beans will 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. Business rule beans enables developers to separate reusable abstractions within a business domain – and base these abstractions on policies, regulations or anything else that tends to change over time. Most rules engines are inference engines. This is not the case with business rule beans, which is actually a lightweight framework. The framework includes a number of basic business rules, while application developers can build others. You will be able to 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 intended to be offered as part of the business rule beans technology.

This solution will be more general purpose than what is provided by WebSphere Personalization (which primarily focuses on matching users with content). Also, this technology will be different from the Versata System – a rapid application development solution from Versata, a key IBM Business Partner. To begin, Versata rules are intended to hide the details of Java technology. Versata is a code generator. Business rule beans, on the other hand, is intended to supplement professional Java development work. Business rule beans will provide a useful facility for programmers to achieve reuse. Versata rules represent the application whereas business rule beans will supplement the application. Business rule beans also tends to be parameter-driven, meaning that their behavior can be dynamically defined through externally maintained data values, while this is perhaps less common in the Versata case. Finally, Versata rules become effective when the application is deployed. Business rule beans take effect at a designated point in time, which could be long after you install it.

### **Message beans and JMS listener**

Another requirement of Dynamic Corporation concerns the seamless blending of component and messaging technologies. The architecture illustrated 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 Web application for order entry, implemented in the J2EE platform and running on WebSphere Application Server, Advanced Edition, 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 phone 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 plans to address this requirement through an Enterprise Service that includes message beans and Java Message Service (JMS) listener technology. The technology represents an early version of the J2EE 1.3 message-driven beans and is virtually identical to the functionality outlined in the J2EE specification.

The initial support planned 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 will solve the common problem of managing the integration between inbound messages and Enterprise JavaBeans operations.



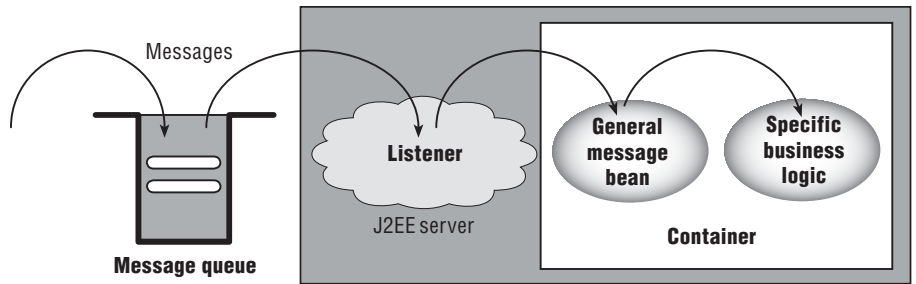


Figure 3. JMS listener and message bean support

Transactional support will also be 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, however. For example, developers still need to be skilled with JMS programming. A subsequent release of this Enterprise Service is planned to deliver even broader solutions that:

- *Eliminate the need for JMS programming expertise*
- *Gracefully 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 seven 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

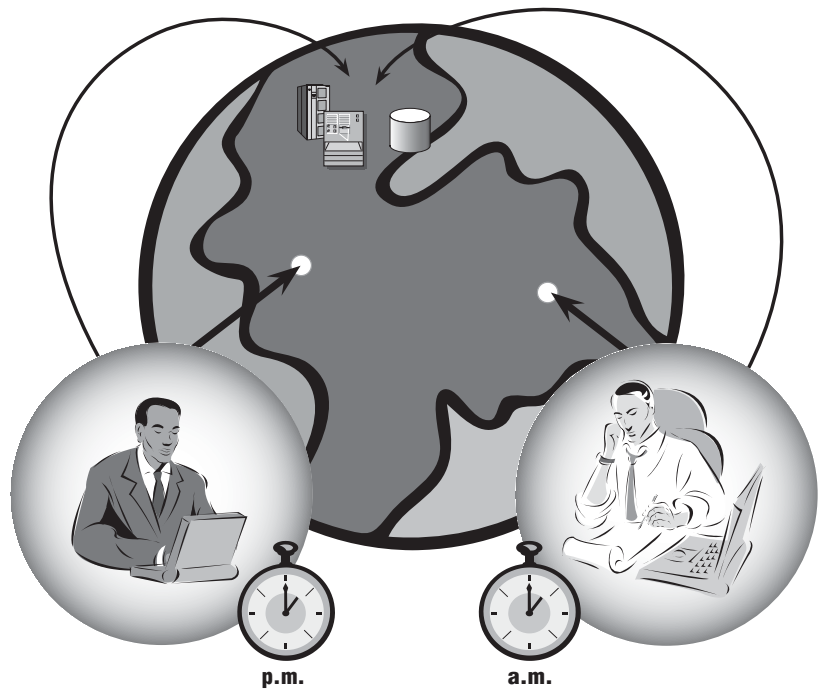


Figure 4. Internationalization requirements

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, as 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.

IBM intends to nonintrusively deliver on these requirements through the Enterprise Service for internationalization. It will provide simple application program interfaces (APIs) that server-side business objects can use to extract propagated information for localizing relevant computations. The point about being nonintrusive 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 Enterprise Service will provide an example of IBM ability to build upon established standards, in this case, to address end-to-end internationalization issues. In fact, IBM intends to submit a Java Specification Request (JSR) that addresses the enterprise requirement for internationalization.

How will this Enterprise Service relate to the WebSphere Translation Server? WebSphere Translation Server is associated with language translation of HyperText Markup Language (HTML), plain text and Lotus® Notes™ files. Internationalization will not do any translation. It only identifies locale and time zone information. So this means that a program using internationalization 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™), as well as Enterprise JavaBeans. 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 ability 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 quality-of-service descriptors for Dynamic Corporation customers.

IBM intends to deliver 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 intends to submit shared work area as a proposed JSR.

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 the first EJB. This EJB reads the information, and then control passes on to a second EJB that chooses to ignore the work area. A third EJB receives control, and this EJB once again reads the work area.

It is significant that work areas flow implicitly:

- *The second EJB 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 Enterprise JavaBeans 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 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 see the new value, and the original value does not resurface until control returns back to the point at which the new value was written.

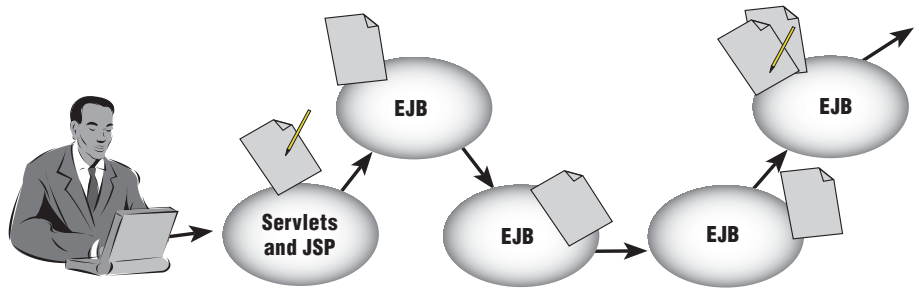


Figure 5. Transparent access to shared information through work areas

### **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 Enterprise JavaBeans (calling *in* to CORBA), as well as invoke Enterprise JavaBeans from these same CORBA applications (calling *out* from CORBA). Bidirectional CORBA connectivity intends to deliver another valuable service function that provides broad support for CORBA and EJB interoperability. This will enable customers with existing CORBA investments to incorporate new business value based on the J2EE model while continuing to leverage their existing systems. Beyond simple conformance to the J2EE 1.3 requirements for Remote Method Invocation-Internet Inter-ORB Protocol (RMI-IIOP), which IBM has supported for years, IBM has invested in extensive testing to ensure interoperability with popular third-party ORBs currently on the market.

CORBA connectivity support will consist of numerous samples, documentation (including restrictions) and a value-type library that simplifies calls from CORBA to EJB. The initially supported CORBA environments include 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 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; Windows® 2000 and Sun Solaris, Version 2.7.

Calls in from either servlets or EJB 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 could automatically enjoy the performance, stability, scalability and interoperability IBM will provide 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 environment in a standards-compliant way. How can the company do this?

One technique is to leverage CORBA technology. As an Enterprise Service, IBM intends to deliver 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 will deliver this CORBA solution as an Enterprise Service so that customers do not need to purchase a CORBA solution from other vendors. Customers could also choose to replace an existing CORBA server with the planned solution from IBM. (IBM intends to obtain CORBA 2.1 certification for the server, and support for later versions of CORBA is possible over time.) Figure 6 illustrates that you could use an IBM-based CORBA server behind or in front of the J2EE server. Customers can also leverage the SDK to build a C++ CORBA client, without needing to create a CORBA server at all. This would provide a simple and effective way to connect a C++ desktop application to the J2EE server, for example.

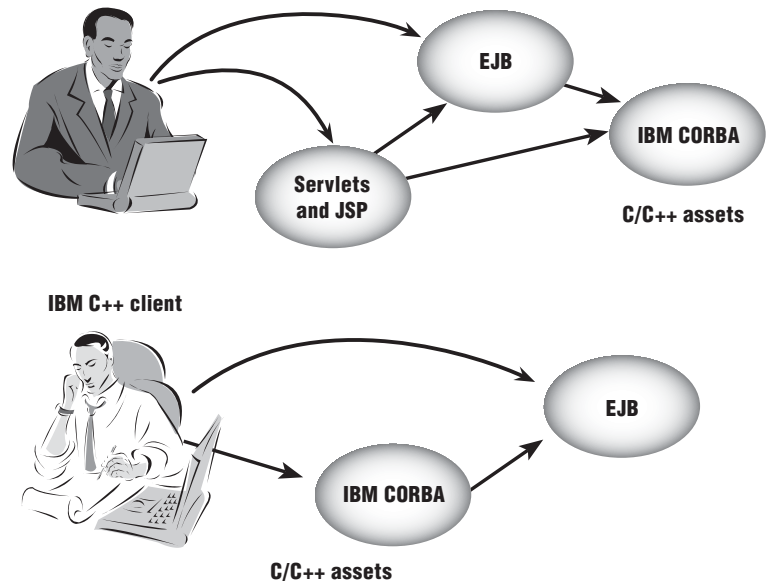


Figure 6. Using IBM-supplied C++ CORBA technology with J2EE technology

Consistent with the strategic emphasis on J2EE technology at IBM, the C++ CORBA solution will provide a basic CORBA environment. It can bootstrap into the J2EE name space, and it can invoke J2EE 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 environment as a service provider. The C++ CORBA technology will first be provided on Sun Solaris (Forte, Version 6.0 C++), IBM AIX (VisualAge® for C++, Version 5.0), and Windows NT and Windows 2000 (Visual C++, Version 6.0).

### **ActiveX bridge**

Looking back at the Dynamic Corporation topology, a requirement exists for J2EE 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 Enterprise JavaBeans be made accessible from the world of ActiveX.

IBM intends to deliver support for ActiveX integration with J2EE technology through an ActiveX bridge. The bridge would make Java classes dynamically available through ActiveX interfaces. This idea is depicted in Figure 7.

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. 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 code using Java Native Interface (JNI). The Java code invokes the services of either a J2EE client container (preferred

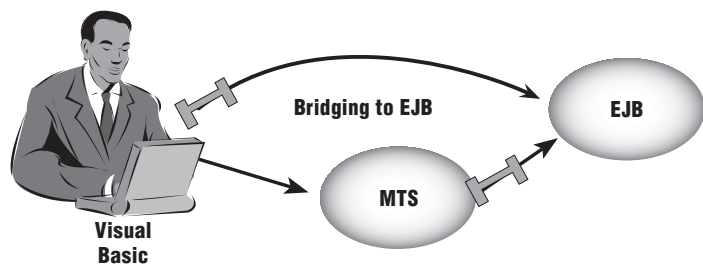
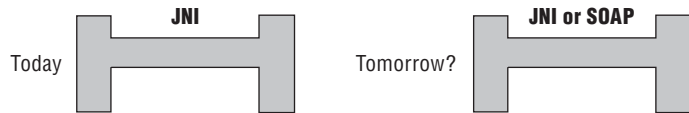


Figure 7. The ActiveX bridge provides a generic ActiveX-to-Java bridging solution.

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 will be available on both Windows 2000 and Windows NT, although this is in one direction only (as shown). 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 a higher quality 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, as SOAP may evolve to the point where it represents an attractive alternative transport, the current JNI-based bridge is flexible to enable the possible substitution of a SOAP-based implementation at some point in the future. Note that SOAP represents a possible additional transport mechanism rather than a replacement transport mechanism for JNI.



**So what's next?**

Remember that Enterprise Services deliver:

- *Additional functionality*
- *Higher qualities of service*

So what are some of the higher qualities of service and additional functions that IBM intends to deliver as future Enterprise Services? The release after WebSphere Application Server, Version 4.0, Enterprise Edition is anticipated to address the enterprise-class quality-of-service requirements described earlier in addition to augmenting the level of enterprise-class functionality.



***The Intelligent Corporation is an international enterprise that plans to build a globally accessible Web-based order fulfillment system. The corporation wants to reward it's 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 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 will deliver the Enterprise Services needed to solve the demands of the most sophisticated e-businesses.***

*Qualities of service*

- *Data caching and prefetching for optimized performance*
- *Support workloads requiring optimistic transactions*
- *Distributed transactional coordination of IBM CICS® and IBM IMS™*
- *Dynamic workload management*
- *High availability optimizations*

*Functions*

- *Workflow and business process support*
- *Increased levels of messaging integration*
- *Internationalization enhancements including bean-managed support*
- *Capabilities to express locking intent for higher throughput*

**Summary**

With the release of WebSphere Application Server, Enterprise Edition, Version 4.0, IBM will continue its tradition of high service to enterprise customers. WebSphere Application Server, Enterprise Edition will be supported by an evolving set of Enterprise Services that will deliver the functionality and qualities of service you 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 qualities of 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 2001

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

Produced in the United States of America  
07-01  
All Rights Reserved

AIX, the e-business logo, IBM, the IBM logo, MQSeries, OS/390, TXSeries, WebSphere and z/OS are trademarks or registered trademarks of International Business Machines in the United States, other countries or both.

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

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

Java, all Java-based trademarks and logos and Solaris 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.