BEA Systems, Inc.

Product Comparison: BEA WebLogic Server and IBM WebSphere Advanced

Why Organizations are Selecting BEA WebLogic over IBM WebSphere

BEA WebLogic versus IBM WebSphere

Introduction

Application servers have emerged as the principal technology for supporting e-commerce initiatives. These servers connect Web browser-based users to backend services such as relational database systems, packaged applications, and personalized content. They also provide transaction capabilities.

BEA WebLogic Server and IBM WebSphere Advanced Edition are frequently cited as leading application server offerings. Both products support the Java 2 Enterprise Edition (J2EE) platform specifications from Sun Microsystems. The J2EE standards offer a significant advance in enterprise software development. With J2EE, companies can develop multi-tier distributed applications that are independent of the underlying hardware, operating system, and database, and allow programs running on one server to be moved to a different server without change.

Companies using the J2EE platform benefit from shorter time to market and lower system development and administration costs. J2EE delivers unprecedented developer productivity by leveraging the Java programming language and reusability of EJB components. J2EE also provides standardization across platforms and critical enterprise application services defined in the J2EE standard, and enables developers to focus on writing business logic and addressing enterprise-critical issues.

This paper compares these two application server products and finds that BEA offers key advantages over IBM's offering in the following areas:

- Java 2 Enterprise Edition (J2EE) and Enterprise Java Bean (EJB) Support -- Each product was evaluated for compliance with accepted standards for applications programming, security, communications, and data access.
- Scalability -- Each product was evaluated for the features that enable scalability, clustering, applications threading, data caching, database connection pooling, and proprietary performance-enhancing techniques.
- Platform Support -- Each product was evaluated for the platforms on which it is supported.
- Performance -- Each product was evaluated for the ability to maintain response time and throughput in high volume environments.
- Data Connectivity -- Each product was evaluated for its capabilities for processing information residing in network-based data repositories (databases, office documents, and multi-media data). Special attention was given to each product's ability to process back-end database repositories.

- COM/COM+ Integration -- Each product was evaluated with respect to support of Microsoft's COM and COM+.
- HTTP Services -- Each product was evaluated for support of HTTP servers.
- Manageability -- Each product was evaluated for the tools and procedures provided for administering the server.
- Development tools -- Each product was evaluated for software tools included to support component developers.

About This Competitive Analysis

This Competitive Analysis compares the features and functionality of BEA WebLogic Server 4.5.1 against those of IBM's WebSphere Applications Server 3.0. It provides a candid and factual product based on hands-on analysis of the two products.

Competitive Overview

Competitive Products Overview

BEA WebLogic Server version 4.5.1 and IBM's WebSphere Server 3.0 Advanced Edition are both middle-tier applications servers. Both are designed to support Internet, intranet, and client/server systems built on architectures that encompass thin clients, middle tier components, and back-end information stores and legacy systems.

The products each represent an entry-level offering for hosting EJB components and other facets of the J2EE specifications. Both IBM and BEA market add-on products to extend the functionality and performance of these products.

BEA WebLogic Server

The award-winning BEA WebLogic Server provides the performance and scalability required to power e-commerce and Internet business solutions. BEA WebLogic Server provides an industrial-strength set of services for building e-commerce applications using J2EE standards, including EJB components. BEA WebLogic Server offers the most comprehensive implementation of J2EE Java standards in the industry.

As the industry leading e-commerce transaction platform, BEA WebLogic Server provides a number of features critical to developing and deploying mission-critical e-commerce applications across distributed, heterogeneous computing environments. These include:

- **Standards leadership** – Comprehensive Java 2 Enterprise Edition (J2EE) support, including EJB and JMS, to ease the implementation and deployment of application components.
- **Enterprise e-business scalability** – BEA WebLogic Server's software clustering of dynamic Web pages (servlets, Java Server Pages) and EJB business components, coupled with client connection sharing and database resource pooling, maximize efficiency in the use of critical resources.

- **Rich client options** – BEA WebLogic Server supports direct HTTP requests from browsers, HTTP servlets to drive dynamic HTML content, and Java Server Pages (JSPs) for managing the online update of dynamic Web pages.

- **Robust administration** – BEA WebLogic Server offers a comprehensive pure-Java console, Zero Administration Client (ZAC) for managing the distribution of applications to remote users, and dynamic application partitioning and cluster membership.

- **E-Commerce-ready security** – BEA WebLogic Server features Secure Sockets Layer (SSL) support for integration of encryption and authentication security into e-commerce solutions.

- **Maximum development and deployment flexibility** – BEA WebLogic Server features tight integration and support of leading databases, development tools, and other environments.

  Refer to Appendix A for more information on BEA WebLogic Server features.

IBM WebSphere Advanced Edition

IBM WebSphere Advanced Edition, V3.0 is IBM's entry-level production Java application server. Advanced Edition runs transactional business objects written to EJB 1.0 standard, as well as Java servlets and JavaServer Pages (JSP).

WebSphere Advanced Edition builds on IBM's WebSphere Standard Edition to provide portability and control of server-side business applications. IBM targets enterprise Internet, intranet, and e-commerce applications with features such as:.

- **Complete Java and Enterprise Java support** – WebSphere Advanced includes a server for applications built to the EJB specification. The focus is on medium- to high-level transactional environments used with dynamic Web content generation and Web-initiated transactions.

- **Apache-based HTTP server** -- WebSphere Advanced is packaged with an Apache-based HTTP server that includes enhancements for security and control. WebSphere Advanced includes support for other major Web servers as well**.**

- **Performance and scaling** -- WebSphere Advanced includes performance and scaling attributes with support for bean-managed and container-managed persistence, for entity beans and session beans.

- **Platform Support** -- WebSphere Advanced provides support for Intel and Unix-based platforms including Windows NT, Sun Solaris, and AIX on HTTP servers from IBM, Apache, Microsoft, and Netscape.

- **Tivoli Ready** -- WebSphere Advanced contains Tivoli Ready modules that can be managed by Tivoli -based tools.

Features Comparison

Support for Standards

Java Standards

Before the emergence of the Java 2 Platform, Enterprise Edition from Sun Microsystems, the industry lacked a set of standards for building enterprise Web applications. J2EE has become the de facto standard for enterprise application development in Java. Each of the J2EE specifications has as specification compliance.

The J2EE application environment consists of the following:

- Application Components including application clients, applets, servlets (and Java Server Pages – JSP), and EJBs.
- Containers that provide the runtime support for application components and include the application client container, the applet container, the Web container, and the EJB container.
- Databases that provide the storage of business data and other information related to the J2EE environment. Databases are accessed from the Web, EJB, and application client components.
- Standard Services such as communication protocols services (HTTP, HTTPS, and RMI/IIOP), transaction services (JTA), CORBA interoperability services (JavaIDL), database access services (JDBC), naming and directory services (JNDI), messaging services (JMS), and electronic mail services.

IBM WebSphere Advanced Edition does not provide support for the JDK 1.2 required by the J2EE platform. Instead, IBM continues to extend its JDK 1.1-based technologies with elements of Java 2 Enterprise Edition (J2EE). Furthermore, IBM recommends using with WebSphere its version of the JVM, which is at JDK level 1.1.8, a further warning sign that IBM is not ready to support the latest standards. WebSphere has confirmed problems with the Sun 1.2 JDK.

BEA WebLogic Server has been tested to run under either JDKs. The Windows NT InstallShield distribution, for example, includes both Java Runtime Environments (JREs), offering the flexibility to switch environments by simply clicking an icon.

Java 2 Enterprise Edition Standard Support

The cornerstone of the J2EE platform is the Enterprise JavaBeans (EJB) specification. BEA WebLogic was early on the EJB scene, offering one of the only EJB servers on the market in 1998 when version 1.0 of the EJB specification was released.

Like any version “1.0 technology,” EJB had its limitations. Many of the problems with EJB 1.0 have been addressed in EJB 1.1. As one of the first EJB servers on the market, BEA

WebLogic has a gained a clear advantage via its early adoption, which is evident in the robustness, reliability, and performance of the product.

> For an EJB solution, BEA is in the lead today in terms of production application implementations and the degree to which evidence exists to support claims of enterprise-class performance, scalability, reliability and availability.
>
> Giga Information Group

BEA WebLogic and WebSphere support both session and entity bean types as well as EJB persistence using both EJB container-managed persistence (CMP) and EJB bean-managed persistence (BMP). WebSphere supports CMP on Oracle and DB2 only.

Beyond the support of EJB, the Sun J2EE specifications define services that support components, including Java Database Connectivity (JDBC) for moving database data to and from components. Here BEA WebLogic ships with JDBC drivers for Oracle, Sybase, Informix, and Microsoft SQL Server, and supports any JDBC driver for any JDBC-compliant database. The presence of native drivers distinguishes BEA WebLogic from WebSphere, which relies on generic JDBC or third-party drivers. BEA WebLogic supports database vendor-provided JDBC drivers running with the Java 2 SDK, v1.2.1. BEA WebLogic JDBC drivers support the new JDBC 2.0 APIs in the BEA WebLogic Server 5.1 Beta currently available on BEA's web site and due for General Availability late March.

BEA's implementation of the JavaSoft's J2EE Remote Method Invocation (RMI) specification goes beyond that of WebSphere as well. RMI is used for accessing components outside of the middle tier. BEA WebLogic's implementation of the RMI specification provides standards-based distributed object computing and far greater scalability and performance, as well as access to all of BEA WebLogic.'s integrated services, such as JDBC Events, etc. BEA WebLogic RMI also addresses some of the limitations of the JavaSoft RMI compiler and is fully integrated with BEA WebLogic's Java Naming and Directory Interface (JNDI).

BEA WebLogic supports a complete implementation of the J2EE Java Message Service (JMS) which is used for implementing message-based applications. Java Message Service (JMS) allows Java programs to exchange messages with other Java programs sharing a messaging system. WebSphere does not support JMS. IBM supports JMS in its MQSeries product, which requires installation and application development and deployment on a separate product platform.

BEA WebLogic Server uses the standard J2EE Java Naming and Directory Interface (JNDI) to publish objects with location transparency. Using Java-standard BEA WebLogic Server can access existing directory services like Novell NDS, Sun NIS+, Microsoft Active Directory, or others supporting the Internet-standard Lightweight Directory Access Protocol (LDAP).

> Our research indicates that, unless IBM wraps WebSphere with significant service (e.g., professional services/systems engineers) capabilities, user organizations will continue to struggle implementing the the Advanced (Enterprise JavaBeans) and Enterprise (CORBA) Application Server Editions.

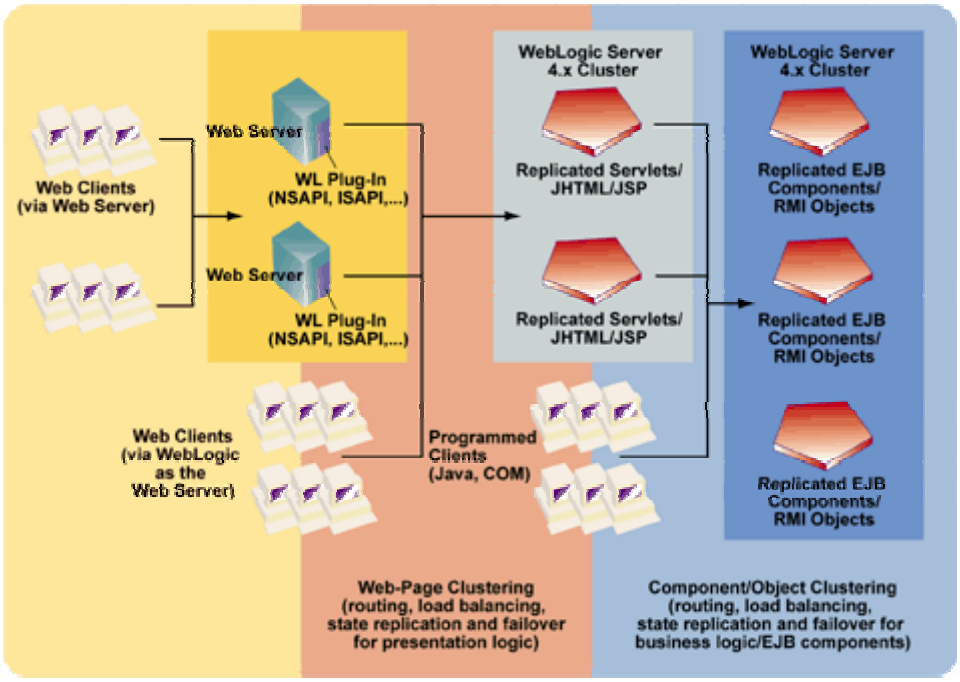
Bottom Line: Users investing in WebSphere must require IBM to also provide the qualified personnel resources necessary for successful implementation (component deployment, application server configuration, and operations) or face delays in successful application deployment.

Morgan Gerhart, Meta Group 02/14/2000

Scalability and Availabilty

Clustering and Load Balancing

BEA WebLogic Server provides the highest levels of scalability, expandability, and high-availability through clustering. BEA WebLogic Server provides both Web-page and EJB component clustering – without any special-purpose hardware or operating system services. Web-page clustering handles transparent replication, load-balancing, and failover for the presentation logic that generates responses to Web clients (for example, the contents of a Web shopping cart). Component clustering handles the complexities of replication, load balancing, and failover for EJBs, or business logic, and recovery of stateful objects like EJB entity beans.



Three configurations of clustered BEA WebLogic Servers are supported. All three configurations use the same basic structure: a series of BEA WebLogic Servers all accessible over the same LAN, and accessible to a set of clients — browser servlet clients and Java application clients — over one of BEA WebLogic's various protocols. The three configurations supported are:

Simple: a three-tier cluster that dispenses servlets. *Servlet clustering* provides load balancing or load balancing plus failover for servlet sessions.

Moderate: Three-tier cluster that serves objects. This type of cluster is the most typical scenario for "thicker" object-based client applications, with Java/VB/C++ code rather than HTML in the client. This scenario is a likely configuration for load balancing and failover of services across a set of BEA WebLogic Servers. Each server handles client communications as well as the execution of requests. The three-tier cluster also provides the ability to "route" applet requests between the server with which the applet is communicating and the server hosting the service object (impl) that the applet wants to use.

Complex: a three-tier cluster of mixed objects and servlets, such as a mixture of HTML browser servlets and EJBs. One of BEA WebLogic's strengths is its integration of standard APIs to support all of the enterprise services (including RMI, EJB, JDBC, and HTTP servlets) in a single application server framework.

BEA WebLogic Server's EJB clustering goes a step further than WebSphere and maintains a transparent connection for recovery of stateful EJB objects even when the server goes down. With BEA WebLogic, the request can failover to another server in the cluster and any in-flight transactions can be rolled back immediately. The ultimate benefit of this is that an EJB will not fail.

The result, continuing the shopping cart example, is that if the server fails when the user gets to the "check out," BEA WebLogic's transparent replication and failover facilities ensure that another application server in the cluster will fulfill the request. BEA WebLogic's load balancing ensures that the server is not overwhelmed — and will balance the high demand across multiple servers.

BEA offers a variety of load-balancing algorithms to ensure that the load is truly balanced, as opposed to WebSphere's round- robin load-balancing, which merely shuffles the requests rather than analyzing where best send them. Further, although WebSphere supports session clusters, as with most functions, WebSphere's failover of these session clusters is maintained via a database rather than in-memory persistence. This is costly from a performance perspective.

BEA WebLogic's clustering technology has proven to be the most scaleable and fault resilent application server architecture available. There are hundreds of BEA WebLogic Server customers processing large transactions volumes every day. And, when business requirements dictate, customers can easily add more servers to handle additional loads.

Both Web-page and component clustering are critical for delivering scalability and availability of e-commerce solutions. BEA WebLogic Server's software clustering of servlets, Java Server Pages and EJB business components, coupled with client connection sharing and database resource pooling, provides maximum efficiency in the use of critical server and network resources. These are key differenciators between BEA WebLogic Server and WebSphere Advanced.

Replication

Replication provides data mirroring and/or state replication from one server to another and is essential for all clustering architectures. Replication facilitates both scalability and availability. In BEA WebLogic Server, distributing workload for stateless services may be based on a variety of factors, including server load, server characteristics, application data partitioning, time of day, and request priority.

BEA WebLogic uses a JNDI tree for its naming services and supports the replication of EJB servers, making it easy to clone EJB servers across multiple nodes, improving availability. In the standard implementation of replication services, BEA WebLogic has a much more sophisticated approach than that of WebSphere. BEA WebLogic's JDBC session persistence and in-memory state replication feature adds to a comprehensive availability solution. WebSphere supports something similar through cloning, but manages session persistence via a database

WebSphere provides rudimentary replication services and requires optional products such as its WebSphere Performance Pack at $7599/server to provide capabilities to support high-volume Web sites.

Database Connection Pooling and Caching

BEA WebLogic Server shares database connections so that the maximum number of simultaneous clients can be supported. BEA WebLogic Server caches database query results improving performance since the client's use of data is independent of the DBMS's supplying of data. The client can view the first record immediately, and while the client is viewing the first record, the retrieval of DBMS data can be cached on the server. BEA WebLogic Server also caches database query results and automatically updates cached data in real time.

A BEA WebLogic JDBC client has its own workspace within the BEA WebLogic Server that can be saved between sessions. A client can log back in to its previous state — including a set of cached connections to the DBMS — to continue work. Within the client workspace, any arbitrary object (for example, a Dataset) can be saved for long-term use, thus reducing DBMS workload and network traffic.

BEA WebLogic's database connectivity is superior to WebSphere's in terms of database support, the queue-based protocols, and saved client workspaces.

Platform Support

BEA WebLogic supports a far greater number of server platform environments than WebSphere. Below is a comprehensive list of vendor supported platforms for both products.

Platform	**BEA WebLogic 4.5.1**	**WebSphere 3.0.2**
Compaq Alpha with Tru64 UNIX	Y	N
IBM AS/400e with OS/400 V4R3	Y	N
IBM AS/400e with OS/400 V4R4	Y	N
IBM S/390 with OS/390 V2R2	Y	N
IBM RS/6000 with AIX 4.2	Y	N
IBM RS/6000 with AIX 4.3	Y	Y
Intel Pentium with NT 4.0	Y	Y
Intel Pentium with Windows 98	Y	N
Intel Pentium with Red Hat Linux	Y	currently testing
Intel Pentium with Windows 2000	currently testing	currently testing
Intel Pentium with Netware	N	Y
HP/9000 with HP-UX 11.0	Y	N
Siemens MIPS with Reliant UNIX 5.44C	Y	N
SGI with IRIX	Y	N

Sun SPARC with Solaris	Y	Y
IBM's JVM for NT	Currently testing	Y
Sequent Dynix	Y	N
Open VMS	Y	N

Performance

Performance generally refers to how fast an application server responds to clients, and how much load can it handle. It is an important metric, as it allows one product to handle the load on cheaper hardware than a competitor, thus increasing that product's value proportionately. Although both products are Java-based and thus reliant on the JVM they run, we found very distinct differences in the way each product approaches performance.

BEA WebLogic Server scales by carefully managing scarce resources like threads and socket connections. With BEA WebLogic Server, a single client/server connection is shared across all bidirectional communications, regardless of the request type and the number of remote objects being accessed. The BEA WebLogic Server internal architecture is highly scalable and is based upon the request queue/dispatch model found in the highest performing transaction systems. Database connections are shared so that the maximum number of simultaneous clients can be supported. BEA WebLogic Server caches database query results and can automatically update cached data in realtime as changes are made to the backing database.

Both WebSphere and BEA WebLogic are multithreaded, but BEA WebLogic allows the user to specify the number of threads spawned by the process. For some platforms, BEA WebLogic Server provides Performance Packs — native libraries that substitute platform-optimized socket multiplexers. Further, BEA WebLogic has made a number of performance enhancements to further improve response time and throughput.

For example, connections are multiplexed in BEA WebLogic's RMI for scalability and speed. BEA WebLogic Server's RMI implementation is more scalable because it allows a single client/server connection to be shared over many remote object communications. It also enables the clustering of RMI across multiple servers. Further, BEA WebLogic's serialization was re-written to be more efficient than the implementation in the JDK while maintaining standards compliance. To ensure that each release will meet enterprise high-performance requirements, BEA stress tests each release with 10,000 concurrent clients.

Based on comparison tests, WebSphere's memory footprint is more than double the size of BEA WebLogic's. A running BEA WebLogic server with no clients (15 threads) used about 35-45 Megabytes of memory, while WebSphere with no clients used 100-125 Megabytes. WebSphere's CPU utilization was also substantially higher.

Database Connectivity

As databases hold a great deal of a company's information, connecting to a database is a necessity for any application server. Both products support the JDBC specification so theoretically they should be able to connect to any JDBC data source; however, WebSphere supports only IBM DB2 and Oracle 8.0.5. In theory, customers should be able to plug in any JDBC driver for the perspective database, but IBM will not support WebSphere with any other databases because of potential conflicts with these drivers.

To compensate for this limitation, IBM bundles DB2 Workgroup Edition with WebSphere Application Server. The DB2 database provided is an older version (5.2) and has a limited-use license.

BEA WebLogic Server ships with its own native high-performance JDBC drivers for leading DBMS products such as Oracle, Sybase, Informix and Microsoft SQL Server. BEA WebLogic also supports DB2 using the IBM driver. BEA WebLogic Server also works with any third-party JDBC driver and ships with a generic driver for any JDBC-compliant database. BEA WebLogic Server's multitier JDBC implementation allows a Java application to access and update databases from anywhere on the network.

WebSphere ships only with a generic driver. The presence of native drivers — as opposed to a generic driver — and the wide database support distinguish BEA WebLogic from WebSphere. Further, WebSphere supports only an internal solution for container managed persistence (CMP) on Oracle and DB2. WebSphere cannot support more substantial CMP functionality on a different database. BEA WebLogic Server supports CMP independent of database. In addition, BEA WebLogic supports third-party plug-ins for CMP including Versant, TOPLink, and ODI.

It's also important to understand that database connectivity is especially important to WebSphere because it uses a database for its administrative repository and cannot run without a live connection to a database. This necessitates running the database on the same computer as WebSphere because constant calls to a remote computer would be too expensive. In this case, the local database would most likely be used solely for WebSphere's administrative repository.

HTTP Services

Both BEA WebLogic Server and WebSphere Advanced Edition include their own HTTP Server. BEA WebLogic integrates a simple HTTP daemon and can be used as the primary Web server. IBM's HTTP server is a separate product packaged with WebSphere Advanced and requires separate installation, system resources, and mangement. Both products offer support for other major Web servers. BEA WebLogic works with all major Web servers, including Apache HTTP Server, Netscape Enterprise Server, Microsoft Internet Information Server (IIS), and Domino Go for AS/400. BEA WebLogic Server also provides plug-ins for Microsoft IIS and Netscape Enterprise Server, and has a plug-in for Apache in the current Beta of BEA WebLogic Server 5.1 available on BEA's web site (www.bea.com). BEA WebLogic Server's plug-ins provide awareness of BEA WebLogic features such as load balancing and servlet session state replication to the HTTP Server.

IBM provides support for these HTTP Servers in addition to a back release of the Domino Go WebServer, which doesn't hold serious marketshare or add a significant competitive advantage and is due primariliy to the fact that IBM owns Lotus.

COM/COM+ Support

Through its support for the Microsoft SDK for Java, BEA WebLogic supports COM+ objects by wrapping these objects in Java and transparently sharing them over the network. Similarly, BEA WebLogic supports COM+ bindings for Java and EJB objects. BEA WebLogic also supports COM clients and can export Java objects as COM objects. The COM compiler can convert COM objects into RMI-usable objects. BEA WebLogic Server also provides for the automatic wrapping of Java objects and EJB components with

a COM+ binding for easy invocation from Microsoft-based applications (Visual Basic, PowerBuilder, Active Server Pages, etc.)

WebSphere does not support COM objects directly, but it is possible through third party Java-COM or CORBA-COM bridging products. The ability to use COM is extremely useful when deploying an application with COM clients or when using ASP technology on the Web server. WebSphere limits its users in this respect.

Component Development and Deployment

With respect to EJB component development and deployment, IBM emphasizes its close integration with the IBM VisualAge Family and the ObjectBuilder toolkit for development in WebSphere Advanced.

For application development for BEA WebLogic Server, BEA has established strong partnerships and ties with best-of-breed tool vendors so that customers can take advantage of the full range of Java-compatible development tools of their choic. BEA's partnerships and relationships with high quality tools vendors make it easier to build comprehensive e-commerce applications and support the whole product development lifecycle. BEA works closely with many tools vendors to facilitate tight integration and availability between our product lines, delivering better ease of use, enhanced developer productivity. Current examples of these integrations include:

- VisualCafé Enterprise Suite -- an industry-leading IDE that offers a complete solution for rapidly developing and debugging Java and EJB applications for the Web. Version 3.1 of the VisualCafé Enterprise Suite features enhanced integration between VisualCafé and BEA WebLogic.

- IBM's VisualAge for Java -- a leading Java IDE with support for building and testing Java applets, servlets, and EJB components. BEA offers the VisualAge Integration Kit which provides integration between VisualAge and BEA WebLogic Server.

- COOL:Joe is a second-generation IDE that is seamlessly integrated with an architecture and design modeling capability to allow developers to architect, create, deploy, and manage reliable and scalable enterprise applications using Enterprise JavaBeans. COOL:Joe’s Deployment Wizard currently supports “auto deployment” to BEA WebLogic and is designed to work seamlessly with BEA WebLogic Server.

- TOPLink is a database integration tool that provides a fully integrated, advanced object to relational container managed persistence (CMP) solution for BEA WebLogic Server. TOPLink for BEA WebLogic Server adds value beyond the EJB spec and the base feature set of BEA WebLogic Server by providing many CMP features.

- StructureBuilder is an EJB construction and deployment tool that contains visual modeling tools for UML model-based development of Java components and applications. StructureBuilder is fully integrated with VisualCafé.

- InLine Standard is an EJB development tool that automates the entire EJB development process. With InLine Standard, developers can work through a user-friendly interface to create either generic EJBs or custom beans for the BEA WebLogic Server.

- Classic Blend — Next Generation is a presentation server that distributes the visual GUI-rendering portion of an application. Extensive documentation, including information and sample code for integrating with BEA WebLogic Server

- JProbe ServerSide Edition is an integrated product suite that provides profiling and performance-tuning tools for Java applications. JProbe's Server Launch Pad gives you point-and-click integration with BEA WebLogic Server.

> …one can achieve roughly the same results with independent Java IDEs combined with BEA WebLogic as one can with VisualAge and WebSphere…
>
> Mike Gilpin, GIGA Information Group

BEA believes that most customer need to choose from the wide selection of development tools available today and that via our close integration with these tools, they can and have achieved more with BEA WebLogic and independent tools.

Systems Administration and Operation

BEA WebLogic Server offers robust administration and management capabilities with a comprehensive pure-Java management console, the Zero Administration Client for managing the distribution of applications to remote users, and dynamic application partitioning and cluster membership. BEA WebLogic has a number of Systems Administration and Operation features that provide an edge over WebSphere.

For example, the Zero Administration Client (ZAC) supports automatic distribution of Java applets, applications, or systems. With ZAC, program libraries - even a new BEA WebLogic Server release - can be installed centrally by an administrator. BEA WebLogic Server pushes each updated component to all appropriate clients.

BEA WebLogic Server also supports dynamic deployment, undeployment and redeployment of EJBs. Specifically, if a bean's interface and semantics do not change, BEA WebLogic supports reployment "on-the-fly", completely transparent to clients of the bean. (If the interface and semantics change, the bean must be first undeployed (thereby invalidating any client references to the bean), prior to redeployment of the new bean.)

BEA WebLogic Server can share the userids and passwords/certificates stored in Netscape Directory Server and the Microsft LDAP LDAP-compliant directories for user authentication. BEA WebLogic Server also features e-commerce-ready security with Secure Sockets Layer (SSL) support for integration of encryption and authentication security into e-commerce solutions. Because of its implementation of the J2EE standard, Weblogic fully supports the Java ACL security model. It also supports Win NT domain security ACLs through a tool called NTRealm.

In evaluating the products, WebSphere was found to be extremely problematic and very difficult to install. WebSphere will not work without a functional web server, a database connection, and IBM's JDK. (There are documented problems encountered using WebSphere with Sun's 1.2 JDK.) The WebSphere CD includes IBM HTTP server (based

on Apache), but the installation program will not automatically install it unless web services have already been installed.

Conclusions

In summary, BEA WebLogic Server leads the market for application server technology today given its comprehensive, standards-based implementation. No application server on the market supports enterprise application requirements better than BEA WebLogic Server. With advanced support for Java Server Pages (JSP), Java Messaging Services, in-memory servlets session-state management, Enterprise Java Beans (EJB), and clustering technology, BEA WebLogic Server goes above and beyond WebSphere in terms of Java and J2EE standards, features, and reliability — particularly in areas of great importance to enterprises.

> For an EJB solution, BEA is in the lead today in terms of production application implementations and the degree to which evidence exists to support claims of enterprise-class performance, scalability, reliability and availability.
>
> Mike Gilpin, GIGA Information Group

Appendix A: BEA WebLogic Server Feature/Benefit Matrix

Feature	Benefit
Enterprise JavaBeans (EJB)	Makes it easy to encapsulate business logic as secure, transactional components. The BEA BEA WebLogic Server provides a complete implementation of Enterprise Java Standards, including Enterprise JavaBeans (EJB) support.
Web-page and component clustering of EJBs across multiple servers	Provides the highest levels of scalability, expandability, and high-availability. Web-page clustering handles transparent replication, load-balancing, and failover for presentation logic. Component clustering handles the complexities of replication, load balancing, and failover for EJBs (business logic). Both Web-page and component clustering are critical for delivering scalability and availability for e-commerce solutions.
Wide Integrated Development Environments (IDEs) support	Provides greater choice in application development. BEA WebLogic Server extends leading Java IDEs to support the development and debugging of Java server applications. Choose from a number of tools that enable you to drag-and-drop JavaBeans to build BEA WebLogic Server applications.
Web publishing tools & Java Server Pages (JSP)	Provides easy development and deployment of dynamic Web content. Java Server Pages can be used with personalization, database access, and transactional EJBs to develop all kinds of reliable, high performance, Web applications.
Scalable Remote Method Invocation (RMI)	Provides scalable distributed object support. With RMI, an application can use distributed objects as easily as local objects. BEA WebLogic Server's RMI implementation is scalable because it allows a single client/server connection to be shared over many remote object communications. It also enables the clustering of RMI across multiple servers.
Java Messaging Service (JMS)	Provides "store and forward" or "point to point" messaging with transactionally-guaranteed delivery. JMS also provides a "publish/subscribe" event management model JMS for applications requiring near-real time information about changing conditions. BEA WebLogic Server JMS is compatible with third-party messaging solutions that support the JMS standard.
Multitier JDBC	Allows a Java application to access and update databases from anywhere on the network. BEA WebLogic Server includes its own high-performance, native JDBC drivers for leading DBMS products, and also works with any third-party JDBC driver.
WebLogic COM	Provides interoperability with Microsoft COM+ objects. The BEA WebLogic Server allows Microsoft COM+ objects to be plugged into the BEA WebLogic Server framework, automatically wrapped with a Java class, and transparently shared over the network. BEA WebLogic Server also provides for the automatic wrapping of Java objects and EJB components with a COM+ binding for easy invocation from Microsoft-based applications (Visual Basic, PowerBuilder, Active Server Pages, etc.).
CORBA & OLTP support	Extend existing application server technologies to Java and the Web. BEA WebLogic Server provides high-performance integration with BEA Tuxedo (on-line transaction processing) and BEA CORBA technology (BEA WebLogic Enterprise). JavaIDL can be used to generate client bindings to remote CORBA objects using the standard Java Developer's Kit ORB.
Global naming	Provides access to existing directory services like Novell NDS, Sun NIS+, Microsoft Active Directory, or others supporting the Internet-standard Lightweight Directory Access Protocol (LDAP).
Distributed transaction support	Safeguards business-critical applications and corporate data so that integrity cannot be compromised, even as transactional components (e.g., EJB and JMS) span multiple machines. Support for the Java-standard Java Transaction API (JTA) allows clients or servers to initiate transactions that are propagated to other

	servers.
RSA Security and firewall support	Secures networked applications with optional encryption, authentication, and authorization based on the RSA Secured Sockets Layer (SSL), X.509 certificates, and access control lists (ACLs). All of BEA WebLogic Server's services are securely available through firewalls via tunneling through HTTP or the SSL-variant of HTTP (HTTPS).
Standard Internet protocols	Provides access for Web browsers to BEA WebLogic Server directly via normal HTTP requests. Forwarding capabilities, such as HTTP proxying, enable dispatching to servers other than the original Web server. For higher performance, HTTP connections are maintained across requests. All BEA WebLogic Server services are also accessible directly through TCP/sockets.
Graphical management console	Offers a comprehensive pure-Java console for remotely monitoring and updating the state of BEA WebLogic Server applications and clusters. Multiple clients and servers can be securely and easily managed from a single remote console.
Zero Administration Client (ZAC)	Supports the automatic distribution of Java applets, applications, or systems. With ZAC, program libraries – even a new BEA WebLogic Server release – can be installed centrally by an administrator. BEA WebLogic Server pushes each updated component to all appropriate clients. The ZAC client itself has a very small footprint.
Integrated logging	Logs diagnostic and security audit information automatically and provides interfaces for applications to log their own exception conditions. Optionally, HTTP traffic can be logged in standard common log and extended log formats. Logs can be viewed remotely from a web browser or from the management console.
Dynamic application partitioning and cluster membership	Permits online application components to be dynamically relocated across machines. EJBs, Java Server Pages, and servlets are loaded, or reloaded, dynamically. BEA WebLogic Servers also dynamically join and leave an active cluster.
Database support	Provides out-of-the-box, native Java Database Connectivity to: Oracle Informix Sybase MS SQL-Server
Platform support	Provides wide choice of development and deployment options. BEA WebLogic Server is certified on the following platforms: Windows NT Sun Solaris HP-UX IBM AIX Compaq Tru64 Unix Siemens Reliant Unix SGI Irix IBM OS/400 Linux