

# BEA WebLogic® Server 6.0 Reviewer's Guide



## Restricted Rights Legend

This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks

BEA, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc. All other company names may be trademarks of the respective companies with which they are associated



BEA Systems, Inc.  
2315 North First Street  
San Jose, CA 95131 U.S.A.  
Telephone: +1.408.570.8000  
Facsimile: +1.408.570.8901

[www.bea.com](http://www.bea.com)

# Contents

**Introduction 1**

**Reviewer Contact Information 1**

**Getting Started 2**

**Hardware/System Platform Support 4**

**BEA WebLogic Server Features 5**

- BEA WebLogic Server Architecture 5
- BEA WebLogic Server as a Web Server 6
  - Virtual Hosting 7*
  - Using Proxy Server Configurations 7*
  - Load Balancing 7*
  - Failover 8*
- Security Services 8
  - Authentication 8*
  - Authorization 8*
  - Alternative and Custom Realms 9*
  - Encryption 9*
- BEA WebLogic Clusters 10
  - Benefits of Using Clusters 10*
  - Cluster Architecture 10*
  - Defining a WebLogic Cluster in a Network 11*
  - How BEA WebLogic Servers in a Cluster Communicate 11*
  - Clustered Services 12*
- Server Management and Monitoring 13
  - Administration Console 13*
- BEA WebLogic jDrivers 14
  - BEA WebLogic jDriver for Oracle 15*
  - BEA WebLogic jDriver for Microsoft SQL Server 15*
  - BEA WebLogic jDriver for Informix 15*
- Integrated Development Tools 16
- New Features in Version 6.0 16
  - Enterprise Messaging 16*
  - Enhanced XML Support 17*
  - Enterprise JavaBeans 2.0 17*
  - Distributed Transaction Management 18*
  - Web-based System Management 18*
  - J2EE Certification 19*
  - Web and Object Clustering Enhancements 19*
  - Web Server Enhancements 20*
  - Security Enhancements 20*
  - Internationalization 20*

**BEA WebLogic E-Business Platform™ 21**

**About BEA 22**



# Introduction

Thank you for taking the time to review BEA WebLogic Server 6.0. This guide provides an overview of the core server features and new features included in the latest release, as well as other information that will provide you with more background information on the product.

## Reviewer Contact Information

<b>Product</b>	<b>BEA WebLogic Server™ 6.0</b>
<b>Web site</b>	<a href="http://www.bea.com">www.bea.com</a>
<b>Address</b>	BEA Systems, Inc., 2315 North First Street, San Jose, CA 95131 USA
<b>Customer Relations</b>	Use the above address or call 1-800-817-4BEA in the U.S., +1-408-570-8000 worldwide.
<b>Download</b>	<a href="http://commerce.bea.com">http://commerce.bea.com</a>
<b>BEA WebLogic Server Documentation</b>	<a href="http://e-docs.bea.com/wls/docs60">http://e-docs.bea.com/wls/docs60</a>
<b>Developer Center</b>	<a href="http://developer.bea.com">http://developer.bea.com</a>
<b>BEA EvalNET</b>	<a href="http://www.bea.com/evalnet/index.html">http://www.bea.com/evalnet/index.html</a>
<b>Press Contact</b>	Christina Grenier, 408-570-8330, <a href="mailto:cgrenier@bea.com">cgrenier@bea.com</a>

# Getting Started

Below are some pointers to resources that will help you get started on a successful evaluation of BEA WebLogic Server.

## *Obtaining BEA WebLogic Server 6.0*

BEA WebLogic Server 6.0 is distributed via a public download or physical shipment. To download the product, go to <http://commerce.bea.com> and follow the links to the BEA WebLogic Server 6.0. If you would like to obtain a physical shipment, contact Christina Grenier at 408-570-8330 or [cgrenier@bea.com](mailto:cgrenier@bea.com).

## *Licensing*

BEA WebLogic Server 6.0 downloads come pre-configured with a 30-day evaluation license. If you would like an extended license, please contact Christina Grenier at 408-570-8330.

## *Service Packs*

Service Packs are released on a regular basis. They include a roll-up of all bug fixes and minor functional enhancements. After you obtain the license key and install BEA WebLogic Server, please go to [http://commerce.bea.com/downloads/weblogic\\_server.jsp - wls](http://commerce.bea.com/downloads/weblogic_server.jsp-wls) or follow the “Featured Downloads” link at [www.bea.com](http://www.bea.com) to download the latest Service Pack. Each Service Pack is a superset of the prior release, so Service Pack 2 includes Service Pack 1. You only need to apply the latest Service Pack.

## *Enterprise JavaBeans (EJB) 2.0*

The EJB 2.0 container for BEA WebLogic Server is now available. The EJB 2.0 container needs to be temporarily installed separately due to Java licensing. When JDK 1.3 is complete, EJB 2.0 will be distributed as part of the server.

## *BEA Developer Center*

The BEA Developer Center was created to accelerate development of e-business applications built using Java technology by providing the developer community with extensive developer resources.

The BEA Developer Center can be found at <http://developer.bea.com>

### *Sample Code*

BEA WebLogic Server 6.0 includes the Java Pet Store Application — a fully functional sample application, with source code and documentation, modeling a complete e-commerce application. To launch the Pet Store Demo, simply select *Run Pet Store Demo* from the *Start* menu.

A large number of code examples are also shipped with the product. Each API is fully documented with sample code that can be deployed to the server. To utilize the code examples select *Start Examples Server* from the *Start* menu.

### *Support*

To ensure your success in reviewing BEA WebLogic Server, BEA Customer Support has created BEA EvalNET. BEA EvalNET provides Support Newsgroups, online technical resources, and access to BEA's team of Developer Relations Engineers. This team of knowledgeable BEA engineers is dedicated to resolving technical issues related to evaluation software. Please register for 30 days of free evaluation support at <http://www.bea.com/evalnet/index.html>.

### *Documentation*

All documentation for the BEA WebLogic Server is accessible online at <http://e-docs.bea.com/wls/docs60>. Together, the Release Notes, Introduction, Administration, Programming, Platform Matrix, and other resources provide comprehensive documentation for BEA WebLogic Server.

# Hardware/System Platform Support

Platform certification is an ongoing process. For the latest certified hardware/operating system platforms see:

<http://e-docs.bea.com/wls/platforms/index.html>

Compaq Alpha with Tru64 UNIX

Compaq OpenVMS

Compaq Tandem NSK

Hewlett-Packard HP/9000 with HP-UX 11.0 and 11i

IBM AS/400e with OS/400 V4R4/V4R5

IBM DYNIX/ptx

Bull/IBM RS/6000 with AIX 4.3

IBM S/390 with OS/390 V2R6

Intel Pentium with Windows 2000

Intel Pentium with Windows NT 4.0

Intel Pentium with Red Hat Linux

SCO Unixware

Siemens MIPS with Reliant UNIX 5.44C

Silicon Graphics with IRIX

Sun Microsystems SPARC with Solaris

Sun Microsystems SPARC with Solaris 8

# BEA WebLogic Server Features

BEA WebLogic Server 6.0 decreases time-to-market, boosts developer productivity, saves substantial development costs, and eases integration challenges by combining — in one server — the capabilities of a Java-based application server, Web server, messaging server, transaction server, and integration server. It also serves as the foundation for the BEA WebLogic E-Business Platform™, the most trusted e-business infrastructure software in the world.

As the market-leading Java application server, BEA WebLogic Server provides all of the features that are critical to developing and deploying mission-critical e-business applications. These include:

- ***Certified standards leadership*** — BEA is committed to aggressively delivering comprehensive Java 2 Enterprise Edition (J2EE) support, including Enterprise JavaBeans (EJB) 2.0, Servlet, JSP, JMS, JNDI, JMX, JDBC, JCA, and RMI.
- ***Enterprise scalability*** — BEA WebLogic Server's software clustering of dynamic Web pages (Servlets, Java Server Pages) and EJB components, coupled with client connection sharing and database resource pooling, provide the highest levels of performance and scalability.
- ***Rich client options*** — BEA WebLogic Server supports programmatic, browser, and wireless clients.
- ***Robust administration*** — BEA WebLogic Server offers a comprehensive, Web-based administration console for configuring and monitoring all aspects of a BEA WebLogic Server deployment. BEA WebLogic Server is also integrated with the leading management frameworks and products.
- ***E-Commerce-ready security*** — BEA WebLogic Server features Secure Sockets Layer (SSL), Access Control Lists (ACLs), RDBMS, Unix, and Custom Security Realms for integration of encryption and authentication services into e-commerce solutions.
- ***Maximum development and deployment flexibility options*** — BEA WebLogic Server features tight integration with leading databases, operating systems, Web Servers and development tools.

## BEA WebLogic Server Architecture

BEA WebLogic Server is a high-performance application server that listens on the network for client requests. It establishes a connection with a client, including negotiating details such as

protocol, encryption, and authentication. It then processes the client's requests by executing Java classes on behalf of the client.

On the front end, BEA WebLogic Server can act as the primary Web server or can process requests redirected to it by other Web servers such as Netscape Enterprise Server, Microsoft Internet Information Server, or Apache. Other Web servers can handle requests for static HTML Web pages, and pass their servlet and JSP page requests to BEA WebLogic Server.

BEA WebLogic Server clients can be Web browsers, wireless devices, or Java clients. In each case, clients specify a URL (Universal Resource Locator), which provides the network protocol and location of the desired BEA WebLogic Server resource. Secure, encrypted connections can be made using SSL (Secure Sockets Layer).

In the middle tier, BEA WebLogic Server hosts business logic in its robust EJB Container. The EJB Container allows developers to focus on encapsulating business logic and not have to negotiate the complex details of transaction management, security, or naming and directory services.

On the back-end, BEA WebLogic Server can connect to virtually any network-accessible service such as relational databases. Other common back-end services are naming and directory services, messaging systems, and legacy applications that are integrated with BEA WebLogic Server applications.

All of the services are hosted on a high-performance, clusterable foundation to ensure maximum scalability and high availability through sophisticated load balancing and failover mechanisms.

## BEA WebLogic Server as a Web Server

BEA WebLogic Server can be used as the primary Web server for advanced Web-enabled applications. A J2EE Web application is a collection of HTML or Extensible Markup Language (XML) pages, Java Server Pages, servlets, Java classes, applets, images, multimedia files, and other types of files.

A Web application runs in the Web container of a Web server. In a WebLogic Server environment, a Web server is a logical entity, deployed on one or more WebLogic Servers in a cluster. The files in a Web application are stored in a directory structure that, optionally, can be packed into a single “.war” (Web ARchive) file using the Java jar utility. A set of Extensible Markup Language (XML) deployment descriptors defines the components and run-time parameters of an application, such as security settings. Deployment descriptors make it possible to change run-time behaviors without changing the contents of Web application components, and it makes it easy to deploy the same application on multiple Web servers.

Features for using the BEA WebLogic Server as a Web server include:

- Virtual hosting
- Support for proxy server configurations
- Load balancing
- Failover

## Virtual Hosting

BEA WebLogic Server supports virtual hosting, an arrangement that allows a single WebLogic Server or WebLogic Cluster to host multiple Web sites. Each logical Web server has its own host name, but all Web servers are mapped in DNS to the same cluster IP address. When a client sends an HTTP request to the cluster address, a WebLogic Server is selected to serve the request. The Web server name is extracted from the HTTP request headers and is maintained on subsequent exchanges with the client so that the virtual hostname remains constant from the client's perspective. Multiple Web applications can be deployed on a WebLogic Server, and each Web application can be mapped to a virtual host.

## Using Proxy Server Configurations

BEA WebLogic Server can be integrated with existing Web servers. Requests can be proxied from a WebLogic Server to another Web server or, using a native plug-in supplied with WebLogic Server, from another Web server to WebLogic Server. BEA supplies plug-ins for Apache Web Server, Netscape Enterprise Server, and Microsoft Internet Information Server.

The use of proxy Web servers between clients and a set of independent WebLogic Servers or a WebLogic Cluster makes it possible to perform load balancing and failover for Web requests. To the client, there appears to be only one Web server.

## Load Balancing

You can set up multiple BEA WebLogic Servers behind a proxy server to accommodate large volumes of requests. The proxy server performs load-balancing, distributing requests across the multiple servers in the tier behind it. The proxy server can be a WebLogic Server, or it can be an Apache, Netscape, or Microsoft Web server. WebLogic Server includes native code plug-ins for some platforms that allow these third-party Web servers to proxy requests to WebLogic Server.

The proxy server is set up to redirect certain types of requests to the servers behind it. For example, a common arrangement is to configure the proxy server to handle requests for static HTML pages and redirect requests for servlets and Java Server Pages to a WebLogic Cluster behind the proxy.

## Failover

When a Web client starts a servlet session, the proxy server may send subsequent requests that are part of the same session to a different BEA WebLogic Server. WebLogic Server provides session replication to ensure that a client's session state remains available.

There are two types of session replication:

- JDBC session replication can be used with a WebLogic Cluster or with a set of independent WebLogic Servers.
- In-memory session replication.

JDBC session replication writes session data to a database. When the proxy server chooses a different WebLogic Server to handle a request, the chosen server gets the session data from the database.

When a WebLogic Cluster is deployed behind a proxy server, servlet sessions can be replicated over the network to a secondary WebLogic Server selected by the cluster, avoiding the need to access a database. In-memory replication uses fewer resources and is much faster than JDBC session replication, so it is the best way to provide failover for servlets when you have a WebLogic Cluster.

## Security Services

BEA WebLogic Server provides security for applications through a service called a security realm. A security realm provides access to two services:

- An authentication service, which allows WebLogic Server to verify the identity of users
- An authorization service, which controls users' access to applications

### Authentication

A realm has access to a store of users and groups and can authenticate a user by testing a user-supplied credential (usually a password) against the username and credential in the security store. Web browsers support authentication by requesting a username and password when a Web client tries to access a protected WebLogic Server service. Other WebLogic Server clients supply usernames and credentials programmatically when they establish WebLogic Server connections.

### Authorization

BEA WebLogic Server services are protected with Access Control Lists (ACLs). An ACL is a list of users and groups who are authorized to access a service. Once it has authenticated a

user, WebLogic Server checks the ACL for a service before allowing the user to access the service.

## Alternative and Custom Realms

The security realm is pluggable — you can use the default File realm, an alternative realm supplied with BEA WebLogic Server, or you can create your own realm. The default File realm stores users, passwords, groups, and ACLs in an encrypted file. The file is managed through the Administration Console.

WebLogic Server includes alternative realms that access an external store of users and groups and sometimes ACLs. Alternative realms are provided for these external security systems:

- Lightweight Directory Access Protocol (LDAP) service, such as Netscape Directory Server, Microsoft Site Server, and Novell NDS
- Unix login service
- Windows NT domain
- RDBMS, a realm that uses a database system to store users, groups, and ACLs

A custom realm can be developed for use in WebLogic Server by implementing the WebLogic Realm interfaces. WebLogic Server supports realms with a built-in caching system, to minimize calls into the external store. Any features that are not implemented in a custom realm fall back on the default File realm. For example, it is common to develop a custom realm that uses an external store for users and groups, but to maintain ACLs in the File realm.

## Encryption

BEA WebLogic Server supports the Secure Sockets Layer (SSL) protocol, which protects data transferred over a wire. SSL is the standard protocol for secure Web connections. WebLogic Server supports SSL on Web (HTTPS) connections and in standalone Java clients that use RMI-based services.

To provide SSL, you must first purchase a Server ID for your WebLogic Server from a certificate authority, such as VeriSign or GTE CyberTrust. When a client establishes a secure connection, WebLogic Server sends its certificate to the client, allowing the client to verify that the connection is proper. Clients examine the certificate to make sure that it has not expired, matches the server that sent it, and is signed by a recognized certificate authority. The server and client then exchange encryption keys and establish a secure connection.

WebLogic Server can also be configured to require mutual authentication, which requires a client to submit a client certificate that the server must validate before accepting a connection.

# BEA WebLogic Clusters

A WebLogic Cluster is a group of WebLogic Servers that work together to provide an application platform that is more powerful and reliable than a single server. A cluster appears to its clients as a single server but it is, in fact, a group of servers acting as one. It provides two key benefits that are not provided by a single server: scalability and availability.

## Benefits of Using Clusters

WebLogic Clusters are designed to bring scalability and high-availability to J2EE applications in a way that is transparent to application developers. The benefit of scalability is that it expands the capacity of the middle tier beyond that of a single BEA WebLogic Server or a single computer. New WebLogic Servers can be added to a cluster dynamically to increase capacity.

A WebLogic Cluster also guarantees high availability by using the redundancy of multiple servers to insulate clients from failures. The same service can be provided on multiple servers in a cluster. If one server fails, another can take over. The ability to have a functioning server take over from a failed server increases the availability of the application to clients.

## Cluster Architecture

A WebLogic Cluster consists of a number of BEA WebLogic Servers deployed on a network, coordinated with a combination of Domain Name Service (DNS), JNDI naming tree replication, session data replication, and WebLogic RMI.

Web proxy servers between Web clients and the WebLogic Cluster coordinate clustering services for servlets and Java Server Pages. Web proxy servers can be WebLogic Servers, or third-party Web servers from Netscape, Microsoft, or Apache, used with a plug-in supplied with WebLogic Server.

Web clients connect with a WebLogic Cluster by directing requests to a proxy server. Java RMI-based clients connect with a WebLogic Cluster using a cluster address defined on the network.

Server-side code also benefits from the load balancing and failover services provided by a WebLogic Cluster. In J2EE applications, most application code runs in the middle tier and can use services distributed among several WebLogic Servers. For example, a servlet running on WebLogic Server A could use an enterprise bean on WebLogic Server B and read messages from a JMS Queue on WebLogic Server C.

## Defining a WebLogic Cluster in a Network

BEA WebLogic Server services are accessed through DNS, the standard naming service for resources on a network, including the Internet. DNS maps IP addresses, such as 170.0.20.1, to names, such as *mycomputer.mydomain.com* or *www.bea.com*. Each WebLogic Server runs on the network at a unique IP address. A client connects to a WebLogic Server by encoding in a URL, its name and the number of the port where it is listening for connections.

For example, a WebLogic Server running on a computer named *onyx*, configured to listen on port 7701, can be accessed with a Web browser using the following URL: *http://onyx:7701*. For this connection to succeed, the name server on the network must be able to resolve the name *onyx* in the local domain. If the destination server is in another domain on the Internet, the full domain name (e.g., *http://onyx.bea.com:7701*) must be supplied.

An additional DNS entry maps the names of all the WebLogic Servers participating in a cluster to a single cluster name. Clients connect to the cluster using the cluster name or through a Web proxy server that directs requests into the cluster. When DNS performs a lookup on a cluster name, it returns a list of all the servers that belong to the cluster. A client usually selects the first server in the list and, if it gets no response, tries the second server, working its way through the list until it gets a response.

DNS provides the initial load-balancing service that distributes requests across the servers in the cluster. Each DNS responds to a lookup on the cluster name, by rotating the list of servers by one, so that eventually each server gets a turn.

An intelligent router, proxy server, firewall, or other software operating on the network may override DNS and select the initial server based on machine load, network traffic, or other dynamic load balancing criteria.

The initial WebLogic Server connection provides the naming service for the client. It looks up the service requested by the client and chooses a server from the cluster to handle the request, using a load-balancing algorithm configured in WebLogic Server.

## How BEA WebLogic Servers in a Cluster Communicate

BEA WebLogic Servers in a cluster communicate with each other using IP multicast to replicate certain classes of information to all servers in the cluster. A common multicast address is configured for each server in the cluster. When one server sends a message to the cluster's multicast address, all servers receive the message. This process is much more efficient than having servers send point-to-point messages. However, it does require all the servers in a cluster to be on a network with multicast support.

For some services, the cluster selects primary and secondary WebLogic Servers. If the primary WebLogic Server starts processing a request and then becomes unavailable, the secondary server can take over processing of the request without interruption. The primary server replicates state to the secondary server using a server-to-server connection.

Most services can be deployed on any number of WebLogic Servers in a cluster. As each service is deployed, the WebLogic Server uses IP multicast to add the service to a cluster-wide naming tree. Any server in the cluster can find a WebLogic Server to provide a given service by looking up the service in the cluster-wide naming tree. When more than one server can provide a service, the cluster uses a configurable load-balancing algorithm to choose a server.

## Clustered Services

Most BEA WebLogic Server services can be clustered; that is, they can be deployed on an unlimited number of servers in the cluster. The cluster selects the WebLogic Server that will provide a service. Once that server has been selected and stateful objects have been instantiated on the server, the client is pinned to that WebLogic Server until it has finished with the service. If a WebLogic Server hosting a pinned object fails, the client must detect the failure and create another instance on another server in the cluster.

To provide more resilient failover, a WebLogic Cluster avoids pinning an object to a server unless absolutely necessary. In some cases the cluster replicates the stateful object to a backup server to enable failover for the service.

Web applications can be clustered, as described in the previous section “WebLogic Server as a Web Server” (page 6). Servlet sessions are replicated to a secondary server, allowing the cluster to recover from a failure transparently.

All Enterprise JavaBeans can be clustered. They can be deployed on an unlimited number of servers in a WebLogic Cluster. However, not all EJB instances can be clustered. An application can get the home interface for an EJB from any server where the bean has been deployed, and it can use that home interface to create bean instances. If the server that provides the home interface fails, a home interface can be retrieved from another server without interrupting the application.

Some types of EJB instances, including stateless session beans and read-only entity beans, can always be clustered. Stateful session beans can be clustered using in-memory replication to provide failover. Read-write entity beans are always pinned to the server where they are instantiated. If the server hosting a read-write entity bean fails, the application must create a new instance.

A JDBC metapool provides clustering for JDBC connection pools deployed on multiple servers in a WebLogic Cluster. When a client requests a connection from the metapool, the cluster selects the server that will provide the connection, allowing load-balancing and protection against server failure. Once a client has a connection, the state (maintained by the JDBC driver) makes it necessary to pin the client to the host WebLogic Server.

JMS objects can be distributed among the servers in a cluster. Each destination (message Queue or Topic) is managed by a single WebLogic Server in the cluster. But connection factories, which clients use to establish a connection to a destination, can be deployed on

multiple servers in a cluster. By distributing destinations and connection factories throughout a cluster, administrators can manually balance the load for JMS services.

## Server Management and Monitoring

BEA WebLogic Server administration is accomplished by setting attributes for the servers in a domain, using either the Administration Console or the command-line interface. The Administration Console is a Web browser application that allows you to configure WebLogic Server services, manage security, deploy applications, and monitor services dynamically.

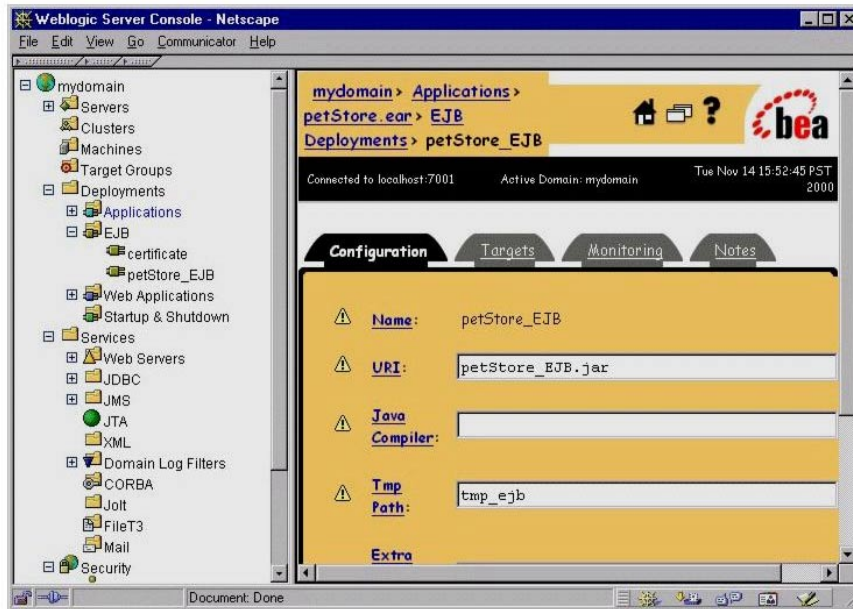
Both the Administration Console and the command-line interface connect to the Administration Server.

The Administration Server is the WebLogic Server used to configure and manage all the WebLogic Servers in its domain. A domain may include multiple WebLogic Clusters and independent WebLogic Servers. If a domain contains only one WebLogic Server, then that server is the Administration Server. In a domain with multiple instances of WebLogic Server, the first instance to start must be the Administration Server.

### Administration Console

The Administration Console runs in a Web browser. It displays the components of the domain it administers, including clusters and independent servers.

The following figure is a sample snapshot from an Administration Console session. WebLogic Servers are in a graphical tree in the left pane of the display. The right pane displays details about the object selected in the left pane.



### BEA WebLogic Server Administration Console session

To use the Administration Console to configure a service, first select an item in the left pane, and then choose the Configuration tab in the right pane. The Administration Console displays the configurable attributes in the right pane. You can use the online help to find detailed information about the displayed attributes.

The usual process for configuring a service in the Administration Console is to configure the service and then select the targets (WebLogic Servers) to which you want to deploy the service. Each deployed service keeps run-time statistics, which you can view in the Administration Console.

## BEA WebLogic jDrivers

BEA offers three WebLogic jDrivers for use with the BEA WebLogic Server software:

- Type-2 native JDBC driver for Oracle that includes distributed transaction capability
- Type 4 JDBC drivers for Informix and Microsoft SQL Server

Type-2 drivers use client libraries supplied by a database vendor, while Type-4 drivers are pure-Java; they connect to the database server at the wire level without vendor-supplied client libraries.

## BEA WebLogic jDriver for Oracle

WebLogic jDriver for Oracle, a Type 2 JDBC driver for the Oracle DBMS, is provided with the BEA WebLogic Server software. To use this driver, you must install a complete Oracle client, including all required libraries, on the machine that will be the client to the Oracle DBMS. This Oracle client installation must contain vendor-supplied client libraries and associated files required by WebLogic Server.

### Oracle Shared Libraries

The WebLogic Server distribution includes a choice of several BEA-supplied native libraries for WebLogic Server. Which library you choose depends on which Oracle client version is installed on your client machine and which version of the Oracle API you will use to access your Oracle server. Before you can install this driver, you must include both the BEA-supplied native library and the Oracle-supplied client libraries in your the client's PATH (Windows NT) or shared library path (Unix).

### Distributed Transactions with BEA WebLogic jDriver for Oracle/XA

WebLogic Server provides a multithreaded JDBC/XA driver for Oracle Corporation's Oracle8i database management system. The WebLogic jDriver for Oracle/XA is the transaction-enabled version of WebLogic jDriver for Oracle. The WebLogic jDriver for Oracle/XA fully supports XA, the bi-directional system-level interface between a transaction manager and a resource manager of the X/Open Distributed Transaction Processing (DTP) model.

## BEA WebLogic jDriver for Microsoft SQL Server

WebLogic jDriver for Microsoft SQL Server is a 100% pure Java implementation of the Java Database Connectivity (JDBC) API, the industry standard for relational database access from Java clients. It provides Java clients with direct access to Microsoft SQL Server. The driver is available in two versions: one for SQL Server versions 6.5 and 7.0, and another for SQL Server 7.0 only.

Like all Type 4 JDBC drivers, WebLogic jDriver for Microsoft SQL Server is pure Java; it requires no vendor-supported client libraries. WebLogic jDriver for Microsoft SQL Server communicates directly with SQL Server through a TCP/IP network, using the SQL Server Tabular Data Stream protocol, so DB-Library does not have to be installed on a client computer.

## BEA WebLogic jDriver for Informix

WebLogic jDriver for Informix is a 100% pure Java implementation of the Java Database Connectivity (JDBC) API, the industry standard for relational database access from Java clients. It provides Java clients with direct access to the Informix database management system (DBMS).

# Integrated Development Tools

Beyond sharing open standards, BEA WebLogic Server is tightly integrated with many of today's leading development tools, including WebGain VisualCafé, IBM VisualAge, Sybase PowerJ, WebGain TopLink, KL Group Jprobe, Versant Enterprise Container, eXcelon Javlin, Object Design ObjectStore, and others.

## New Features in Version 6.0

The market-leading and most innovative Java application server is now even more powerful, easier to use, and provides broader support for e-business with greater reliability and manageability.

BEA WebLogic Server 6.0 delivers simplified manageability and ease-of-use across the complete application lifecycle — from development to deployment to system management. It delivers unprecedented power and flexibility for driving a wider range of e-business applications by integrating an enterprise messaging server into the industry's leading e-business application platform.

## Enterprise Messaging

BEA WebLogic Server 6.0 includes high-volume messaging based on the Java Messaging Service (JMS).

The use of asynchronous messaging allows for the development of loosely connected applications. These systems are typically more resilient in the event of failures, and more easily extensible as new applications are developed. Additionally, messaging provides an effective means of transmitting events between applications. Java Messaging Service (JMS) brings asynchronous messaging services to Java applications developers.

BEA WebLogic Server delivers a powerful, flexible, and tightly integrated messaging foundation — written to the JMS standard — which serves as the backbone for enterprise messaging systems. The power of BEA WebLogic JMS comes from its integration with BEA WebLogic Server clustering and its ability to do database, file, or in-memory persistence. This flexibility gives developers the option to do Publish/Subscribe or Point-to-Point messaging. WebLogic JMS also supports either Multicast or TCP/IP transport protocols. Since it is integrated with the core WebLogic services, WebLogic JMS can directly access all of the Enterprise JavaBean (EJB) and transaction services needed to fully leverage a message-based architecture from a single, manageable messaging foundation.

Features include:

- Clustering of multiple servers
- Multicasting, allowing the delivery of messages to a select group of hosts using IP multicast
- Messages containing Extensible Markup Language (XML)
- Graphical configuration using the WebLogic Administration Console and/or programmatic configuration using the JMS API

## Enhanced XML Support

Extensible Markup Language (XML) comes to the forefront in BEA WebLogic Server 6.0. By leveraging the power and flexibility of XML from the server to the client, developers can seamlessly share and process content across the enterprise or over the Internet.

Features include:

- Single XML and XSL processors with XSL processing tags for JSP
- XML schema and DTD repository
- JAXP implementation for easier access to XML parsers
- SOAP implementation
- Data binding reference implementation
- DOClet/DOMlet examples

## Enterprise JavaBeans 2.0

EJB 2.0 makes major improvements in the areas of messaging and database integration. The integration of JMS and EJB allows enterprise beans to participate fully in loosely connected systems. Through this integration, message-driven EJBs can receive and act upon JMS messages without the involvement (or existence) of an application client user interface. This new capability makes EJB an excellent choice for the development of shared business services. Additionally, any EJB can send asynchronous messages via the JMS API. For the first time, asynchronous data processing applications can be built on the J2EE platform — a realm previously dominated by mainframes.

The EJB architecture greatly simplifies the connection between the application and the database tiers. The EJB 2.0 specification takes this benefit to the next level by allowing developers to develop portable applications that are database-independent and free of database access code using the advanced Container Managed Persistence features in BEA WebLogic Server 6.0.

Features include:

- Support for message-driven beans
- Transactional integration with WebLogic JMS and other messaging systems
- EJB 2.0 container-managed persistence services, including support for EJB-QL
- EJB 2.0 XML deployment properties

## Distributed Transaction Management

The new transaction manager, leveraging BEA's expertise with BEA Tuxedo, gives developers the ability to do complex, distributed transactions from Java-based applications.

The Internet is about transactions. Sometimes transactions involve many systems that need to be coordinated to either perform a transaction, or not. The system may have to check inventory, validate a credit card, and see if delivery is available — all part of one long transaction.

BEA WebLogic Server provides a distributed transaction manager to oversee such complex transactions. If any portion of the transaction fails, the entire transaction fails. In doing so, BEA WebLogic Server centrally coordinates these various systems, including databases, mainframes, messaging systems and any other application that is exposed as an XA resource.

Features include:

- Java transaction manager implemented by core BEA Tuxedo transaction processing team
- Two-phase commit transactions
- Ability to work with any XA-compliant resource
- Support for databases via JDBC
- Support for message queuing systems (MQSeries, WebLogic JMS)
- Support for distributed transactions across multiple resource managers

## Web-based System Management

New management and usability features greatly enhance administrator and developer productivity. These features dramatically improve every aspect of the application lifecycle — from installation to configuration, development, deployment, and management.

Now more than ever, companies are deploying e-business systems that should never go down. When a system is down, opportunities are lost. To maintain these systems, administrators need the proper tools to oversee all aspects of their applications.

BEA WebLogic Server comes with a powerful, Web-based management console to provide administrators all of the tools they need to deploy, configure, and monitor their applications.

The Administration Console is the window into the WebLogic Administration Service. The Administration Service, an implementation of Sun's Java Management Extension (JMX) standard, provides the facilities for managing all BEA WebLogic resources. Through the Administration Console, you can configure attributes of resources, deploy applications or components, monitor resource usage (such as server load or Java Virtual Machine memory usage or database connection pool load), view log messages, shutdown servers, or perform other management actions.

Features include:

- A browser-based console
- JMX architecture
- The ability to manage multiple clusters from single console
- The capability to manage all areas of configuration
- SNMP enhancements

## J2EE Certification

BEA WebLogic Server 6.0 is certified J2EE-compliant. Certification assures enterprises and developers alike that APIs and development features will work in a uniform way. BEA WebLogic Server 6.0 has been updated to support the latest J2EE specification (J2EE Version 1.2).

BEA WebLogic Server also supports significant amounts of the proposed J2EE Version 1.3 enhancements that will be certified when tests become available.

## Web and Object Clustering Enhancements

BEA WebLogic Server 6.0 extends the clustering architecture to improve scalability and high availability.

JDBC MultiPools create a list of connection pools to be used by a single WebLogic Server. A configurable algorithm determines which connection is returned. MultiPools provide support for load balancing and high availability, and make it easier for an application to switch to another RDBMS for distributed processing or during a failover situation. BEA WebLogic Server 6.0 extends the existing servlet replication model to session EJBs. It also has the ability to do in-memory replication of stateful session EJBs to provide performance failover of business logic.

## Web Server Enhancements

BEA WebLogic Server 6.0 eliminates the requirement for a separate Web server with a powerful and tightly integrated Web/application server architecture.

The BEA WebLogic Web server can handle high volume Web sites, serving static HTML, as well as servlets and Java Server Pages (JSP).

Features include:

- Support for the HTTP 1.1 standard
- Support for Virtual Hosting
- Integration with WebLogic Administration Console
- Integration with hardware-based and software-based Web load balancing solutions

## Security Enhancements

The security functionality of BEA WebLogic Server has been improved.

Features include:

- JAAS Login, replacing realm authentication
- Enhanced support for generating audit trails
- ACL improvements
- Password guessing protection
- Denial-of-service protection
- Improved administration of the RDBMS realm

## Internationalization

BEA WebLogic Server 6.0 is able to support virtually any language, including those that require double-byte character sets.

In addition, a Kanji (Japanese) version of BEA WebLogic Server is available.

# BEA WebLogic E-Business Platform™

The key to becoming an agile, adaptable, *future-proof* e-business is the technology infrastructure. The BEA WebLogic E-Business Platform provides the essential infrastructure for building an integrated e-business. It includes:

- Market-leading BEA WebLogic® application servers, including BEA WebLogic Server, for rapidly building, deploying, and managing e-business applications
- BEA WebLogic Commerce Server™, for rapidly developing and adapting e-commerce applications
- BEA WebLogic Personalization Server™, for building applications that win and retain customers
- BEA WebLogic Collaborate™ to support powerful new ways of linking and interacting with suppliers and partners
- BEA WebLogic Process Integrator™, to automate business process workflow
- BEA WebLogic M-Commerce Solution™, to support mobile commerce and anytime, anywhere e-business
- A wide range of education, customer support, and professional services to enable businesses to get in the game fast and integrate their e-business from end to end

The BEA WebLogic E-Business Platform enables e-businesses to reliably service customers, scale to handle unpredictable levels of growth across the entire chain of commerce, rapidly bring new products and services to market, personalize services for customers to capture their loyalty, collaborate flexibly with partners, suppliers and customers, and adapt nimbly to an increasing rate of change. In short, it *future-proofs* e-business.

# About BEA

BEA Systems, Inc. (Nasdaq: BEAS) is one of the world's leading e-business infrastructure software companies, with over 8,000 customers around the world, including the majority of the *Fortune* Global 500. BEA and its WebLogic® brand are among the most trusted names in e-business. Businesses built on the award-winning BEA WebLogic E-Business Platform™ are reliable, highly scalable, and poised to bring new services to market quickly. BEA's e-business platform is the de facto standard for over 1,200 systems integrators, independent software vendors (ISVs) and application service providers (ASPs) to provide complete solutions that fast-track and future-proof e-businesses for high growth and profitability. Headquartered in San Jose, Calif., BEA has 90 offices in 30 countries and is on the Web at [www.bea.com](http://www.bea.com).





**BEA Systems, Inc.**  
2315 North First Street  
San Jose, CA 95131 U.S.A.  
Telephone: +1.408.570.8000  
Facsimile: +1.408.570.8901  
[www.bea.com](http://www.bea.com)  
CRG0206E0500-1A