

---

# Glossary

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

## A

### access control list (ACL)

Used to authenticate users and manage access to network services. The WebLogic implementation of ACLs is based on the `java.security.acl` package. Each entry in an ACL contains a set of permission associated with a particular principal.

*See also* **authentication, realm, permission, principal** and Using WebLogic ACLs (Access Control Lists).

### ACID properties

The essential characteristics of transaction processing systems:

- ◆ **Atomicity:** All changes that a transaction makes to a database are made permanent; otherwise, all changes are rolled back.
- ◆ **Consistency:** A successful transaction transforms a database from a previous valid state to a new valid state.
- ◆ **Isolation:** Changes that a transaction makes to a database are not visible to other operations until the transaction completes its work.
- ◆ **Durability:** Changes that a transaction makes to a database survive future system or media failures.

*See also* **transaction**.

### ACL

*See* **access control list (ACL)**.

### API

*See* **application programming interface (API)**.

---

### **applet**

Client-side Java program, usually embedded in a HTML page and viewed with a web browser.

*See also* Using WebLogic for applet programming and Troubleshooting applet security problems

### **application programming interface (API)**

A specification for a programmer about how a given application accesses the behavior and state of classes and objects.

### **authentication**

(one-way authorization and two-way client/server authorization) Used to verify the identity of a user or computer. Also the technology that guarantees the source of information to a recipient.

*See also* **access control list (ACL)** and **secure sockets layer (SSL)**.

### **▲ Back To Top**

## **B**

### **bean-managed transaction**

A transaction where an Enterprise JavaBean demarcates the transaction boundaries. In a bean-managed transaction, controls can be specified using JTA. The EJB code manages the transaction, which can begin in one method and end in another.

*See also* **container-managed transaction** and **transaction**.

### **byte code**

A compiled Java program which can be run (interpreted) on any computer with a Java virtual machine (JVM).

*See also* **Java virtual machine (JVM)**.

### **▲ Back To Top**

---

## C

### **cab**

A compressed file used to load classes and other files required to run an applet in a single hypertext transfer protocol (HTTP) request. Cab files can only be used with applets running under Microsoft Internet Explorer, version 4.0 or later.

*See also* **jar**, **applet**, **hypertext transfer protocol (HTTP)** and Using the Apple-  
tArchiver.

### **cipher suite**

A set of algorithms used in cryptography.

*See also* **secure sockets layer (SSL)**.

### **CLASSPATH**

The full path to the directory where the classes necessary for a Java program to run are located.

### **client/network**

A network where requests -- from heterogeneous clients, probably making requests over various protocols -- for many different services on the network are fulfilled transparently by a high-performance, intelligent intermediate server or cluster of servers, like the pure-Java WebLogic Server.

*See also* **client/server** and **WebLogic Server**.

### **client/server**

A network architecture where computer processing is distributed among clients (desktop PCs) and a server or servers (central computer).

*See also* **client/network**.

### **clusterable stub**

A clusterable stub is a wrapper that delegates its calls to a service instance. It has intelligence about how to deal with load balancing and failover situations, in which a service might be handled transparently by another server. You generate a clusterable stubs by setting the “-clusterable” flags for RMIC or similar EJB deployment descriptor properties.

*See also* Using WebLogic Clusters, **failover**, **load balancing** and **stub**.

---

### **CODEBASE tag**

The base URL for locating an applet's classes on the server host. Combining the CODEBASE with an applet's CODE name will provide the complete URL for the applet.

*See also* **applet**, **CODE tag**, **URL** and Troubleshooting CODEBASE.

### **CODE tag**

The full package name of the classes for an applet. One of the attributes of the HTML applet tag.

*See also* **applet**, **package name** and Troubleshooting CODEBASE.

### **commit**

- 1) Complete a transaction so that changes are recorded and stable. Protected resources are released.
- 2) The declaration or process of making a transaction's updates and messages visible to other transactions. When a transaction commits, all its effects become public and durable. After commitment, the effects of a transaction cannot be reversed automatically.

*See also* **transaction**.

### **common object request broker architecture (CORBA)**

A standard from the Object Management Group (OMG) for communicating between distributed objects. CORBA uses IDLs (interface definition language) and ORBs (object request brokers), and communicates with IIOP (Internet Interoperability Protocol).

*See also* **IDL**, **IIOP**, **ORB** and [Object Management Group](#).

### **container-managed transaction**

A transaction where an Enterprise JavaBean container demarcates the transaction boundaries. In a container-managed transaction, controls are specified in the deployment descriptor. When a bean method is invoked, the container manages the transaction, which begins and ends in the same method. An entity bean must use container-managed transactions.

*See also* **bean-managed transaction** and **transaction**.

### **CORBA**

*See* **common object request broker architecture (CORBA)**.

---

## **credential**

Validation of a user's identity. A credential may be a password, an X509 certificate or certificates chain, or any other token that validates a user.

*See also* **realm** and **user**.

## **Back To Top**

## D

### **daemon**

A program that runs automatically and continuously in the background.

### **dbKona**

WebLogic's set of high-level Java objects, usable with any JDBC driver, for database access.

*See also* **JDBC**.

### **database management system (DBMS)**

#### **DBMS**

*See* **database management system (DBMS)**.

### **direct programming model**

A lookup using this model produces a replica-aware stub that may be handled by different service providers each time it is executed upon. The direct model is appropriate for use with stateless objects.

*See also* Programming models in Using WebLogic Clusters, **indirect programming model**, **stateless object** and **targeted programming model**.

### **distributed object model (DOM)**

An abstraction that describes the way applications are partitioned, that is, how the logic is divided between the tiers (multitier or two-tier) that support it.

*See also* **multitier** and **two-tier**.

### **distributed transaction**

A transaction involving multiple transaction managers. In a distributed transaction environment, a client application may send requests to several servers resulting in resource updates at multiple resource managers. To complete the transaction, the

---

transaction manager for each participant (client, servers, and resource managers) must be polled to coordinate the commit process for each participant within its domain.

*See also* **transaction**.

## **DLL**

*See* **dynamic link library (DLL)**.

## **domain**

An interrelated ensemble of WebLogic Server resources defined in a single configuration file.

## **DOM**

*See* **distributed object model (DOM)**.

## **DTD**

**Document Type Definition**

## **dynamic link library (DLL)**

(Microsoft Windows only). WebLogic uses DLLs published by database vendors for the two-tier native drivers. Using a DLL means that a Java program has native code in it.

## **Back To Top**

## **E**

### **encryption (MD5, RC4, DES, RSA)**

Algorithms for protecting communications from the casual viewer, providing access only to specific parties.

*See also* **secure sockets layer (SSL)**.

### **exception**

Java's method for handling runtime errors.

*See also* [What's an Exception and Why Do I Care?](#), on the Sun website.

---

### **extensible markup language (XML)**

A specification submitted to the W3C designed to add functionality to web documents, primarily through the use of customized tags. XML is a subset of standardized general markup language (SGML).

#### **Back To Top**

## **F**

### **failover**

Failover is the ability to transparently handle failure of a service invocation by handing the request off to another service provider.

This includes the ability to handle failure of the service that performs lookups of service providers, the ability to recover in the event of a server failure and being able to find another instance of a service on a reachable server.

*See also* Using WebLogic Clusters and **clusterable stub**.

### **firewall**

A firewall monitors traffic between an internal network and the Internet and regulates the type of network traffic that can pass through it.

#### **Back To Top**

## **G**

### **garbage collection (GC)**

GC reclaims memory from unused objects and variables while a Java program is executing. Usually executes in the background automatically, but can also be specified by the programmer.

### **GC**

*See* **garbage collection (GC)**.

### **general protection fault (GPF)**

An error condition in Windows programs that results from a program attempting to access an area of memory in use by another program.

### **global transaction**

*See* **distributed transaction**.

---

## **GPF**

*See* **general protection fault (GPF)**.

## **group**

A set of users that share some characteristics. An ACL can assign permissions to a group. Since all permissions are positive in the WebLogic realm, giving a permission to a group is the same as giving the permission to each user who is a member of the group. In realms that support negative permissions, individual permissions override group permissions.

*See also* **ACL**, **realm** and **permission**.

## **Back To Top**

## **H**

### **heuristic commit**

An heuristic completion where all updates related to the transaction were committed.

*See also* **heuristic completion**.

### **heuristic completion**

An heuristic completion occurs when a resource makes a unilateral decision during the completion stage of a distributed transaction to commit or rollback updates. This can leave distributed data in an indeterminate state. Network failures or transaction time-outs are possible causes for heuristic completion.

*See also* **commit**, **distributed transaction**, and **rollback**.

### **heuristic decision**

*See* **heuristic completion**

### **heuristic hazard**

An heuristic completion where the resource manager does not know whether at least one of the updates related to the transaction was rolled back or committed.

*See also* **heuristic completion**.

### **heuristic mixed**

An heuristic completion where some updates related to the transaction were rolled back and other updates were committed.

---

*See also* **heuristic completion**.

**heuristic rollback**

An heuristic completion where all updates related to the transaction were rolled back.

*See also* **heuristic completion**.

**htmlKona**

WebLogic's Java objects for dynamically generating HTML pages or servlets.

**HTTP**

*See* **hypertext transfer protocol (HTTP)**.

**HTTP server (HTTPD)**

A server that transfers data using the hypertext transfer protocol.

**HTTPD**

*See* **HTTP server (HTTPD)**.

**hypertext transfer protocol (HTTP)**

Part of the TCP protocol family, based on TCP/IP, used to fetch hypertext files from remote hosts.

A HTTP server (HTTPD) is a server that employs HTTP to transfer data.

Hypertext transport protocol secure (HTTPS) is a protocol for accessing a secure web server.

*See also* **TCP**.

 **Back To Top**

|

**I18N**

*See* **Internationalization**

**IDE**

*See* **integrated development environment (IDE)**.

---

## **idempotent**

A function  $f$  is considered idempotent if  $f(f(x)) = x$ . Computer science defines a procedure idempotent if its result is the same when called repeatedly with the same arguments.

This is an especially useful property when dealing with retry of a replicated object. When a communication failure occurs while invoking a remote method, it's not always possible to know if the failure occurred before or after the method invoked was completed on the server. If the failure occurred after the method completed on the server, a retry will result in calling the same method twice (albeit on another server). Such behavior can lead to incorrect results unless the method is idempotent. In that case, invoking the same method with the same arguments multiple times is no different than invoking it once.

*See also* Using WebLogic Clusters.

## **IDL**

*See* **interface definition language (IDL)**.

## **I1OP**

*See* **Internationalization**.

## **implementation**

A named object that implements a named interface; for example, *foo\_object* implements *foo\_impl*. Used with RMI to invoke a remote interface.

## **impl**

*See* **implementation**.

## **indirect programming model**

A lookup using this model returns a replica-aware stub that is guaranteed to be handled by the same service provider each time it is executed upon. You should use this programming model for stateful objects.

*See also* Programming models in Using WebLogic Clusters, **direct programming model**, **stateful object** and **targeted programming model**.

## **integrated development environment (IDE)**

Examples include Borland's JBuilder, Sybase's PowerJ, and Symantec's Visual Cafe.

*See also* Using Borland's JBuilder with WebLogic.

---

**interface definition language (IDL)**

Objects in the CORBA distributed object environment are defined by an IDL, which describes the services performed by the object and how the data is to be passed to it.

*See also* **CORBA**.

**Internationalization**

The preparation of software for proper behavior in multiple locales.

**Internet interoperability protocol (IIOP)**

The wire format used by CORBA clients to communicate with ORBs, etc.

*See also* **CORBA** and **ORB**.

**Internet**

A network of networks, consisting of literally millions of hosts around the world, that uses the TCP family of protocols to communicate. It is constructed from numerous smaller networks, using common set of protocols.

*See also* **intranet** and **TCP**.

**Internet protocol (IP)**

One of the protocols in the TCP family.

*See also* **TCP**.

**intranet**

A private network that may be a LAN, WAN, or linked over the Internet.

*See also* **L10N**, **WAN** and **Internet**.

**IP**

*See* **Internet protocol (IP)**.

**IP multicast**

An unreliable packet delivery service supported by all operating systems and most routers, used by WebLogic Clusters, specifically by WSAP.

*See also* **IP**.

▲ **Back To Top**

---

## J

### **jar**

Java archive. A compressed (zip) file used to load classes and other files required to run an applet in a single HTTP request. Jar files can only be used with applets running under Netscape 3.0 or later.

*See also* Using the AppletArchiver, **applet**, **HTTP** and **cab**.

### **JavaBeans**

JavaBeans is a portable, platform-independent component model written in Java, developed by [JavaSoft](#).

*See also* **Java Enterprise API**

### **Java heap memory**

An area of memory designated for runtime use by a Java program.

### **Java system property**

Provides a way for a Java program to obtain information about its environment; similar to the way native programs use environment variables. WebLogic's use of properties conforms to the class `java.util.Properties`, which defines the use of Java system properties.

### **Java database connectivity (JDBC)**

A [JavaSoft](#) specification for Java access to relational databases. Current release is 2.0.

*See also* **Java Enterprise API**

### **Java Development Kit (JDK)**

From [JavaSoft](#). Current release is 1.2.2.

*See also* **JVM**.

### **Java Enterprise API**

The [Java Enterprise APIs from JavaSoft](#) are a collection of APIs designed to facilitate the development of cross-platform Java Enterprise software. Links to information on WebLogic's implementations of the Java Enterprise APIs are listed below.

*See also* WebLogic Enterprise JavaBeans, WebLogic HTTP servlets, WebLogic JDBC Options, WebLogic JNDI, WebLogic RMI, and WebLogic ACL

---

Additional implementations of the Java Enterprise APIs are currently under development.

### **Java naming and directory interface (JNDI)**

A specification describing how application servers should make named resources and file systems known and accessible to all kinds of users. WebLogic JNDI is an implementation of the [JavaSoft JNDI specification](#).

*See also* **Java Enterprise API** and **WebLogic JNDI**.

### **Java Runtime Environment (JRE)**

A self-contained Java environment that includes just the JVM. With a JRE installed, your users can run a Java program. You can publish a JRE (they come in several versions from various vendors) with an application with ZAC to insure that your users have the Java environment your application expects.

*See also* **JDK** and **JVM**.

### **Java Transaction API (JTA)**

A high-level application transaction interface and a Java mapping to XA. Allows an application to control user transaction boundaries. The XA interface allows an external transaction manager to control transaction boundaries for operations performed by multiple resource managers using the two-phase commit X/Open XA protocol. The API is defined in the `javax.transaction` and `javax.transaction.XA` packages.

*See also* **transaction**, **transaction manager**, and **resource manager**.

### **Java transaction services (JTS)**

A way to maintain control of distributed transactions in order to protect them from sharing violations or other failures. WebLogic uses JTS internally for its transaction processing functionality, but does not export any of its JTS implementation as a public API.

*See also* **Java Enterprise API**

### **Java virtual machine (JVM)**

The Java virtual machine consists of a bytecode instruction set, a set of registers, a stack, a garbage-collected heap, and an area for storing methods. Java, an interpreted language, runs (or is interpreted) in the JVM on a particular platform (UNIX, Windows 95/NT).

*See also* **JDK**.

---

## **JDBC**

*See* **Java database connectivity (JDBC)**.

## **jdbcKona**

Formerly known as jdbcKona, now known as WebLogic jDriver.

*See also* **JDBC** and **WebLogic jDriver**.

## **JDBC-ODBC bridge**

General database connectivity through the ODBC client library, created as a joint project between [Intersolv](#) and [JavaSoft](#). Allows Java connectivity to any relational database.

*See also* **ODBC**.

## **JDK**

*See* **Java Development Kit (JDK)**.

## **JIT**

*See* **just-in-time compiler (JIT)**.

## **JNDI**

*See* **Java naming and directory interface (JNDI)**.

## **JRE**

*See* **Java Runtime Environment (JRE)**.

## **JTA**

*See* **Java Transaction API (JTA)**.

## **JTS**

*See* **Java transaction services (JTS)**.

## **just-in-time compiler (JIT)**

Stores generated machine code in memory and re-uses it when possible. Can result in improved performance for some Java applications. If you're debugging, turn off the JIT so that the stack traces show line numbers in the code.

*See also* **WebLogic FAQ**

## **JVM**

*See* **Java virtual machine (JVM)**.

---

▲ **Back To Top**

K

▲ **Back To Top**

L

**L10N**

*See* **Localization**

**LAN**

Local Area Network.

*See also* **WAN**.

**load balancing**

Load balancing means moving around the work of answering client requests in response to individual server load. Load balancing requires the ability to move a provider, both its identity and state, from one host to another.

*See also* Using WebLogic Clusters (Developers Guide), Setting up a WebLogic Cluster (Admin Guide), and **clusterable stub**.

**local transaction**

A transaction that includes a single resource manager as a transaction participant. Contrast this with **distributed transaction**.

**Localization**

The use of locale-specific language and constructs (for example, date formats) at runtime.

▲ **Back To Top**

M

**Message Catalog**

A description of a collection of text messages indexed by a unique identifier.

---

**multithreading**

Allows computer applications to perform multiple tasks simultaneously.

**multitier**

A networked environment or application that includes clients (one tier), all the network services to which they need access (one or more tiers), and brokers, application servers, and other kinds of resources suppliers (one or more tiers). Two-tier is commonly used to refer to a client-server environment or application.

*See also* **client/network** and **two-tier**.

**▲ Back To Top****N****native method**

A Java mechanism for calling methods or functions in a language other than Java.

*See also* **WebLogic jDriver**.

**▲ Back To Top****O****object request broker (ORB)**

In CORBA, the common interface used by the client. The ORB directs the request to the appropriate server that contains the object and redirects the results back to the client.

*See also* **CORBA**.

**ODBC**

*See* **open database connectivity (ODBC)**.

**OLTP**

*See* **online transaction processing (OLTP)**.

**online transaction processing (OLTP)**

A method of assuring the integrity of each database transaction, usually through a process called two-phase commit.

---

### **open database connectivity (ODBC)**

The Microsoft standard used to access relational databases.

*See also* **JDBC** and **JDBC-ODBC bridge**.

### **open software distribution (OSD)**

A file format specification submitted to the [W3C](#) used to facilitate automatic distribution of software over a network. OSD specifies the use of an XML-encoded manifest for a software distribution that details the component files and their dependencies.

*See also* **XML**.

### **ORB**

*See* **object request broker (ORB)**.

### **OSD**

*See* **open software distribution (OSD)**.

### **Back To Top**

## **P**

### **package name**

A logical grouping of classes or interfaces in Java, declared by the package keyword on the first line of a `.java` file. A fully qualified package name is a unique naming scheme (for example, `weblogic.jdbc.t3.Driver`) reflecting the hierarchy and location of the compiled class file. Compiled Java class files that are part of a named package, must be placed in the **CLASSPATH** of the host VM in order to be accessed by the Java interpreter or other utilities. Package name forms the **CODE** value in an applet tag.

*See also* **CLASSPATH** and **CODE tag**.

### **permission**

Ability to carry out certain operations on certain objects, as defined in an access control list. Permissions may be positive (explicitly allowing certain operations on certain objects), or negative (explicitly disallowing certain operations on certain objects). In the default WebLogic realm, all permissions are positive; to disallow an operation, you exclude a principal from the **ACL** that allows the operation.

*See also* **realm**, **access control list (ACL)** and **principal**.

---

**persistence**

A process that saves information or state that would otherwise be transient.  
*See also* Using WebLogic Enterprise JavaBeans.

**PKCSI**

*See* **public-key cryptography standards (PKCS1)**.

**principal**

An entity that has a set of permissions within a realm, as defined in an access control list. Both users and groups are of type Principal. Except in certain special cases, an ACL must always have at least one Principal owner, who holds authority for managing the ACL.

*See also* **realm, user, permission, access control list (ACL) and group**.

**proxy**

To transfer data processing tasks to another program or device.

*See also* **proxy server**, Installing the WebLogic-to-Microsoft-IIS Bridge (ISAPI), and Installing the WebLogic-to-Netscape-Enterprise-Server Bridge (NSAPI).

**proxy server**

A proxy server sends requests to another server for processing. WebLogic supports proxying of HTTP requests with its HTTPProxyServlet. You can proxy to WebLogic from Netscape and IIS via WebLogic's NSAPI & ISAPI plug-ins. The use of a proxy server is invisible to the end user.

*See also* **HTTP, proxy**, Installing the WebLogic-to-Netscape-Enterprise-Server Bridge (NSAPI), and Installing the WebLogic-to-Microsoft-IIS Bridge (ISAPI).

**public-key cryptography standards (PKCS1)**

A standard administered by [RSA Data Security](#) for the implementation of encrypted data transmission.

*See also* **SSL** and **RSA**.

▲ **Back To Top**

Q

▲ **Back To Top**

---

## R

### realm

Domain for a set of security features. The realm organizes security information and defines its range of operations. A realm has its own idea of principals, permissions, and ACLs. Particular security domains are reflected in Java as realm instances. A realm determines how a user is authenticated and retrieves access control lists for given names.

*See also* **credential, group, user, permission, principal, ACL** and Using WebLogic ACLs.

### remote method invocation (RMI)

With RMI, an application can invoke methods on objects stored in any remote JVM as if those objects existed locally. WebLogic RMI is an implementation of the [JavaSoft specification](#).

*See also* **Java Enterprise API** and **JVM**.

### remote procedure call (RPC)

A database procedure that is stored on the database and can be executed by name. A client with the proper permissions can request that the procedure be executed and the result returned to the client.

### replica handler

A replica handler is responsible for load balancing and failover for a replica-aware stub. The replica handler maintains a list of available replicas and chooses a replica for execution when a remote invocation is made on the replica-aware stub. You choose one of WebLogic's built-in replica handlers by setting various flags for RMI and EJB compilation; these flags affect how the replica handler treats such service characteristics as failures and retries, and server affinity.

*See also* **RMI, replica-aware stub, stub** and Using WebLogic Clusters.

### replica-aware stub

A replica-aware stub is a wrapper that delegates its calls to a service instance. It has intelligence about how to deal with load-balancing and failover situations, in which a service might be handled transparently by another server. A replica-aware stub uses a component called the replica handler to encapsulate such behavior. You generate a replica-aware stub by setting the “-replicaAware” flags for RMIC or a similar EJB deployment descriptor property.

*See also* **replica handler, stub** and Using WebLogic Clusters.

---

**resource manager**

An interface and associated software that provides access to a collection of information and processes; for example, a database management system. Resource managers provide transaction capabilities and permanence of actions; they are the entities accessed and controlled within a global transaction.

*See also* **transaction**.

**RMI**

*See* **remote method invocation (RMI)**.

**roll back**

To terminate a transaction in such a way that all resources updated within a transaction revert to their original state.

**rollback**

The event that ends a transaction and nullifies or undoes all changes to resources that were specified during that transaction.

**RPC**

*See* **remote procedure call (RPC)**.

**RSA**

A public-key encryption algorithm for encrypting data transmissions. RSA is a product of the [RSA Data Security](#) company and is one of the algorithms used by SSL to encrypt transmissions between web servers and browsers.

*See also* **SSL** and **encryption (MD5, RC4, DES, RSA)**.

**▲ Back To Top****S****serialize**

A way to pass objects from one JVM to another. Objects must be serialized before they are sent across the wire.

*See also* **JVM**.

---

**server**

An application that answers and fulfills client request. Examples of servers include a DBMS server that provides connectivity to a relational database and an HTTP server that provides HTML pages and services HTTP requests.

*See also* **HTTPD**, **HTTP** and **WebLogic Server**.

**server-side**

Java code that runs in a server's JVM instead of on the client's JVM. Examples of server-side code include a Java-Servlet-API (HTTP) servlet, a WebLogic startup class, an EJBBean, or a WebLogic Remote class.

*See also* **HTTP**, **JVM**, **WebLogic Remote**, **servlet**, Using WebLogic HTTP Servlets, and Using WebLogic Enterprise Java Beans.

**servlet**

A server-side Java program that is usually executed in response to an HTTP request and produces its output in a browser.

*See also* **server-side** and Using WebLogic HTTP Servlets.

**skeleton**

A server-side representation of a remote object that takes serialized requests from a stub, deserializes and unpacks it, and submits it as a method call to be invoked on the object's implementation. The server-side skeleton is responsible for deserializing and unpacking the request from its companion client-side stub.

*See also* **server-side**, **implementation** and Using WebLogic RMI.

**secure sockets layer (SSL)**

A standard for secure Internet communications, proposed by [Netscape](#). WebLogic security services support X.509 certificates and access control lists (ACLs) to authenticate participants and manage access to network services.

*See also* **ACL**, **PKCSI**, **encryption (MD5, RC4, DES, RSA)**, **RSA**, **authentication**, **cipher suite** and Using WebLogic SSL.

**SSL**

*See* **secure sockets layer (SSL)**.

**stack trace**

A Java Exception can be examined for its stack trace, which is useful in debugging.

---

### **stateful object**

A stateful object stores application data; its state may change from one method execution to another. In a clustered environment, both load balancing and failover have certain requirements for stateful objects, since if a server hosting a stateful service fails or balances, the state of the service must accompany the object as it moves to another host.

### **stateless object**

A stateless object saves no application data; it is idempotent. Each operation on the object is independent of any other operations. (Don't confuse this general concept of "stateless" with the very specific definition when applied to a session EJB.)

### **stub**

A client-side representation of a remote object that is used to invoke methods on the implementation of the remote object. Defines the interface to the remote object implementation of an object. The stub is responsible for packaging up the client request, serializing it, and shipping it to the companion skeleton on the server side.

*See also* **implementation**, **skeleton** and Using WebLogic RMI.

### **▲ Back To Top**

## **T**

### **targeted programming model**

A lookup using this model returns a stub for a singular, stateful object, like a file system -- that is represented in the cluster-wide JNDI tree. Such a stub can't do failover or load balancing, since the service itself is unique.

*See also* Programming models in Using WebLogic Clusters, **direct programming model**, and **indirect programming model**.

## **TCP**

*See* **transmission control protocol (TCP)**.

### **transmission control protocol (TCP)**

The family of protocols that specify internet communications.

*See also* **Internet**.

---

## **Tengah**

*Tengah* is the former name of various products and APIs which are now referred to by the name *WebLogic*.

Generally, all entities whose name formerly included *Tengah* -- such as the Tengah Server -- are now called *WebLogic* (or WebLogic Server). Where the word “*tengah*” is part of a method or class name, the method or class name has been retained for backwards compatibility.

## **thread**

A path of execution; a process.

*See also* **multithreading**

## **transaction**

1) A complete unit of work that transforms a database from one consistent state to another. In distributed transaction processing, a transaction can include multiple units of work performed on one or more systems.

2) A logical construct through which applications perform work on shared resources (for example, databases). The work done on behalf of the transaction conforms to the four ACID properties: atomicity, consistency, isolation, and durability.

*See also* **ACID properties**.

## **transaction branch**

A resource manager’s internal unit of work in support of a distributed transaction. Each transaction identifier that the transaction manager provides to a resource manager identifies both a global transaction and a specific branch.

*See also* **distributed transaction, resource manager, and transaction manager**.

## **transaction manager**

A system software component that manages global transactions on behalf of application programs. A transaction manager coordinates commands from application programs to start and complete global transactions by communicating with all resource managers that are participating in those transactions. When resource managers fail during global transactions, transaction managers help resource managers decide whether to commit or roll back pending global transactions.

*See also* **resource manager**.

---

**tunneling**

A method for transmitting data from a network by using a different network.  
*See also* **HTTP**.

**two-phase commit (2PC)**

A method of coordinating a single transaction across more than one resource manager. It guarantees data integrity by ensuring that transactional updates are committed in all of the participating databases, or are fully rolled back out of all the databases, reverting to the state prior to the start of the transaction.

**two-tier**

A client/server relationship.  
*See also* **client/server**.

**▲ Back To Top****U****Unicode**

A 16-bit character encoding scheme used to display, process, and exchange text written in most of the world's languages. In Java, *Strings* are, by default, Unicode strings.

*See also* [Unicode Consortium](#)

**uniform resource locator (URL)**

Used for identifying and locating resources over the Internet. The current URL guidelines are available from the [World Wide Web Consortium](#).

*See also* **HTTP**.

**URL**

*See* **uniform resource locator (URL)**

**user**

An individual Principal. An individual's permissions (both negative and positive) override any permissions that are granted to a group of which the user is a member.

*See also* **realm, permission, group** and **principal**.

---

▲ **Back To Top**

V

**vendor client library**

The native layer that translates function calls from the client to the wire protocol used by the database.

**virtual machine (VM)**

Usually refers to the Java Virtual Machine (JVM).

*See also* **JVM**.

▲ **Back To Top**

W

**WAN**

*See* **wide area network (WAN)**.

**WAP Gateway**

A bridge between a mobile network containing mobile clients and a computer network containing application servers. A WAP Gateway typically includes a protocol gateway that translates requests from the WAP protocol stack to the WWW protocol stack (HTTP and TCP/IP), and content encoders and decoders that translate Web content into compact encoded formats.

*See also* **Wireless Application Protocol (WAP)**.

**wide area network (WAN)**

Unlike a LAN, which is usually private and may not allow access to other networks, a WAN is usually a connection of many networks and often operates over long geographical distances. The Internet is the largest WAN.

*See also* **L10N**.

**WebLogic Beans (WebLogic's distributed server-side Beans)**

An extension of the [JavaSoft JavaBean](#) component architecture, used for distributing JavaBeans and dramatically simplifying the process of assembling distributed applications.

*See also* **JavaBeans** and Using WebLogic's distributed server-side Beans.

---

### **WebLogic cluster**

Fail-over, replicated services shared among multiple WebLogic Servers acting in an interdependent way to support failsafe, high-availability operations for clients. A WebLogic Cluster has a single, cluster-wide JNDI tree for client access to service objects.

*See also* **JNDI**.

### **WebLogic COM**

The WebLogic bridge for accessing Microsoft COM objects from within WebLogic RMI.

*See also* **WebLogic RMI** and Using WebLogic COM

### **WebLogic EJB**

A component architecture for building distributed, object-oriented business applications in Java. The EJB architecture addresses the development, deployment, and runtime aspects of an enterprise application's lifecycle.

An Enterprise JavaBean (EJB) encapsulates business logic inside a component framework that manages the details of security, transaction, and state management.

WebLogic EJB is an implementation of the [JavaSoft specification](#) for transactional components.

*See also* **Java Enterprise API** and Using WebLogic Enterprise JavaBeans.

### **WebLogic Events**

WebLogic's event management and notification service that works within the WebLogic framework.

*See also* Using WebLogic Events

### **WebLogic Express**

A product configuration for database application development in Java. WebLogic Express includes your choice of a two-tier WebLogic jDriver, security, HTTP servlet support, name and directory services, as well as WebLogic JDBC, a multi-tier implementation of the [JDBC](#) specification.

*See also* **WebLogic JDBC**, **WebLogic jDriver** and **JDBC**.

---

## WebLogic File Services

WebLogic's file management facility for reading and writing files to a WebLogic server. WebLogic File services provides server-side read and write access for WebLogic clients using standard `java.io` methods.

*See also* Using WebLogic File Services.

## WebLogic home

WebLogic home is the root directory for the WebLogic distribution and is the directory that contains the individual server directories.

## WebLogic HTTP Servlets

A WebLogic service, part of WebLogic's built-in webserver, that provides support for HTTP servlets that conform to the [JavaSoft Java Servlet API](#).

*See also* **servlet** and Using WebLogic HTTP Servlets.

## WebLogic jDriver

Two-tiered drivers consisting of the native, Type 2 WebLogic jDriver for Oracle and the all-Java, Type 4 WebLogic jDriver for Informix and WebLogic jDriver for Microsoft SQL Server.

*See also* **JDBC**.

## WebLogic JDBC

A pure-Java, multitier implementation of the **JDBC** specification that is included in both the WebLogic Express and the full WebLogic product packages.

*See also* **WebLogic Express**, **JDBC** and Using WebLogic JDBC.

## WebLogic JHTML

A service that is no longer implemented in WebLogic Server that provided for automatic generation of servlets from standard HTML pages that include Java code. JHTML has been replaced by JavaServer Pages (JSP).

## WebLogic JNDI

A WebLogic service, implementing the [JavaSoft JNDI](#) standard, that provides for transparent bind and lookup of objects in a WebLogic Server or Cluster.

*See also* **JNDI** and Using WebLogic JNDI

---

### **WebLogic Remote**

WebLogic Remote supports T3Servlets, a simple, efficient distributed computing model supported in WebLogic, similar to RPC, that allows you to offload computation-intensive or privacy-sensitive tasks from clients to WebLogic.

*See also* Using WebLogic Remote

### **WebLogic RMI**

WebLogic's implementation of the [JavaSoft RMI](#) specification, which provides standards-based distributed object computing within the WebLogic framework.

*See also* **RMI** and Using WebLogic RMI.

### **WebLogic Server**

WebLogic's standards-based, pure-Java application server, for assembling, deploying, and managing distributed Java applications. WebLogic manages application components and DBMS connections to ensure security, scalability, performance, and transaction integrity. It includes support for distributed component services and enterprise database access, including Enterprise JavaBeans, RMI, distributed JavaBeans, and JDBC.

*See also* **Java Enterprise APIs**, **RMI** and **JDBC**.

### **WebLogic SSL**

A WebLogic facility that provides encrypted, authenticated client access to WebLogic via Secure Socket Layer (SSL) and support authentication and access via access control lists (ACLs).

*See also* **SSL**, **ACL** and Using WebLogic SSL.

### **WebLogic Workspaces**

Each WebLogic client has a unique, named context within the WebLogic Server where the client can store and retrieve arbitrary objects and make objects available to other clients for sharing.

*See also* Using WebLogic Workspaces

### **WebLogic service advertisement protocol (WSAP)**

Used within WebLogic to bind and unbind objects in the replicated, cluster-wide [JNDI](#) tree. WSAP allows WebLogic Servers to dynamically join and leave a cluster; it tracks which services are available in the cluster, and it also mediates conflicts for services offered at the same node in the cluster-wide JNDI tree.

*See also* **JNDI**.

---

### **Wireless Application Environment (WAE)**

WAE defines the framework for network-neutral, wireless applications for narrow-band devices. Two of the main components of WAE are Wireless Markup Language (WML) and WMLScript (WMLS).

*See **Wireless Markup Language (WML)** and **WMLScript (WMLS)**.*

### **Wireless Application Protocol (WAP)**

A set of protocols developed by the WAP Forum that allow for the development of Internet and Web-based services for mobile phones and other mobile devices.

### **Wireless Markup Language (WML)**

An XML-based language specifically designed to interface with the micro-browsers that exist in WAP-enabled devices. The Wireless Markup Language Specification defines the tags and structure of a WML document.

*See **Wireless Application Protocol (WAP)** and **XML**.*

### **WMLScript (WMLS)**

WMLScript provides general scripting capability to the WAP architecture and is designed to overcome the limitations of narrowband communication and mobile clients.

*See also **Wireless Application Protocol (WAP)**.*

### **Back To Top**

### **X**

### **XA**

The XA interface within JTA allows a transaction manager to control transaction boundaries for operations performed by multiple resource managers using the two-phase commit X/Open XA protocol.

*See also **Java Transaction API (JTA)**, **resource manager**, and **transaction manager**.*

### **XML**

*See **extensible markup language (XML)**.*

### **Back To Top**

---

Y

 **Back To Top**

Z