JBoss Application Server.

STANDARDIZE ON JBOSS AS TODAY.

QUALITY SUPPORT FROM THE SOURCE.

JBoss Application Server (JBoss AS) is the market-leading, open source Java 2 Enterprise Edition (J2EE^{TM}) application server, delivering a high-performance, enterprise-class platform for e-business applications. Combining a robust, yet flexible, architecture with a no-cost open source software license, JBoss AS has quickly become the most popular middleware system for developers, independent software vendors (ISVs), and enterprises alike.

JBoss AS not only offers the performance and reliability expected from an industrial-strength application platform, it enables customers to scale technically and economically as well. With its revolutionary service-oriented architecture (SOA), JBoss AS ensures that applications are developed, deployed, integrated, and managed in a consistent manner.

Furthermore, JBoss AS provides an aspect-oriented architecture (AOA) layer on top of its SOA foundation, further simplifying the way developers interact with middleware services, and enabling IT organizations to deliver greater value in less time. Faster development, combined with a zero-cost software license, greatly reduces overall business costs. Best of all, support for JBoss AS is delivered by full-time JBoss product experts and JBoss Authorized Service Partners, making JBoss AS the safe choice among enterprises for powering their mission-critical applications.

Highlights and Key Values

Open-source, Zero-cost Product License

JBoss AS is licensed under the Lesser Gnu Public License (LGPL) and is, therefore, free to download and use for development and production deployment—regardless of the size of your deployment. The license also allows ISVs to embed and distribute JBoss AS free of charge

Built for Standards

JBoss AS not only meets the needs of today's standards, but also is best positioned to handle the standards and services of tomorrow by the nature of its forward-looking, modular architecture. As new standards emerge, they can plug into JBoss AS in a consistent and predictable manner, which ensures that developers can quickly leverage these new technologies.

Enterprise-class Reliability

JBoss AS provides a proven foundation for mission-critical applications that require superior performance and scalability. Its high-availability services provide the clustering, caching, fail-over, load balancing, and distributed deployment features expected in a best-of-breed platform. Furthermore, you can develop and deploy your applications with confidence knowing that JBoss AS is J2EE compliant—having passed the 23,000 tests of the Test Compatibility Kit.

Embeddable, Service-Oriented Architecture

JBoss AS is assembled from a set of independent, yet cooperating, components and services which are neatly packaged and fully hot-deployable. It is architected to be seamlessly embeddable in applications, and the nature of its embedding is completely customizable to the requirements of the application itself. Only the critical and necessary application server components, therefore, need to be brought along as part of the application's baseline footprint. Developers can also easily create and add their own services to the system, thus ensuring that custom services exhibit the same consistent behavior as the JBoss standard set of services.

Consistent Manageability

With its elegant Java Management Extensions (JMX) based microkernel architecture, JBoss AS not only offers a consistent behavior of modules, but also offers the ability to manage and configure services through any management console that supports JMX. This ensures that applications built on JBoss AS can be managed easily and consistently.

Middleware Services for any Java Object

JBoss AS' aspect-oriented architecture (AOA) provides a modular way of defining and applying cross-cutting concerns, enabling the delivery of enterprise-class functionality, such as persistence, security, and distributed transactional caching, to any plain old Java object (POJO). Since this functionality is applied to POJOs using simple annotations, this greatly improves developer productivity by avoiding lengthy development, compilation, and deployment cycles altogether.

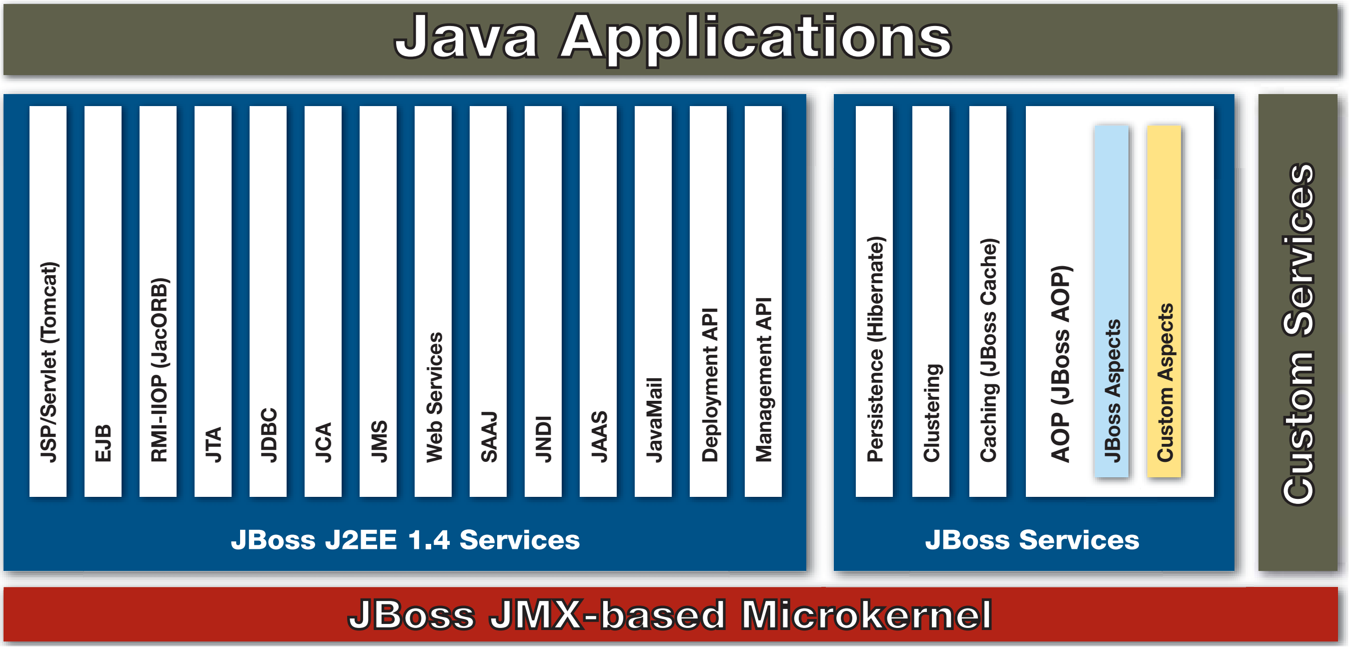
Enhanced Interoperability

JBoss AS fully supports the Web Services standards required in the J2EE 1.4 specification, enabling different applications to work together, even if they are running on different operating systems and hardware architectures, and using different application infrastructures and programming languages. This technology independent interoperability enables IT organizations to maximize the value of their existing solutions, as well as enable enterprise application integration and eliminate corporate silos.

Superior 24x7 Support from the Source

JBoss, Inc. pioneered the Professional Open Source model whereby open source software is backed up with expert support services delivered by the product experts themselves. JBoss, Inc. and our Authorized Service Partners offer a comprehensive set of support offerings to ensure that our products—such as JBoss AS— are the safe choice for your business.

JBoss AS Architecture



Microkernel Layer

At the core of JBoss AS is a microkernel-based server that is extremely small in footprint. By utilizing JMX, the microkernel delivers a lightweight component model that clearly defines the lifecycle, configuration, and management of services and supplies a standard mechanism for assembling service components which ensures that all services are accessed, managed, and integrated in a unified and consistent manner.

Services Layer

On top of the microkernel resides a service-oriented architecture (SOA) that is comprised of a set of packaged and hot-deployable services. Services deployed in JBoss AS can range anywhere from transaction and messaging services to connection pooling and security services. Services can easily be added or removed from the system—thus tuning the overall footprint to suit specific needs. Furthermore, developers can easily extend JBoss AS by creating custom services and packaging them as SARs (Service Archives) that are then individually, hot-deployable to JBoss AS.

Aspect Layer

The aspect layer—based on the aspect-oriented programming (AOP) model—complements the services layer by exposing a wide range of cross-cutting services in a simpler-to-use, aspect-oriented fashion. Developers merely develop POJOs and apply enterprise-level services—such as object persistence, caching, replication, remoteness, transactions, and security—later in the development cycle without having to change a line of Java code. By simply annotating their existing objects, developers can weave extended functionality into their existing applications quickly and easily.

Application Layer

The application layer is where applications built on JBoss AS reside. These applications can leverage the flexible infrastructure of JBoss AS, by utilizing the container services directly (programmatically) or indirectly (AOP annotations). Either way, applications built on JBoss AS benefit from its enterprise-class, production-ready services.

Components and Features

JMX-based Microkernel

JBoss AS not only supports JMX for management extensions – as defined by the J2EE specifications – but also uses it as the basis for achieving the modularity of its microkernel architecture. JBoss AS is founded on a service-oriented, JMX-based microkernel architecture which enables services to plug-in, describe and advertise themselves, as well as recognize, engage, and utilize other services that are plugged-in. JBoss AS is essentially a set of MBean (Managed Bean) components that reside in an MBean server, which enables these MBean components (or “services”) to communicate between each other. Services include both core-level services, such as class loading, naming, and logging, as well as advanced services, such as transactions, fail-over, and security.

Aspect-oriented Framework

JBoss embeds JBoss AOP—an aspect-oriented framework—that works in concert with object-oriented programming (OOP) to provide cross-cutting, declarative services (such as persistence, logging, and caching) to object-oriented POJOs. JBoss AOP improves the readability, maintainability, and reusability of your code, and provides a great way to expand the Java language in a clean and elegant manner. JBoss AOP is not only a framework, but also a prepackaged set of aspects that are applied via annotations, pointcut expressions, or dynamically at runtime. JBoss AS includes a variety of pre-packaged aspects for such things as caching, synchronous communication, transactions, security, and remoting

Web Application Services

JBoss AS supports the latest JSP and Servlet specifications and implements many other high-performance, advanced features to meet the demands of today’s most complex Web applications.

- Embedded Tomcat 5 container provides industry-leading JSP 2.0/Servlet 2.4 specification support
- Integration between Tomcat and other JBoss AS services provides increased reliability, scalability, and availability
- Integration with Apache Web Server via mod_jk provides the best Web Server, Web Container, and Application Server combination on the market

Enterprise Services

JBoss AS includes the following enterprise-class features that are required to satisfy the needs of today’s scalable, distributed, transactional, and fault tolerant applications.

- EJB 2.1 support, including:
  - > Stateful and stateless Session Beans
  - > BMP (Bean Managed Persistence) Entity Beans
  - > CMP (Container Managed Persistence) Entity Beans
  - > Message driven beans (MDBs)
  - > Interoperability using RMI-IIOP or JAX-RPC
  - > Ability to expose stateless session beans and MDBs as Web Services
  - > Support for sending and receiving messages via Web Services
  - > Easy provisioning and hot deployment of EJB and JMX-based Web Services.
  - > Access to EJBs from external CORBA objects

- Bundled ORB (JacORB) that supports:
  - > Interoperability between EJB 2.1 and CORBA objects
  - > Full support for CORBA 2.3
  - > Portable Object Adaptor (POA)
  - > Asynchronous Method Invocations (AMI)
  - > Portable interceptors
  - > Dynamic Invocation Interface (DII), Dynamic Skeleton Interface (DSI), and Dynamic Management of Anys (DynAny)

- Built-in Java naming and directory interface (JNDI) support
- Built-in JTA for two-phase commit transaction support

Web Services and Interoperability

One of the major advances of J2EE 1.4 over J2EE 1.3 was the addition of Web Services standards, more complete XML processing, and enhanced Web Services interoperability.

The complete incorporation of all of these Web Services standards enables applications built on JBoss AS to dynamically interact with Web Services—using Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Universal Description, Discovery and Integration (UDDI), and XML—and benefit from the reliable, scalable, and production-ready infrastructure provided by JBoss AS.

O/R Mapping and Persistence Services

JBoss AS bundles and integrates Hibernate, the de-facto leader in O/R mapping and persistence, to provide the following features:

- Flexible object / relational mappings
- Transparent persistence without bytecode processing
- Human-readable Object/Relational mapping definitions that can be generated via XDoclet
- Simple APIs
- Powerful Object-oriented query language expressed in a SQL-like language
- Automatic primary key generation
- Threadsafe, non-blocking Hibernate Dual-Layer Cache Architecture (HDLCA)
- Ultra-high performance

Integration and Messaging Services

JBoss AS supports the latest connector and messaging specifications in order to provide the base infrastructure needed to support the most demanding integration requirements.

Java Database Connectivity (JDBC) 3.0 support
J2EE Connector Architecture (JCA) 1.5 support for connection pooling, transaction, and security context propagation for any JCA-compliant resource adapters, including databases, legacy systems, and JMS servers.
JMS 1.1 support that allows asynchronous delivery of messages in distributed systems with optional quality of services parameters such as persistence, guaranteed delivery, and transactions. Supports both queue and topic messaging styles and offers high-availability features that allow JMS applications to be deployed in a clustered environment.

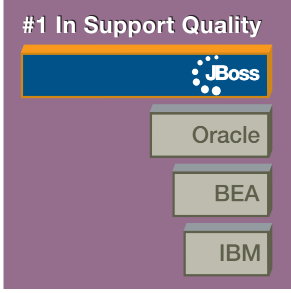
High Availability Services

JBoss AS provides the following distinctive high-availability services:

- Clustering, fail-over, load balancing, and distributed deployment features provide the means to deploy large scalable robust applications. Features include:
  - > Automatic discovery of nodes
  - > Cluster-wide replicated JNDI context
  - > Failover and load balancing for JNDI, RMI, and all EJB types
  - > Stateful session bean state replication
  - > HTTP session replication
  - > High availability JMS
  - > Farming/distributed deployment of JBoss AS components; deploying on one node deploys on all nodes
- Embedded JBoss Cache that provides a replicated, transactional, tree-structured cache which enables easy clustering of data on a JBoss AS grid. JBoss Cache:
  - > Significantly increases performance via local access to data
  - > Enables session data to be replicated thereby providing an always-on, high availability cache of data on your grid
  - > Provides fine-grained replication for POJOs—an industry first!

Security Services

JBoss AS implements the standard J2EE Authentication and Authorization (JAAS) security model, but also supports extended security models to provide support for context-sensitive authorization. Moreover, support for pluggable authentication modules (PAMs) enable integration with existing authentication frameworks in your enterprise.



Platform and Standards Support

Minimum System Requirements

- JDK 1.4 or higher (1.4.2 is recommended)
- 512 MB RAM
- 100 MB hard disk space
- 400 MHz CPU

Supported Operating Systems

Since JBoss AS is 100% pure Java, it is interoperable with most operating systems that are capable of running a Java Virtual Machine (JVM); including Windows, UNIX, Linux, and others.

Supported Databases

JBoss AS is interoperable with any JDBC-compliant database; including Oracle, SQL Server, DB2, and others.

Supported Standards

- EJB 2.1
- JSP 2.0 and Servlets 2.4
- CORBA 2.3.1
- JDBC 3.0
- J2EE Management API 1.0
- J2EE Deployment API 1.1
- Java API for XML-based RPC (JAX-RPC 1.1
- Java API for XML Registries (JAXR) 1.0
- Java Naming and Directory Interface (JNDI) 1.2.1
- JavaBeans Activation Framework (JAF) 1.0
- Java Management Extension (JMX) 1.2
- Java Transaction API (JTA) 1.0.1B
- J2EE Connector Architecture (JCA) 1.5
- Java API for XML Processing (JAXP) 1.2
- SOAP with Attachments API for Java (SAAJ) 1.2
- Web Services 1.1
- Java Authentication and Authorization Service (JAAS) 1.0
- JavaMail 1.3
- Java Authorization Service Provider Contract for Containers (JACC) 1.0

Consoles and Tools

JBoss AS addresses the needs of application developers, architects, and administrators by providing easy to use tools and consoles.

JBoss Web Console provides ability to:

- View user-defined domain and MBeans
- Get J2EE Management (JSR 77) statistics for Servlets, EJBs, and deployment descriptor views
- Graph any numeric JMX attribute dynamically
- Monitor JMX attributes
- Receive email alert notifications based on specified thresholds
- Record snapshots of JMX attributes over time
- View into the microkernel and display all registered and active MBean services
- Display JNDI tree
- Generate thread dump
- Manage the deployment scanner
- Shut down JBoss AS

JBoss IDE is available separately and provides:

- World-class development environment via integration with Eclipse 3
- Integrated debugging and monitoring of JBoss servers and the controlling of their life cycles
- Easy configuration of the packaging layout of archives (packed or exploded)
- Simple deployment of the packaged and/or exploded archive to a JBoss AS server
- Several J2EE wizards to ease and simplify the J2EE development
- JSP, HTML, and XML editors with syntax highlighting
- Comfortable and sophisticated support for XDoclet
- Integrated support for JBoss AOP and the development and use of aspects
- Integration of XDoclet, Packaging, and Deployment with Eclipse Job API

JBoss Support Services

Your success greatly depends upon your ability to learn, implement, and ultimately support technology such as JBoss AS. That's where our team of product experts comes into play. JBoss, Inc. and our network of JBoss Authorized Service Partners can provide the training, consulting, and professional support that you will need whether you are testing a proof of concept, deploying several mission-critical applications, or rolling out JBoss open source products into production across your entire enterprise.

More information regarding JBoss services can be found at:
http://www.jboss.com/services/index

We want your JBoss experience to be a positive one. If at any time you would like to contact JBoss, Inc. regarding our support, training, or consulting options, please send us an e-mail at: sales@jboss.com, call us at our US Headquarters +1 (404) 467-8555 or contact our European Headquarters at +41 32 720 9260.

Getting Started
Contact us to learn more about JBoss Inc. support services. Please visit us at **www.jboss.com**, e-mail us at **sales@jboss.com**, or contact one of our sales offices:

US Headquarters:
JBoss Inc.
3340 Peachtree Road, NE
Suite 1200
Atlanta, GA USA 30326
Phone (404) 467-8555
Fax (404) 948-1496

European Headquarters:
JBoss Europe
Place Numa-Droz 2, CP
2522
CH - 2001 Neuchâtel
Switzerland
Phone +41 32 720 92 60
Fax +41 32 720 92 61

UK Sales Office:
JBoss UK
Regents Place
338 Euston Road
London NW1 3BT
United Kingdom
Phone: +44 20 7543 6910
Fax: +44 20 7544 8401