

Chapter 1

GETTING STARTED

*Get on the
Fast Track!*



TM

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- WAS: WebSphere Application Server.
- WebSphere Application Server packaging.
- New features.
- Web services.
- WebSphere architecture.
- Web Administrative Console and application.
- Applications.
- Session database.
- Security server.
- Web services engine.
- WAS network deployment.
- Managed processes.

1 WAS: WebSphere Application Server

WebSphere Application Servers, WAS, provides the core software to deploy, integrate, and manage e-business applications. WebSphere Application Servers have been designed to be J2EE 1.3-compliant and utilize the feature set associated with scalability, security, standardization, and performance.

IBM has extended the J2EE programming model and offers multiple WAS configurations. As operational requirements change, migration can be made to higher qualities of service offered by other configurations.

WebSphere Application Servers include:

- J2EE 1.3 support.
- JCA: J2EE Connector Architecture support.
- Messaging support through the JMS API.
- Asynchronous processing of messages using EJB 2.0 message-driven beans and container-managed messaging.
- Enhanced security support.
- Balancing and clustering.
- Edge services which provide content-based routing and caching.
- Browser-based administration based on JMX-Java Management eXtension interfaces.
- Web services for building, publishing, and managing application services.
- Support for private IBM UDDI Registry and IBM Web Services Gateway.
- Support across multiple platforms.

2 WAS Product Packaging

WebSphere Application Server V5 comes in several configurations:

- WebSphere Application Server Express
- WebSphere Application Server Base Configuration
- IBM WebSphere Application Server Network Deployment
- IBM WebSphere Application Server Enterprise

2.1 WebSphere Application Server - Express

WebSphere Application Server - Express V5.0 provides development tools and application server in a single integrated package.

There is a simplified programming model for creating new web applications and converting existing static applications to dynamic applications. There is a migration path to higher-end WebSphere Application Server and WebSphere Studio configurations.

The following is contained in a single installable package:

- Application Server - Express Version 5.0.
- WebSphere Studio Site Developer Version 5.0.
- Preloaded samples.

2.2 WebSphere Application Server - Base Configuration

The WebSphere Application Server base configuration provides an application deployment environment for a single-server. It contains a base application server that supports the full J2EE environment.

The following standards are supported:

- J2EE 1.3
- EJB 2.0 and EJB 1.1 support
- JRE 1.3.1
- Servlet 2.3
- JSP 1.2
- JTS/JTA 1.0 with distributed transactions
- JMS 1.0.2 with native provider and MQ plug-in
- JDBC 2.0 with 2PC across heterogeneous databases
- JNDI 1.2 for EJB lookup and CosNaming
- RMI/IIOP 1.0
- JavaMail/JAF 1.2 plus Domino support
- SSL Security 2.0: JSSE and JCE
- XML JAXP 1.0: XML in EJBs
- J-IDL/CORBA IIOP 1.2
- J2C 1.0 bean and container managed
- LDAP with SecureWay, iPlanet, Active Directory
- HTTP 1.1 across multiple web servers
- SOAP 2.2.2 for web services
- SOAP-SEC 1.0 technical preview
- COM/ASP support with Java wrapping and proxy
- JMX 1.0
- XML4J 4.0
- XSL 2.3

2.3 WebSphere Application Server Network Deployment

WebSphere Application Server Network Deployment adds non-programming enhancements to the features in the base configuration and facilities for running applications on multiple servers and multiple physical nodes.

WebSphere Application Server Network Deployment provides:

- Deployment Manager for centrally managing a number of different application server instances and clustering for workload management and failover.
- Edge Components, including Load Balancer, Caching Proxy, and IBM HTTP Server.
- A private UDDI Registry for deployment of internal web services applications and a web services gateway for external applications that request web services that use security features for accessing an internal web services provider application.

2.4 WebSphere Application Server - Enterprise

WebSphere Application Server Enterprise provides the features in Network Deployment, plus programming model extensions for sophisticated application designs.

Facilities are provided for application adapters, application workflow composition and choreography, extended messaging, dynamic rules-based application adaptability, and internationalization, and asynchronous processing.

WebSphere Application Server Enterprise:

- Simplifies build-to-integrate tasks by reusing resources and automating business processes.
- Accelerates large-scale application development in the J2EE environment.
- Enables real-time application flexibility and the adaptation of the application to the environment.

Features of WebSphere Application Server Enterprise include:

- SOA: Service-Oriented Architecture

SOA represents all software assets as services - legacy applications, packaged applications, J2EE components, and Web services.

- Integrated J2EE-based workflow

The integrated J2EE-based workflow offers flow-based application development. Different software assets and services can be reused and composed into a business process.

- Advanced transactional connectivity

WebSphere Application Server Enterprise transaction provides support for enterprise applications.

Feature Supported	Explanation
Dynamic application adapter	Supports transactions for adapters, built on open standards, to connect to other systems
Last participant	Automated coordination for transactions that include two-phase commit resources and a single one-phase commit resource.
ActivitySession service	Extend the scope of and group multiple local transactions.
CORBA C++ Support	Enables C++ clients to invoke J2EE components using the CORBA technology.

- Optimized application performance

Application profiling techniques are used for tuning performance without changing the code.

- Programming model extensions

Extension	Description
Extended messaging	Applications can be created that integrate with other systems through a messaging infrastructure. There is automated support for inbound and outbound messaging by hiding the complexities of the messaging API.
Asynchronous beans	Performance enhancements for resource-intensive tasks by enabling single tasks to run as multiple tasks.
Startup beans	Automatic execution of business logic when the application server starts or stops.
Scheduler service	Schedule processes in a timely manner for batch processing or better computing resource utilization.
Object pools	Multiple instances of objects can be reused, reducing the overhead associated with instantiating, initializing, and garbage-collecting the objects.
Internationalization service	Automatic recognition of the calling client's time zone and location information.
Work areas	A "global variable-like" ability to share information across distributed applications.

- Real-time application flexibility

Applications require the capability to make changes and adopt to current conditions without changing the application code.

Business Rule Beans	Provide a real-time framework for defining, executing, and managing business rules that encapsulate business policies.
Dynamic query	Allows the EJB container, using the EJB Query Language, to submit queries that select, sort, join, and perform calculations on application data at runtime.

2.5 WebSphere Development Packaging

WebSphere also has packages for development use.

Three packages have been designed for software providers:

- WebSphere Application Server for Developers.
- WebSphere Application Server Enterprise for Developers.
- WebSphere Application Server for Developers - Trial Use Only

2.6 Features by Configuration

The supported features in each configuration are:

	IBM WebSphere Application Server Express	IBM WebSphere Application Server, Base	IBM WebSphere Application Server, Network Deployment	IBM WebSphere Application Server, Enterprise
J2EE 1.3 support	Partial	Full	Full	Full
Servlet 2.3	Yes	Yes	Yes	Yes
JSP 1.2	Yes	Yes	Yes	Yes
EJB 2.0	No	Yes	Yes	Yes
J2EE Application Client	No	Yes	Yes	Yes
JavaMail	Yes	Yes	Yes	Yes
J2C: Java2 Connector	No	Yes	Yes	Yes
J2C Resource Adapters	No	Yes	Yes	Yes
JDBC	Yes	Yes	Yes	Yes
Embedded Messaging Support	No	Yes	Yes	Yes
XML support	Yes	Yes	Yes	Yes
Web services	Yes	Yes	Yes + Web Services Gateway & Private UDDI Registry	Yes + Web Services Gateway & Private UDDI Registry
Workload management	No	No	Yes	Yes
Dynamic caching	No	Yes	Yes	Yes
Security Authentication	Local OS	<ul style="list-style-type: none"> • Local OS • LDAP • Custom Registry 		
Tools	WebSphere Studio tools	<ul style="list-style-type: none"> • Application Assembly Tool – AAT • Log Analyzer • Tivoli Performance Viewer • Performance Monitoring Instrumentation - PMI • Migration tools 		

	IBM WebSphere Application Server Express	IBM WebSphere Application Server, Base	IBM WebSphere Application Server, Network Deployment	IBM WebSphere Application Server, Enterprise
Database access	<ul style="list-style-type: none"> • IBM DB2 • SQL Server • Oracle • Informix • Sybase 			
Administration	Studio-based	Browser-based, single-server	Browser-based Single point for admin of multiple servers	
JMS supported message-driven beans	No	Yes	Yes	Yes
Failure bypass	No	No	Yes	
Cluster support	No	No	Yes	Yes

3 New Features

Improvements and additions have been made to WebSphere Application Server V5.

3.1 J2EE 1.3 Compliance

The J2EE specification provides:

- APIs that developers must use to build applications.
- Runtime services that must be provided by an application server.
- The communication protocols that the servers and applications running on those servers must communicate on.

WebSphere Application Server V4 supported J2EE 1.2. WebSphere Application Server V5 updates that support to J2EE 1.3.

J2EE 1.3 introduces enhancements in a number of areas and includes support for Servlet 2.3, EJB 2.0, and JSP 1.2.

3.2 JMX Framework: Administration Model

WebSphere Application Server V5 provides a new administration model based on the JMX framework.

JMX can be used to wrap all of the hardware and software resources in Java and expose them in a distributed environment.

JMX provides a mapping framework for integrating existing management protocols, such as SNMP, into JMX's own management structures.

The benefits in using a JMX administrative model are:

- Improved availability.
- Reduction of interdependencies among process.
- Increased usability of resources.
- Adoption of a standardized framework for resource management.

Configuration data is now stored directly into XML documents, eliminating the need for a relational database repository and the XMLConfig tool.

The administrative console is web-based and can be viewed using a Web browser. The wsadmin command line scripting tool has been provided and serves to replace the WSCP tool from previous releases.

3.3 WLM: Workload Management

Enhancements have been made to WLM: Workload management.

Enhancement	Description
Scalability	WebSphere can be configured to serve more users. WLM enables the applications running under WebSphere to be scaled to any number of machines; increasing the amount of requests the applications can serve.
Load balancing	Applications can share the workload fairly across machines participating in the enterprise.
Availability	WLM enables the identification of systems that will take over system processing in case of server failure. WebSphere utilizes the concept of cells, nodes, and clusters for enabling workload management.

4 Web Services

Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Once a web service is deployed, other applications and other web services can discover and invoke the service.

Web services are independent of specific programming languages or operating systems. They rely on pre-existing transport technologies and standard data encoding techniques for their implementation.

The web services approach to programming is built upon the concept of building applications by discovering and invoking network-available applications to accomplish some task.

4.1 Web Services Support

WebSphere Application Server V5 has improved SOAP support and provides a private UDDI and web services gateway. WebSphere Application Server V5 provides a web services gateway that is used to integrate with other web services in heterogeneous web services environments.

WebSphere Version Comparison

WebSphere Application Server Version	WebSphere Application Server 5.0	WebSphere Application Server 4.0
SOAP	2.2	2.2
SOAP Client Integration	2.2	None
UDDI	2.0	1.0.4

Code written to one version of the specification will not necessarily be compatible with the runtime environment that uses another specification.

4.2 Private UDDI Registry

The public registry is a UDDI business registry where the services can be published and anyone with access to that UBR will be able utilize those services. There is no validity check done on the web service; it serves to publicly distribute and utilize existing Web services.

A private registry is new to WebSphere Application Server V5. The private UDDI allows customers to become the service provider without having to comply with any additional requirements by the platform. The private UDDI can be used for publishing their services directly; rather than have a third-party broker provide that service.

4.3 Web Services Gateway

The Web Services Gateway component provides a framework for invoking web services between Internet and intranet environments. It provides the ability to enable the use of web services across administrative boundaries. The main benefit to the gateway is that it allows interoperability between web services deployed on different vendor platforms. The gateway abstracts the vendor details and publishes and serves web service requests based on the standard protocols and transports.

A web services gateway can leverage the existing infrastructure by creating web services as a means of communicating between systems in the enterprise.

The gateway is deployed as a J2EE application and can be managed by the WebSphere administrator.

4.4 Web Services Security

WebSphere Application Server V5 provides security services-specific web service implementations. The Web services security provides security handlers for the client and server to interface with WebSphere security services.

The security handler intercepts SOAP messages and modifies them based on the target service security requirement supplied through configuration files in WebSphere. The server will validate each request, based on the security configuration. In V5, there is only support for request messages.

The server signs messages with digital signatures and no encryption is supported. This service is fully integrated into WebSphere security.

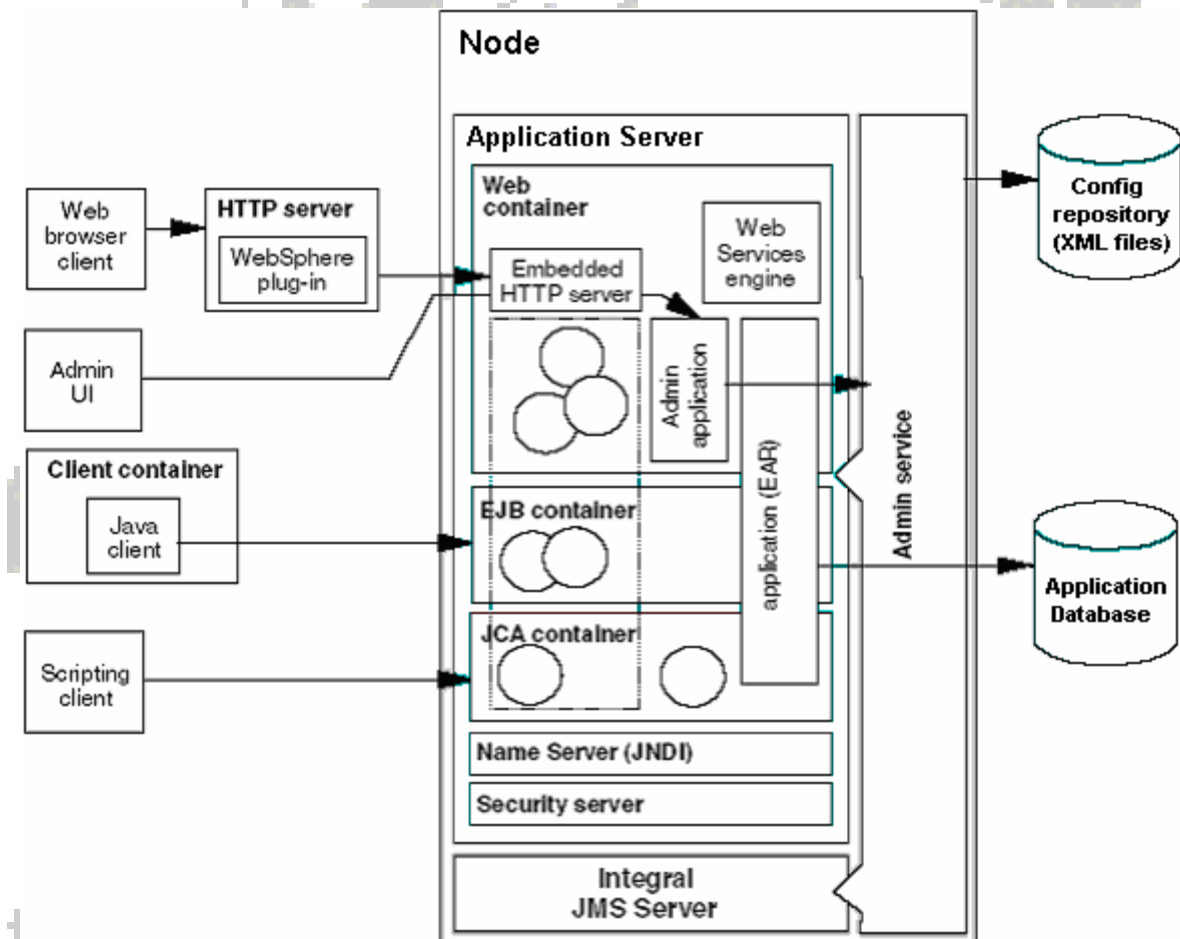
5 WebSphere Architecture

With WebSphere Application Server Network Deployment V5; a network deployment environment consists of multiple WebSphere Application Server V5 nodes grouped into a single administrative domain.

5.1 WebSphere Application Server

A base WebSphere Application Server V5 configuration includes only the application server process.

- There is no node agent or Deployment Manager involved in this configuration.
- No coordination between application server processes is supported in the base configuration; each application server instance has to be separately administered.



5.2 Components

Component	Description
Node	Logical grouping of WebSphere managed server processes that share common configuration and operational control.
Configuration Repository	Holds copies of all the individual component configuration documents. Starting with Version 5 all configuration information is stored in XML files. The application server's admin service manages the configuration and ensures that it is consistent during the runtime.
Application Server	<p>Is the primary component of WebSphere. It runs in a Java virtual machine – JVM and provides the runtime environment for the application's code. The application server provides containers that specializes in enabling the execution of specific Java application components. There are three containers in the application servers: web container, EJB container, and J2C container.</p> <p>The Application Server provides other services besides the containers:</p> <ul style="list-style-type: none"> • ORB - Object Request Broker • JNDI - Name service, security service • JAAS and Java 2 security • JMX Admin service, trace service • PMI - Performance Monitoring Interface • Transaction management • Messaging interfaces - JMS • E-mail interfaces - JavaMail • Database connection - JDBC • Connection pooling <p>Application servers can access a shared database for storing data.</p>
Web Server and Web Server Plug-in	The WebSphere Application Server works with a web server to handle requests for dynamic content from web applications. The web server and application server communicate using the Web server plug-in. The web server plug-in uses an XML configuration file to determine whether a request should be handled by the web server or the application server. It uses the standard HTTP protocol to communicate with the application server, but can also be configured to use secure HTTPS, if required.
Embedded HTTP Server	The HTTP server is embedded within the application server and is useful for testing or development purposes. In terms of performance and security, web server and web server plug-in for the web server needs to be used in the production environment.
Virtual hosts	Is a configuration which enables a single host machine to resemble multiple host machines. It allows a single physical machine to support several independently configured and administered applications and is not associated with a particular node. It is a configuration, rather than a "live object". Each virtual host has a logical name and a list of one or more DNS aliases by which it is known.

6 Web Container

The web container processes servlets, JSP files and other types of server-side includes. Each web container automatically contains a single session manager. When handling servlets, the web container creates a request object and a response object, then invokes the servlet service method. The web container invokes the servlet's destroy method when appropriate and unloads the servlet, after which the JVM performs garbage collection.

The web container runs an embedded HTTP server for handling HTTP(S) requests from external web server plug-ins or web browsers. A web container configuration provides information about the application server component that handles servlet requests forwarded by the Web server. Each application server runtime has one logical Web container, which can be modified, but not created or removed.

The administrator specifies Web container properties including:

- Default virtual host.
- Session management properties.
- Number and type of connections between the Web server and the Web container.
- Port(s) on which the Web container listens for incoming HTTP(S) requests.

6.1 EJB Container

The EJB container provides all the runtime services needed to deploy and manage Enterprise Java Beans - EJBs. It is a server process that handles requests for both session and entity beans.

The enterprise beans installed in an application server do not communicate directly with the server; instead, the EJB container provides an interface between the EJBs and the server. The container and the server together provide the bean runtime environment.

The container provides many low-level services, including threading and transaction support. From an administrative viewpoint, the container manages data storage and retrieval for the contained beans. A single container can host more than one EJB JAR file.

6.2 JCA Container

The JCA: Java Connector Architecture container is a component provided by WebSphere Application Server.

The JCA Resource Adapters from EIS vendors can be plugged into, configured, and used by JCA compliant applications.

6.3 Client Application Container

The client application container is a separately installed component on the client's machine. It allows the client to run applications in an EJB-compatible J2EE environment.

There is a command-line executable (launchClient) that is used to launch the client application along with its client container runtime.

Instead of the single global name space used by WebSphere V4.0, WebSphere V5.0 uses a distributed name space. This affects J2EE clients, since each client runs in its own Java virtual machine - JVM. Each client must be specifically configured to bind to the name space root of a chosen application server, node agent or Deployment Manager.

7 Web Administrative Console and Applications

The Web-based administration interface is installed as a standard J2EE 1.3 compliant web application called “adminconsole”.

The administrator connects to the application using a web browser client.

Users assigned to different administration roles can manage the application server and certain components and services using this interface.

In the base configuration, the adminconsole application runs on the application server and can manage only that application server.

In the Network Deployment configuration, it is installed and run on the Deployment Manager only.

7.1 Admin Service

The Admin service runs within each server JVM.

In the base configuration, the Admin service runs in the application server.

In the Network Deployment configuration, each of the following servers hosts an Admin service:

- Deployment Manager
- Node agent
- Application server
- JMS server

The Admin service provides the functions for manipulating configuration data for the server and its components. The configuration is stored in a repository; a set of XML files is stored in the server's file system.

Admin services has a course-grained security control and filtering functionality, providing different levels of administration to certain users or groups using the following admin roles:

- Administrator
- Monitor
- Configurator
- Operator

7.2 Scripting Client

The scripting client wsadmin provides flexibility over the Web-based administration application. It allows administration using the command-line interface. The scripting client not only makes administration quicker, but helps to automate the administration of multiple application servers and nodes using scripts.

The scripting client uses the BSF - Bean Scripting Framework, which allows a variety of scripting languages to be used for configuration and control.

7.3 JMS Server

The embedded WebSphere JMS provider uses a JMS server to implement the integrated messaging functions. It supports point-to-point and publish/subscribe styles of messaging and is integrated with the transaction management service.

The JMS server is used for:

- Support of message-driven beans.
- Messaging within a WebSphere cell.

In the base configuration, the JMS server runs in the same JVM as the application server.

In the Network Deployment configuration, the JMS server is separated from the application server and runs in a separate dedicated JVM.

7.4 Applications

Applications are custom designed and hosted and run by the application server.

An application is packaged into an EAR - Enterprise Application Archive that is deployed to one or more application servers.

Module	Filename	Content
Web module	<module>.war	Servlets, JSP files, and related code artifacts.
EJB module	<module>.jar	Enterprise beans and related code artifacts.
Application client module	<module>.jar	Application client code.
Resource adapter module	<module>.rar	Library implementation code that your application uses to connect to enterprise information systems (EIS).

Application Database

Data storage is an essential part of many applications. The application database runs on a database server in an enterprise system where multiple application servers can share the same database.

Session Database

In a multi-server environment, session information can be stored in a central session database for session persistence. The multiple application servers hosting a particular application need to share this database information in order to maintain session states for the stateful components.

An alternative approach is to use the memory-to-memory session replication functionality in the Network Deployment environment.

7.5 Name Server

Each application server JVM hosts a name service that provides a JNDI - Java Naming and Directory Interface name space. The service is used to register all EJBs and J2EE resources - JMS, J2C, JDBC, URL, JavaMail hosted by the application server.

The JNDI implementation in WebSphere Application Server V5 is built on top of a CORBA: Common Object Request Broker Architecture and CosNaming: naming service.

Each application server JVM hosts a security service that uses the security settings held in the configuration repository to provide authentication and authorization functionality.

8 Web Services Engine

The web services engine does not stand as a separate component. The application server implements numerous APIs for additional services. Web services is provided as a set of APIs in cooperation with the J2EE applications.

The WebSphere web services engine is based on AXIS, and implements the following specifications:

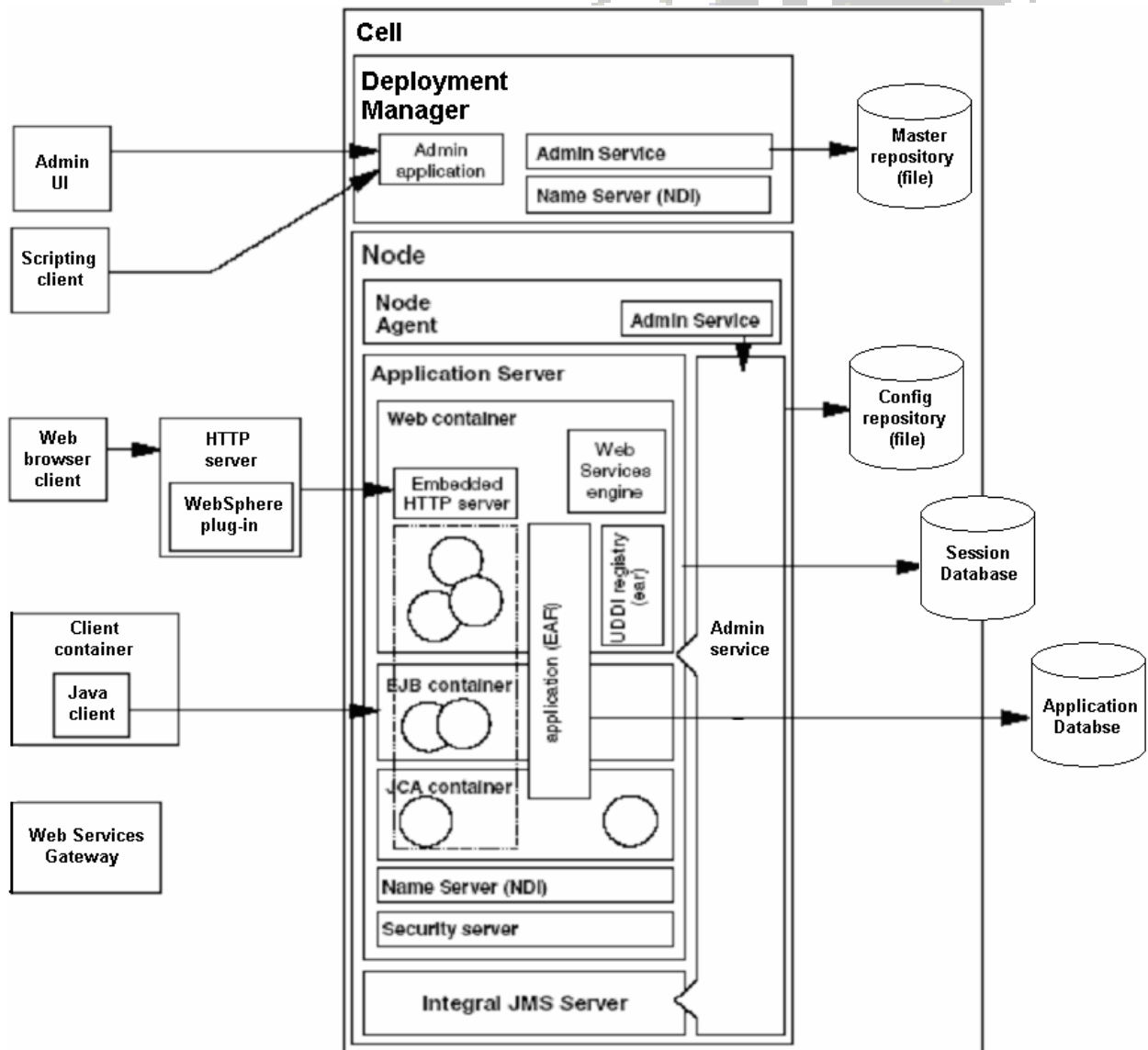
SOAP	Simple Object Access Protocol	Defines the messaging between objects. It is based on the XML and XML schema specification.
WSDL	Web Services Description Language	Describes the services that can be located and used by applications.
UDDI	Universal Description, Discovery and Integration	Enables an application to find services on the network published by service brokers.
WSIF	Web Services Invocation Framework	Provides a standard API for invoking services described in WSDL, no matter how or where the services are provided.

9 WebSphere Application Server Network Deployment

A Network Deployment configuration includes support for multiple nodes, each with a node agent process and several application servers, all coordinated within an administrative cell by the Deployment Manager process.

Clusters of load-balanced application servers can be configured within a Network Deployment cell.

The configuration and application binaries of all components in the cell are managed by the Deployment Manager and synchronized out to local copies on each of the nodes.



Component	Explanation
Cell	Is a grouping of nodes into a single administrative domain. Each node is identified by a logical name for configuration purposes. The configuration and application binaries of all nodes in the cell are centralized into a cell master configuration repository.
Deployment Manager	Provides a single, central point of administrative control for all elements in the cell. It hosts the Web-based administrative console application. Administrative tools that need to access any managed resource in a cell usually connect to the Deployment Manager as the central point of control.
Master Configuration Repository	Contains all of the cell's configuration data. Updates to this repository are performed by the Deployment Manager. The configuration repository at each node is a synchronized subset of the master repository.
Node Agent	Is an administrative process and is not involved in application serving functions. It hosts important administrative functions such as file transfer services, configuration synchronization, and performance monitoring. The node agent aggregates and controls all the managed processes on its node by communicating with the Deployment Manager.
UDDI Registry	It implements Version 2.0 of the UDDI specification. This enables the enterprise to run its own web services broker within the company or provide brokering services to the outside world. The UDDI Registry is installed as a J2EE 1.3 compliant web application by the administrator into each application server that requires its services.
Web Services Gateway	Is a middleware component that bridges the gap between Internet and intranet environments during Web service invocations. It is used to manage web services, channels that carry requests, filters, and references to UDDI registries in which web services can be registered. The gateway builds upon the WSDL: Web Services Definition Language and the WSIF: Web Services Invocation Framework for deployment and invocation.
Edge Components	Provides a mechanism for efficiently and economically hosting web-accessible content and provide Internet access. The software usually runs on machines close to the boundary between an enterprise intranet and the Internet. The Caching Proxy and Load Balancer are Edge Components.
Cluster	A logical collection of application server processes, responsible for providing workload balancing. Application servers that belong to a cluster are "members" of that cluster and must all have identical application components deployed on them. Other than the applications configured to run on them, cluster members do not have to share any other configuration data. The members of a cluster can be located on a single node - vertical cluster, across multiple nodes - horizontal cluster or a combination of the two.

10 Managed Processes

All operating system processes that are components of the WebSphere product are called managed servers or managed processes. This means that the servers all participate in the administrative domain.

JMX support is embedded in all managed processes and these processes are available to receive administration commands and to output administration information about the state of the managed resources within the processes.

WebSphere provides the following managed servers/processes:

Deployment Manager	Network Deployment only
Provides a single point to access all configuration information and control for a cell. The Deployment Manager aggregates and communicates with all of the node agent processes on each node in the system.	
Node agent	Network Deployment only
Aggregates and controls all of the WebSphere managed processes on its node. There is one node agent per node.	
Application server	Base and Network Deployment
Managed server that hosts J2EE applications.	
JMS server	Network Deployment only
Managed server that hosts the embedded messaging service for a node. There is one JMS server per node. In a base configuration, the JMS server functions are provided within the application server process.	