WebSphere Application Server Standard and Advanced Editions

Getting Started

Version 3 Release 5

WebSphere Application Server Standard and Advanced Editions

Getting Started

Version 3 Release 5

Note

Before using this information and the product it supports, be sure to read the general information under “Appendix. Notices” on page 21.

Fourth Edition (July 2000)

This edition applies to the licensed program IBM WebSphere Application Server Standard Edition and Advanced Edition Version 3 Release 5, Program Number 5648-C84.

Order publications by phone or fax. IBM Software Delivery Solutions/IBM Publications Support takes publication orders between 8:30 a.m. and 7:00 p.m. Eastern Standard Time (EST). The phone number is (800) 879-2755. The fax number is (800) 284-4721.

You can also order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for your comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation
Attn: Software Reengineering
Department G7IA, Building 503
P.O. Box 12195
Research Triangle Park, NC 27709-9990

You can fax comments to (919) 254-0206.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

About This Book

This information is designed to help you understand what the IBM WebSphere Application Server Standard Edition and Advanced Edition Version 3 Release 5 comprises as well as its functions and features.

This book familarizes you with the concepts you need to quickly get started. This book also points you to additional information resources for installing, configuring, migrating, administering, and learning to use the IBM WebSphere Application Server and its features. These resources include Installation Guides for each operating system supported, the InfoCenter, and the IBM WebSphere Application Server help system (see "Where to Find More Information").

Who Should Use This Book

Read this book if you will evaluate, install, or administer the IBM WebSphere Application Server, or if you are a program developer, system architect, or other information technology professional who needs to understand the components of the IBM WebSphere Application Server. You need to have some familiarity with distributed computing and Web computing.

How This Book Is Organized

This section outlines the organization of this book and gives you a brief abstract about each of the chapters, appendixes, and other major topics.

- "Chapter 1. Determining how to get started on the IBM WebSphere Application Server" on page 1 helps you determine how to get started with the IBM WebSphere Application Server. This chapter provides you with a charted course, based on your role as a user of the IBM WebSphere Application Server, for using this book and other information resources.
- "Chapter 2. Introducing the IBM WebSphere Application Server" on page 5 introduces you to IBM's e-business strategy and the WebSphere Family of products, including the IBM WebSphere Application Server. This chapter also is your start to understanding the IBM WebSphere Application Server, and the concepts that make it foremost in e-business strategy. It discusses major features and functions of the product and provides references to other key information.
- The final chapter, "Chapter 3. Obtaining service" on page 19, tells you how to obtain services after you have begun your IBM WebSphere Application Server e-business journey.

Where to Find More Information

There are several ways to get the most recent information about the IBM WebSphere Application Server, including our Web site and InfoCenter, and the WebSphere Administrative Console help.

InfoCenter

The InfoCenter provides planning, installation, systems administration, and problem determination information. The InfoCenter installed with the product contains installation and administration topics.

- **<was_root>\web\InfoCenter\index.html**

To obtain all of the documentation, including a search facility, download the ″full″ InfoCenter from the library page of the product Web site:

http://www.ibm.com/software/webservers/appserv/library.html

Information on the Web

The IBM WebSphere Application Server Web site includes:

- Updates to documentation included in the product package.
- Release Notes that describe known defects and workarounds.
- Product support.
- Product news.
- Case studies and education.
- Ordering information.

Visit the site at:

http://www.ibm.com/software/webservers/appserv/

Related information

The IBM WebSphere Application Server includes the IBM HTTP Server powered by Apache Server software. To the Apache Server base, the IBM HTTP Server adds SSL security, enhanced platform-specific installation programs, and (on Windows NT only) the Fast Response Cache Accelerator (FRCA) feature. For details, visit the following Web site:

http://www.ibm.com/software/webservers/

Note: Consult the IBM WebSphere Application Server Web site for corrections and additions to this information before installing the IBM WebSphere Application Server. Also visit the Library page of the Web site to view the latest Release Notes:

http://www.ibm.com/software/webservers/appserv/library.html

The IBM WebSphere Application Server includes portions of DB2 Universal Database for persistent storage of Java components and for logging data for site analysis. You may use these components only in association with your licensed use of the IBM WebSphere Application Server for the storage and management of data used or generated by WebSphere, and not for other data management purposes.

Chapter 1. Determining how to get started on the IBM WebSphere Application Server

The IBM WebSphere Application Server users typically play one of four roles:

- Evaluator or Reviewer
- Planner/Installer
- System Administrator
- Programmer

Identify the role below that most closely matches your responsibilities, along with the corresponding chart to guide you through a sequence of activities tailored to your specific goals.

Evaluator or reviewer

Refer to this chart if you are responsible for evaluating or reviewing the product. This path provides a high-level introduction to the product, including a quick way to get some hands-on experience. Resource information regarding the details you may need to complete your assessment are also provided.

Step	Objective	Information Resources
1. Get a good start.	Familiarize yourself with IBM WebSphere Application Server concepts.	Finish reading this book.
2. Perform a **Quick** install.	Have a working system to evaluate quickly.	Select the Quick install option for your operating system. See InfoCenter topic 2, Installing the product.
3. Configure and run the samples.	See the IBM WebSphere Application Server in action.	Refer to the IBM WebSphere Application Server Samples Gallery.
4. Download the full, updated InfoCenter from the product Web site.	Broaden your knowledge of the IBM WebSphere Application Server to complete your assessment.	• Visit the IBM WebSphere Application Server product library at www.ibm.com/software/webservers /appserv/library.html • Follow the instructions for downloading the InfoCenter to replace the installed copy, or browse the online InfoCenter.
5.Determine if you need a more robust environment to evaluate the IBM WebSphere Application Server. If so, proceed to the next section, “Planner/Installer” on page 2.	Extend the evaluation environment to evaluate other databases and run specific applications.	Refer to the InfoCenter Planning and Installing Guide.

Planner/Installer

Refer to this chart if you are responsible for formulating and implementing an installation plan. This path gets you started quickly with a fully functioning system. Use this system to familiarize yourself with the IBM WebSphere Application Server while you are learning about planning and installing test and production environments.

Step	Objective	Information Resources
1. Get a good start.	Familiarize yourself with IBM WebSphere Application Server concepts.	Finish reading this book.
2. Perform a **Full*** or **Custom** install.	Pursue fastest way to a fully functioning system; familiarize yourself with the system before learning more about planning and installing.	Select the Full install or the Custom install option for your operating system. See InfoCenter topic 2, Installing the product.
3. Configure and run the samples.	See the IBM WebSphere Application Server in action.	Refer to the IBM WebSphere Application Server Samples Gallery.
4. Plan your solution.	Learn more about different environment configurations and topologies available to efffectively plan your solution.	Refer to the InfoCenter Planning and Installing Guide.
5. Install and configure a test environment.	Use the **Full** or **Custom** install process to begin setting up a test environment.	Refer to the InfoCenter Planning and Installing Guide.
6. Verify your test environment.	Make sure the installation is working correctly.	Refer to the InfoCenter Planning and Installing Guide.
7. Work with your system administrator to configure and deploy applications in a test environment.	Get applications ready to run in a test environment.	Refer to the InfoCenter System Administration Guide.
8. Verify application deployment in a test environment.	Ensure the applications are working correctly in the test environment before moving to production.	Refer to the InfoCenter Planning and Installing and System Administration Guides.
9. Move to a production environment.	**Welcome to e-business!**	Refer to the InfoCenter Planning and Installing Guide.
*Note that in the IBM WebSphere Application Server, the **Full** install is available only with the Windows NT operating system.		

System Administrator

Refer to this chart if you are responsible for administering the product after installation. This path starts with a fully functioning system so that you can familiarize yourself with the product while you are learning about the administration tasks. Information resources about working with your Programmers in deploying applications in test and production environments are also provided.

Step	Objective	Information Resources
1. Get a good start.	Familiarize yourself with IBM WebSphere Application Server concepts.	Finish reading this book.
2. Perform a **Full*** or **Custom** install.	Pursue fastest way to a fully functioning system; familiarize yourself with the system before learning more about planning and installing.	Select the Full install or the Custom install option for your operating system. See InfoCenter topic 2, Installing the product.
3. Configure and run the samples.	See the IBM WebSphere Application Server in action.	Refer to the IBM WebSphere Application Server Samples Gallery.
4. Work through the Application Configuration and Deployment Tutorial	Gain essential training on configuring and deploying your applications.	Refer to the InfoCenter for the Configuration and Deployment Tutorial
5. Learn more about IBM WebSphere Application Server system administration.	Determine the setup and configuration required to get your system working.	Refer to the InfoCenter System Administration Guide.
6. Work with Programmers and the planner/installer to configure and deploy applications in the test environment.	Get the applications ready to run in a test environment.	Refer to the InfoCenter System Administration Guide and Planning and Installing Guide.
7. Verify application deployment in the test environment.	Ensure the applications are working correctly in the test environment before moving to production.	Refer to the InfoCenter Planning and Installing Guide.
8. Move to a production environment.	**Welcome to e-business!**	Refer to the InfoCenter Planning and Installing Guide.
9. Manage and maintain a production environment.	Keep things running smoothly.	Refer to the InfoCenter System Administration Guide.
*Note that in the IBM WebSphere Application Server, the **Full** install is available only with the Windows NT operating system.		

Programmer

Refer to this chart if you are responsible for developing applications for the IBM WebSphere Application Server environment.

Step	Objective	Information Resources
1. Get a good start.	Familiarize yourself with IBM WebSphere Application Server concepts.	Finish reading this book.
2. Perform a **Full*** or **Custom** install.	Pursue fastest way to a fully functioning system; familiarize yourself with the system before learning more about planning and installing.	Select the Full install or the Custom install option for your operating system. See InfoCenter topic 2, Installing the product.
3. Configure and run the samples.	See the IBM WebSphere Application Server in action.	Refer to the IBM WebSphere Application Server Samples Gallery.
4. Learn more about the IBM WebSphere Application Server application development.	Familiarize yourself with the product APIs and the programming model.	Refer to the InfoCenter Programming Guide.
5. Develop and test applications.	Make sure your applications are working as designed.	Refer to the InfoCenter Programming Guide.
6. Work with the systems administrator to configure and deploy applications in test and production environments.	Get the applications ready to run in test and production environments.	Refer to the InfoCenter Programming Guide.
*Note that in the IBM WebSphere Application Server, the **Full** install is available only with the Windows NT operating system.		

Chapter 2. Introducing the IBM WebSphere Application Server

This chapter examines the IBM approach to e-business and discusses how the products in the IBM WebSphere Family provide solutions to your e-business challenges. It also provides an overview of the IBM WebSphere Application Server. For more information about the IBM WebSphere suite of products, see, the *WebSphere Application Server Enterprise Edition Introduction to WebSphere Application Server* at

http://www.ibm.com/software/webservers/appserv/library.html

This chapter gives you an introduction to the following concepts:

- IBM and e-business
- The WebSphere family
- Distributed computing and IBM WebSphere Application Server
- Component options within IBM WebSphere Application Server

IBM and e-business

The World Wide Web (the Web) is still relatively new, but its popularity among both individuals and businesses has grown rapidly. Although individuals use the Web for an array of different purposes, businesses use the Web primarily to provide products, services, and information to their customers, suppliers, and employees.

When the first businesses moved onto the Web, it was enough for them to provide a few static Web pages that listed products and services for sale and provided a telephone number or address to order those products and services. Businesses that provided information services (like software companies) were among the first to enter this new frontier, and they often made their products, in the form of information or software, directly available for downloading.

As the Web matured and new technologies were developed, static Web pages were no longer sufficient. In response, businesses built active Web sites where customers can order products directly, customers and suppliers can communicate with the business, and employees can communicate with each other.

While the Web side of many businesses was changing rapidly, non-Web business systems also went through some major changes as application development spread into distributed systems from mainframe systems. The Open Group's Distributed Computing Environment (DCE) and the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) were two major technologies that provided the infrastructure for these types of systems.

Until recently, Web and non-Web business systems remained largely detached from each other. The IBM e-business initiative and the WebSphere Family changed that by enabling businesses to integrate their Web-based systems with their non-Web systems, to produce a single enterprisewide business system. Further, the WebSphere Family is available in three different editions so that customers can approach the challenge of implementing e-business solutions in several different ways.

The WebSphere Family: providing e-business solutions

The IBM WebSphere Family was designed to help users realize the promise of e-business. The IBM WebSphere Family is a set of software products that helps customers develop and manage high-performance Web sites and integrate those Web sites with new or existing non-Web business systems. Its focus is the following business types:

- Businesses that want to use the latest technologies to establish a powerful Web presence or upgrade their current Web presence
- Businesses that want to develop distributed, enterprisewide business systems and applications
- Businesses that want to integrate their Web presence with their non-Web systems and applications

The WebSphere Family consists of the WebSphere Application Server and other WebSphere Family software that is tightly integrated with the WebSphere Application Server and enhances its performance.

WebSphere Application Server: three editions for different customer needs

To enable customers to achieve their e-business goals, WebSphere is available in three editions:

- The WebSphere Application Server Standard Edition (also called the Standard Application Server) combines the portability of server-side business applications with the performance and manageability of Java™ technologies to offer a comprehensive platform for designing Java-based Web applications. It enables powerful interactions with enterprise databases and transaction systems.
- The WebSphere Application Server Advanced Edition (also called the Advanced Application Server) builds on the Standard Application Server. It introduces server capabilities for applications built to the Enterprise JavaBeans™ Specification from Sun Microsystems and provides some support for integrating the Web applications to other non-Web business systems.
- The WebSphere Application Server Enterprise Edition (also called the Enterprise Application Server) builds on the Advanced Application Server and also offers a robust solution to grow e-business applications into enterprise environments. It combines TXSeries™, IBM's world-class transactional application environment (consisting of both Enicna and CICS), with the fully distributed object and business-process integration capabilities of Component Broker. The Enterprise Application Server contains a complete version of the Advanced Application Server.

These three editions are available on two UNIX® platforms (IBM AIX® and Sun® Microsystems Solaris™) and Microsoft® Windows NT®. WebSphere Standard and Advanced Editions are also available on HP-UX™.

On OS/390®, WebSphere Application Server for OS/390 consists of Standard Edition and Enterprise Edition Component Broker. CICS is also available on the OS/390 platform.

WebSphere Application Server Standard and Advanced Editions also available for the AS/400® platform.

Distributed computing and WebSphere Application Server

The IBM WebSphere Application Server provides an environment for open distributed computing. Users and processes on a wide variety of platforms can interact by using the facilities provided by IBM WebSphere. Both the IBM WebSphere Application Server Advanced Edition and the IBM Enterprise Application Server provide a distributed computing environment. This section provides an overview of the basic concepts involved in distributed computing.

Three-tiered client/server computing

A common way of organizing software to run on distributed systems is to separate functionality into two parts—clients and servers. A *client* is a program that uses services provided by other programs called *servers*. The client makes a request for a service, and a server performs that service. Server functionality often involves some sort of resource management, in which a server synchronizes and manages access to the resource, responding to client requests with either data or status information. Client programs typically handle user interactions and often request data or initiate some data modification on behalf of a user.

For example, a client can provide a form on which a user (a person using a Web browser, for example) can enter orders for a product. The client sends this order information to the server, which checks the product database and performs tasks needed for billing and shipping. A single server is typically used by multiple clients. For example, dozens or hundreds of clients can interact with a handful of servers that control database access.

A common design of client/server systems uses three tiers: a client that interacts with the user, an application server that contains the business logic of the application, and a resource manager that stores data. This approach is shown in Figure 1 on page 8. In this model, the client is isolated from having to know anything about the actual resource manager. If you change the database you are using, the server may have to be modified, but the client does not need to be modified. Because there are usually fewer copies of the server than the client, and because the servers are often in locations that are easier to update (for example, on central machines rather than on PCs running on users' desks), the update procedure is also simplified. Furthermore, this approach provides additional security. Only the servers, not the clients, need access to the data controlled by the resource manager.

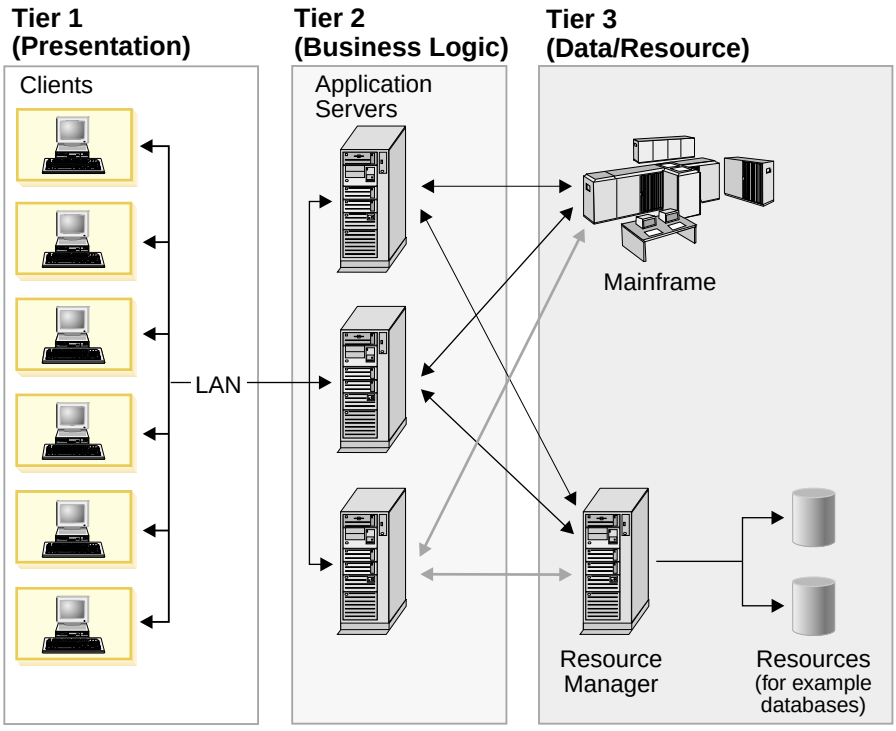


Figure 1. Three-tiered client/server architecture

WebSphere Application Server provides the middle tier in this architecture, allowing clients—applets, Visual Basic® clients, C++ clients, and so on—to interact with data resources (relational databases, MQSeries®, and so on) and existing applications.

The IBM WebSphere Application Server, Standard and Advanced Editions

The IBM WebSphere Application Server Advanced Edition and the IBM WebSphere Application Server Standard Edition provide many powerful enterprise tools, allowing you to build powerful e-business solutions.

What is the difference between the WebSphere Application Server Standard and Advanced Editions?

There are several major differences between the WebSphere Application Server Standard Edition and the WebSphere Application Server Advanced Edition:

- The WebSphere Application Server Advanced Edition licenses and supports the development and use of enterprise beans written to the Enterprise JavaBeans™ (EJB) Specification from Sun Microsystems. The WebSphere Application Server Standard Edition neither supports nor licenses the development of enterprise beans.

- The WebSphere Application Server Advanced Edition supports the replication of application server models that makes it easy to clone application servers across multiple nodes, improving availability. The WebSphere Application Server Standard Edition does not allow replication.
- The WebSphere Application Server Advanced Edition supports a multiple machine environment for servers and servlets. The WebSphere Application Server Standard Edition supports only a single-machine environment for servers and servlets. Both editions support access from multiple client machines.
- The administrative interfaces to the two application servers differ somewhat as a result of the differences in functionality. The interface to the WebSphere Application Server Advanced Edition cannot be used to administer a WebSphere Application Server Standard Edition environment and the interface to the Standard Edition cannot be used to administer a WebSphere Application Server Advanced Edition.

Despite these differences, there is complete compatibility between the two editions, which makes upgrading from the WebSphere Application Server Standard Edition to the Advanced Edition a simple task.

The Enterprise Application Server includes the WebSphere Application Server Advanced Edition. So if you purchase the Enterprise Application Server, you can use any of the products in any of the three WebSphere Application Servers to implement your e-business solutions.

The remainder of this chapter focuses on the WebSphere Application Server Advanced Edition because it contains everything that is in the Standard Edition and more.

The WebSphere Application Server Advanced Edition

The WebSphere Application Server Advanced Edition provides the following major functionality:

- Tools for developing active Web sites through the use of Java servlets and JavaServer Pages (JSP). This functionality is also available in the Standard Edition.
- Tools for developing and deploying enterprise beans written to the EJB Specification. Enterprise beans can act as a bridge between your Web site and your non-Web computer systems.
- A graphical user interface (GUI), the WebSphere Administrative Console, for administering the components of the WebSphere Application Server Advanced Edition environment. This functionality is also available in the Standard Edition.
- A set of application programming interfaces (APIs) for generating, validating, and presenting extensible markup language (XML) documents. This functionality is also available in the Standard Edition.

The WebSphere Application Server Advanced Edition environment

The WebSphere Application Server Advanced Edition contains the following components, which can be combined to create a powerful Java-centered three-tiered system that puts heavy emphasis on a customer's Web site. These components are illustrated in Figure 2 on page 11.

- *Browser-based applications*—Allow users to send and receive information from Web sites by using the Hypertext Transfer Protocol (HTTP). There are three general types

of browser-based applications: Java applets, Java servlets, and JavaServer Pages™ (JSP). For more information, see “Java applets and servlets” on page 11 and “JavaServer Pages” on page 12.

- *Web servers*—Except for stand-alone Java applets, which are restricted by built-in Java security, browser-based applications require that a Web server be installed on at least one machine in your WebSphere Application Server Advanced Edition environment. For more information, see “Web servers” on page 12.
- *Application servers* and *enterprise beans*—The WebSphere application server contains one or more enterprise beans, which encapsulate the business logic and data used and shared by EJB applications. The enterprise beans installed in an application server do not communicate directly with the server. An *EJB container* provides an interface between the enterprise beans and the application server, providing many low-level services such as threading, support for transactions, and management of data storage and retrieval. For more information, see “Application servers and enterprise beans” on page 13.
- *Java applications*—Java applications can interact directly with an application server by using Java remote method invocation over the Internet Inter-ORB Protocol (RMI/IIOP).
- *Data sources*—There are two types of enterprise beans: session beans, which encapsulate short-lived, client-specific tasks and objects, and entity beans, which encapsulate permanent or *persistent* data. The application server stores and retrieves this persistent data in a database.
- *WebSphere Programming Model Extensions*—These tools provide reusable business logic for Java programs. For more information, see “WebSphere Programming Model Extensions” on page 13.
- *Administration server* and the *administrative interface*—The administration server manages servlets, JSP files, enterprise beans, and application servers. This management is directed by the WebSphere Application Server administrator who uses the WebSphere Administrative Console, which is the administrative interface to the administration server. For more information, see “The administration model in the WebSphere Application Server Advanced Edition” on page 14.

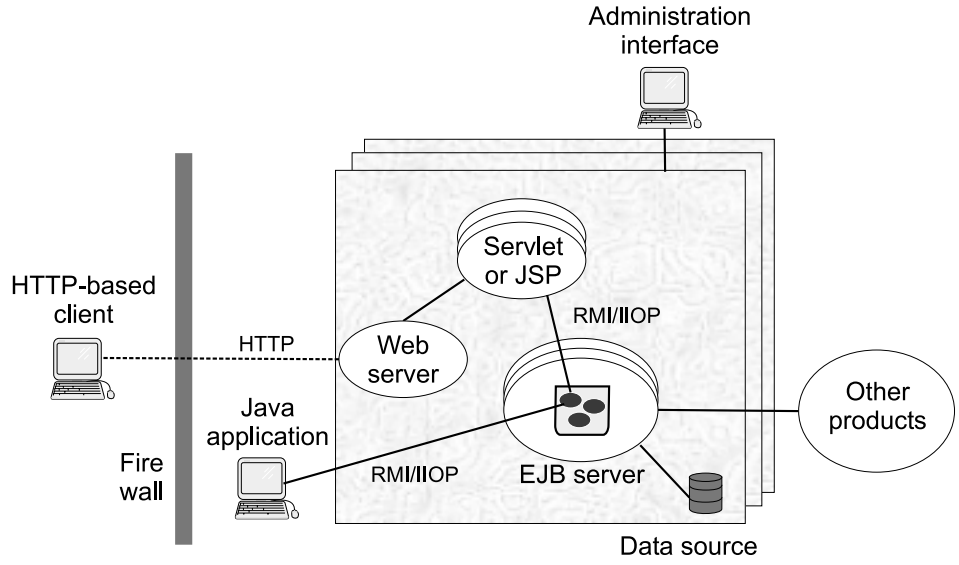


Figure 2. The components of the WebSphere Application Server Advanced Edition environment

Java applets and servlets

Java *applets* are Java applications that run on a browser and extend browser capabilities. Java applets can be designed by using the standard packages found in the Java2 SDK™ or by using the components of the Java Foundation Classes (JFC). For a Java applet to run inside a browser, the browser must support the classes used within the Java applet; however, most browsers can be updated to support the latest SDK by installing browser plug-ins.

Java *servlets* run on a Java-enabled Web server and extend server capabilities. Servlets are Java programs that use the Java Servlet API and associated classes and methods. In addition to the Java Servlet API, servlets can use Java class packages that extend and add to the API. Java applets can be designed to interact with Java servlets, though this is not required.

Servlets extend Web server capabilities by creating a framework for providing request and response services over the Web. When a client sends a request to the server, the server can send the request information to a servlet and have the servlet construct the response that the server sends back to the client.

Unlike the widely-used Common Gateway Interface (CGI) programs, which require an entire process to handle user requests, servlets can handle user requests by using threads. This capability makes servlets much more efficient than CGI programs.

A servlet can be loaded automatically when the Web server is started, or it can be loaded the first time a client requests its services. After being loaded, a servlet continues to run, waiting for additional client requests.

Servlets perform a wide range of functions; for example, a servlet can:

- Create and return an entire HTML Web page containing dynamic content based on the nature of the client request.
- Create a portion of an HTML Web page (an HTML fragment) that can be embedded in an existing HTML page.
- Communicate with other server resources, including databases and Java-based applications.
- Handle connections with multiple clients, accepting input from and broadcasting results to the multiple clients. For example, a servlet can be a multiplayer game server.
- Open a new connection from the server to an applet on the browser and keep the connection open, allowing many data transfers on the single connection. The applet can also initiate a connection between the client browser and the server, allowing the client and server to easily and efficiently carry on a conversation. The communication can be through a custom protocol or through a standard such as IIOP.
- Filter data by MIME type for special processing, such as image conversion and server-side includes (SSI).
- Provide customized processing to any standard server routines. For example, a servlet can modify how a user is authenticated.

JavaServer Pages

The WebSphere Application Server Advanced Edition supports a powerful, new approach to dynamic Web page content: JavaServer Pages (JSP). The JSP function in the Application Server is based on the Sun Microsystems JavaServer Pages Specification.

JSP files are similar in some ways to server-side includes in static HTML because both embed servlet functionality into the Web page. However, in a server-side include, a call to a servlet is embedded within a special servlet tag; in JSP, Java servlet code (or other Java code) is embedded directly into the HTML page.

One of the many advantages of JSP is that it enables you to effectively separate the HTML coding from the business logic in your Web pages. You can use JSP to access reusable components, such as servlets, Java beans, enterprise beans, and Java-based Web applications.

Web servers

The Web server provides the communications link between browser-based applications and the other components of WebSphere Application Server Advanced Edition. The WebSphere Application Server Advanced Edition contains a Java-based servlet engine that is independent of both your Web server and its underlying operating system.

WebSphere Application Server Advanced Edition supports many of the most widely used Web servers. The IBM HTTP Server, which is a modified version of the Apache

server, comes with the WebSphere Application Server Advanced Edition. For information on the supported Web servers, refer to the IBM WebSphere Application Server site at **http://www.ibm.com/software/webservers/**.

Application servers and enterprise beans

An enterprise bean is a Java component that can be combined with other enterprise beans and other Java components to create a distributed, three-tiered application. An application server provides the run-time environment for enterprise beans, handling low-level programming tasks like transaction management, naming, and security. For more information on these services, see “Services used by the WebSphere Application Server Advanced Edition” on page 16.

There are two types of enterprise beans:

- An *entity* bean encapsulates permanent data, which is stored in a data source like a database or a file system, and associated methods to manipulate that data. In most cases, an entity bean must be accessed in some transactional manner. Instances of an entity bean are unique, and they can be accessed by multiple users.

  For example, the information about a bank account can be encapsulated in an entity bean instance. An account enterprise bean might contain an account ID, an account type (checking or savings), and a balance.
- A *session* bean encapsulates one or more business tasks and nonpermanent data associated with a particular client. Unlike the data in an entity bean, the data in a session bean is not stored in a permanent data source and no harm is caused if this data is lost. Nevertheless, a session bean can update data in an underlying database, usually by accessing an entity bean. For this reason, a session bean can be transaction aware. When created, instances of a session bean are identical, though some session beans can store semipermanent data that makes them unique at certain points in their life cycle. A session bean is always associated with a single client.

  For example, the task associated with transferring funds between two bank accounts can be encapsulated in a session bean. Such a transfer enterprise bean might find two instances of an account enterprise bean (by using the account IDs), and then subtract a specified amount from one account and add the same amount to the other account.

Before an enterprise bean can be installed in an application server, the enterprise beans must be deployed. During deployment, several application server-specific classes are generated. The *deployment descriptor* contains attribute and environment settings that define how the application server invokes enterprise bean functionality. Every enterprise bean (both session and entity) must have a deployment descriptor that contains settings used by the application server; these attributes can often be set for the entire enterprise bean or for the individual methods in the bean.

The WebSphere Application Server Advanced Edition provides tools for creating deployment descriptors and deploying enterprise beans.

WebSphere Programming Model Extensions

The Programming Model Extensions are general-purpose utilities, designed to provide common functions in a reusable way. There are two sets of tools provided in the

WebSphere Application Server Advanced Edition environment for Java programmers, the command package and the distributed-exception package.

The command package provides a way for distributed applications to bundle remote requests together, reducing the number of individual remote invocations. Remote invocations are expensive, so the command package can help you improve the performance of distributed applications. In addition, the command package provides a generic way of making requests. The package provides a common way to issue a command, locally or remotely, and independently of server implementation. Any server (an enterprise bean, a JDBC server, and so on) can be the target of a command.

The distributed-exception package helps you manage exceptions in distributed applications. When writing complex distributed applications, you face a choice in handling exceptions. One option is to manage each exception explicitly, catching and rethrowing each by name. This ensures that the information about the original exception is not lost, but can lead to unmanageable code as the number of exceptions increases. The other option is to adopt a strategy of throwing one exception when you catch any of a group. This choice allows you to keep the number of exceptions manageable, but you lose information as exceptions pass through an application. The distributed-exception package allows you to chain a sequence of exceptions into a throwable object. With an exception chain, you can throw one exception in response to another, without losing the previous exceptions. You can also retrieve exceptions from the chain.

The administration model in the WebSphere Application Server Advanced Edition

WebSphere Application Server provides central administration of application servers and other resources. In WebSphere Application Server, an *administrative domain* is a collection of host machines called managed nodes. Each managed node runs an *administration server* (administration servers are also application servers). The node administration server is responsible for configuring, monitoring, and managing resources on that node. Resources include ″live″ objects such as application servers, containers, deployed beans, JSP files, Java servlets, and applications. Resources also include objects such as method groups or policies that are used to define security for resources in the domain.

Resource beans are container-managed persistent (CMP) entity beans. The persistent data associated with a resource (for example, the name, current state, and executable of an application server) is stored in a central data repository. The administration server communicates with a repository server to access, define, and modify resource information stored in the repository. An administration server also communicates with other (remote) administration servers to delegate tasks and to respond to requests. The IBM DB2 relational database, which is packaged with the WebSphere Application Server Advanced Edition, acts as the repository server. You can also use Oracle, Sybase, or InstantDB.

Administration takes place through method calls to resource beans in the repository server. The WebSphere Administrative Console makes requests to an administration server to access or modify a resource in the domain. In the administration server,

session beans invoke methods on the resource beans. Each resource bean has an associated attribute class that contains methods for getting and setting attribute values.

All administration servers in a domain share the central storage for resources in that domain. Regardless of the node it is running on, any administration server can view and modify the characteristics or status of resources on other nodes. If an administration server calls a method on a resource that is running on a different remote node, the method is delegated from the local administration server to the remote administration server.

Resources are modeled in an object type hierarchy that relates the object types to each other. Other object types represent entities such as server groups. Related objects inherit methods from objects above them in the tree hierarchy.

Certain objects in the administrative domain, such as application servers, can be copied (*modeled*) to create replicas (*clones*) that perform identical functions to the object from which they are replicated. This enables the administrator to duplicate server functionality across multiple nodes, improving availability and efficiency. After you clone a resource, modifying the model automatically propagates the same changes to all of the clones. You can efficiently administer several copies of a server or other resource by administering its model.

Resources that can be cloned include the following:

- Application servers
- EJB containers
- Enterprise beans
- Servlets
- Servlet engines
- Web applications

Administration tools

The WebSphere Administrative Console is the administrative interface to the WebSphere Application Server Advanced Edition. It can be used for a range of administrative tasks—from configuring resources and setting security policies, to starting servers and deploying beans, to identifying and responding to system failures and monitoring usage patterns. The tasks supported by the WebSphere Administrative Console fall into six categories: configuration, operation, security, troubleshooting, performance, and data storage.

The WebSphere Administrative Console provides a centralized hierarchical view of resources in an administrative domain, guides for performing administrative operations, forms for viewing and modifying resource attributes, a central browsing facility for JAR files, a messages window for monitoring critical events, and context-sensitive help. The WebSphere Administrative Console modifies information in the repository in response to user commands and reflects any changes to the configuration and status of the administrative domain.

The extensible markup language (XML)

XML is a framework for defining document markup languages. In simple terms, a document markup language is a set of elements (frequently called tags) that have one or more of the following functions:

- Describing the structure of the document.
- Describing the content of the document.
- Controlling how the document is presented to the user.

While HTML is the most widely used markup language for Web-based documents, as the popularity of HTML increased, the limitations of the language became more apparent. Those limitations include restricting the user to a relatively small set of tags. HTML authors cannot create their own HTML tags, because commercially available Web browsers have no knowledge of tags that are not part of the HTML standards that the browsers support.

HTML is further limited because the tags that control presentation are in the same file with tags that describe the document content. Although HTML 4 and Cascading Style Sheets enable HTML authors to separate content from presentation, HTML 4.0 is limited in its ability to describe the content of a document.

XML and Hypertext Markup Language (HTML) are derived from the more complex Standard Generalized Markup Language (SGML). SGML's complexity and high cost of implementation spurred the interest in developing alternatives.

The *XML Document Structure Services* contained in the WebSphere Application Server Advanced Edition enables users to develop servlets and applications that implement server-side XML document processing. It includes a set of APIs for setting servlet configuration parameters without using the administration interface. This alternative method involves creating an XML servlet configuration file (which is an XML document named servlet_instance_name.servlet) that contains the following:

- The name of the servlet class file.
- A description of the servlet.
- The servlet initialization parameters.
- A page list that contains the universal resource identifiers (URIs) of each JSP file that the servlet can call. The page list can include a default page, an error page, and one or more target pages that are loaded if their name appears in the HTTP request.

Services used by the WebSphere Application Server Advanced Edition

Although the WebSphere Application Server Advanced Edition is primarily concerned with the Web side of your business, the application server can act as a bridge to connect your Web and the non-Web applications to span all of your business systems. This section looks at some of the generic tasks that must be accomplished to enable the development and use of distributed applications. It also describes the tools used to approach each of these tasks in the WebSphere Application Server Advanced Edition.

Naming service

In an object-oriented distributed computing environment, clients must have a mechanism to locate and identify the objects as if the clients and objects were all on the same machine. A naming service provides this mechanism. In the application server environment, the Java Naming and Directory Interface (JNDI) is used to provide a common front-end to the naming service.

JNDI provides naming and directory functionality to Java applications, but the API is independent of any specific implementation of a naming and directory service. This independence ensures that different naming and directory services can be used by accessing it behind the JNDI API. Therefore, Java applications can use many existing naming and directory services, for example, the Lightweight Directory Access Protocol (LDAP) or the Domain Name System (DNS).

Transaction service

A *transaction* is a set of operations that transforms data from one consistent state to another. The application server manages transactions for EJB applications by using the mechanism defined in the Java Transaction API (JTA).

For most purposes, enterprise bean developers can delegate the tasks involved in managing a transaction to the application server. The developer performs this delegation by setting the deployment descriptor attributes for transactions. The enterprise bean code does not need to contain transactional logic.

Security service

In the WebSphere Application Server Advanced Edition environment, the main component of the security service is an application server that contains security enterprise beans. When system administrators administer the security service, they manipulate the security beans.

After an EJB client is authenticated, it can attempt to invoke methods on the enterprise beans that it manipulates. A method is successfully invoked if the principal associated with the method invocation has the required permissions to invoke the method. These permissions can be set by application (an administrator-defined set of Web and object resources) and by method group (an administrator-defined set of Java interface-method pairs). An application can contain multiple method groups.

In general, the principal under which a method is invoked is associated with that invocation across multiple Web servers and application servers (this association is known as *delegation*). Delegating the method invocations in this way ensures that the user of an EJB client needs to authenticate only once. HTTP cookies are used to propagate a user's authentication information across multiple Web servers. These cookies have a lifetime equal to the life of the browser session.

Workload management service

WebSphere Application Server supports modeling and cloning of application servers using WebSphere Administrative Console. WebSphere Application Server supports cloning for servlet engines, Web applications, and servlets for workload management, load balancing, and failover.

The workload management service improves the scalability of the application server environment by grouping multiple application servers into application server groups. Clients then access these application server groups as if they were a single server, and the workload management service ensures that the workload is evenly distributed across the application servers in the application server groups. An application server can belong to only one application server group. The creation of application server groups is an administrative task that is handled from within the WebSphere Administrative Console.

Chapter 3. Obtaining service

The WebSphere Application Server Web site contains frequently asked questions with answers:

http://www.ibm.com/software/webservers/appserv/library.html

If you experience a problem with WebSphere Application Server, call:

- Your IBM systems integration consultant, if your implementation is being assisted by IBM Global Services
- The IBM Software Service Support: 1-800-237-5511

To learn more about IBM Software Support, see the IBM support page at:

http://www.ibm.com/Support

You can also e-mail us directly with your suggestions and requirements for future releases, and to report noncritical defects that do not require a personal interaction or formal support: **WASTEAM@US.IBM.COM**

Appendix. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Director of Licensing
IBM Corporation
North Castle Drive
Amonk, NY 10504-1785
USA

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

This product includes computer software created and made available by CERN. This acknowledgement shall be mentioned in full in any product which includes the CERN computer software included herein or parts thereof.

Trademarks

The following terms are trademarks of IBM Corporation in the United States or other countries or both.

AIX

IBM

OS/390

RS/6000

TXSeries

WebSphere

WebSphere Application Server

Microsoft, Windows, Windows NT and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

HP-UX is a trademark or registered trademark of Hewlett-Packard Company.

Sun, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

Readers' Comments — We'd Like to Hear from You

WebSphere Application Server Standard and Advanced Editions
Getting Started
Version 3 Release 5

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

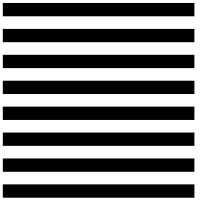
Phone No. _____

Readers' Comments — We'd Like to Hear from You

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Attn: Software Reengineering
Department G7IA, Building 503
Research Triangle Park, NC 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Part Number: CT7C8IE
Program Number: 5648-C84

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

(1P) P/N: CT7C8IE