IBM WebSphere Application Server, Version 5.0.2

Getting started

??

Before using this information, be sure to read the general information under "Trademarks and service marks" on page v.

Compilation date: July 14, 2003

Contents

Trademarks and service marks

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

- Cloudscape
- Everyplace
- iSeries
- IBM
- Redbooks
- ViaVoice
- WebSphere
- zSeries

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product and service names may be trademarks or service marks of others.

Chapter 1. Overview of WebSphere Application Server components

Java 2 platform, Enterprise Edition (J2EE)

The J2EE platform is the standard for developing, deploying, and running enterprise applications. Read this document for a brief overview of key J2EE concepts, including the parts of the J2EE runtime environment and the J2EE application packaging and deployment. The ultimate source of J2EE information is the specification itself, available from the Sun Microsystems Web site (java.sun.com). For a list of the specification levels that comprise the J2EE 1.3 specification (Servlet, EJB, and so on), see http://java.sun.com/j2ee/1.3/docs/#specs.

IBM WebSphere Application Server, Version 5 has completed the full J2EE certification test suite. The product supports all of the J2EE 1.3 APIs, and exceeds many with its extensions. You can check the list of J2EE-compatible configurations posted by Sun Microsystems at http://java.sun.com/j2ee/1.2_compatibility.html.

Java 2 Platform, Enterprise Edition defines a standard that applies to all aspects of designing, developing, and deploying multi-tier, server-based applications. The standard architecture defined by the J2EE specification is composed of the following elements:

- Standard application model for developing multi-tier applications.
- Standard platform for hosting applications.
- Compatibility test suite for verifying that J2EE platform products comply with the J2EE platform standard.
- Reference Implementation providing an operational definition of the J2EE platform.

The J2EE platform specification describes the runtime environment for a J2EE application. This environment includes application components, containers, and resource manager drivers. The elements of this environment communicate with a set of standard services that are also specified. For more information, see Three-tier architectures.

J2EE platform roles

The J2EE platform also defines a number of distinct roles performed during the application development and deployment life cycle:

- The product provider designs and offers the J2EE platform, APIs, and other features defined in the J2EE specification for purchase.
- The tool provider provides tools used for the development and packaging of application components.
- The application component provider creates Web components, enterprise beans, applets, or application clients to use in J2EE applications.
- The application assembler takes a set of components developed by component providers and assembles them in the form of an enterprise archive (EAR) file.
- The deployer is responsible for deploying an enterprise application into a specific operational environment.
- The system administrator is responsible for the operational environment in which the application runs.

Product providers and tool providers have a product focus. Application component providers and application assemblers focus on the application. Deployers and system administrators focus on the runtime.

These roles help to identify the tasks that need to be performed and the parties involved. Understanding this separation of roles is important, because it helps to understand the approach that should be taken when developing and deploying J2EE applications.

For information about the relationship of IBM WebSphere Application Server roles to J2EE roles, see User roles and activities.

J2EE specification benefits

The J2EE specification provides customers a standard by which to compare J2EE offerings from vendors and develop applications that will execute on any J2EE compliant platform. Comprehensive, independent Compatibility Test Suites ensure vendor compliance with J2EE standards.

Some benefits of deploying to a J2EE-compliant architecture are:
- A simplified architecture based on standard components, services and clients, that takes advantage of the write-once, run-anywhere Java technology.
- Services providing integration with existing systems, including Java DataBase Connectivity (JDBC); Java Message Service (JMS); Java Interface Definition Language (Java IDL); the JavaMail API; and Java Transaction API (JTA and JTS) for reliable business transactions.
- Scalability to meet demand, by distributing containers across multiple system and using database connection pooling, for example.
- A better choice of application development tools, and components from vendors providing off-the-shelf solutions.
- A flexible security model that provides single sign-on support, integration with legacy security schemes, and a unified approach to securing application components.

The J2EE specifications are the result of an industry-wide effort that has involved, and still involves, a large number of contributors. IBM alone has contributed to defining more than 80 percent of the J2EE APIs.

Application components and their containers

The J2EE programming model has four types of application components, which reside in four types of containers in the Application Server:
- Enterprise JavaBeans — Executed by the EJB container
- Servlets and JavaServer Pages files — Executed by the Web container
- Application clients — Executed by the application client container
- Applets — Executed by the applet container

For a thorough description of the components and containers that apply specifically to the IBM WebSphere Application Server, see Architectural features.

J2EE containers provide the run-time support of the application components. There must be one container for each application component type in a J2EE application. By having a container between the application components and the set of services, the J2EE specification can provide a federated view of the APIs for the application components.

A container provides APIs to application components for accessing services. A container can also handle security, resource pooling, state management, as well as naming and transaction issues.

Standard services

The J2EE platform provides components with a set of standard services that they can use to interact with each other. See the Sun Web page, http://java.sun.com/products/ for descriptions of each standard service.

- HTTP and HTTPS
- Java Transaction API (JTA)
- Remote Method Invocation/Internet Inter-ORB Protocol (RMI/IIOP)
- Java Interface Definition Language (Java IDL)
- Java DataBase Connectivity (JDBC)
- Java Message Service (JMS)
- Java Naming and Directory Interface (JNDI)
- JavaMail and JavaBeans Activation Framework (JAF)
- Java Transaction API (JTA and JTS)
- XML
- J2EE Connector Architecture
- Resource managers

J2EE packaging

Perhaps the most significant change introduced by the J2EE specification is how application components are packaged for deployment.

During a process called assembly, J2EE components are packaged into modules. Modules are then packaged into applications. Applications can be deployed on the Application Server. Each module and application contains a J2EE deployment descriptor. The deployment descriptor is an XML file providing instructions for deploying the application.

For more information, including IBM WebSphere Application Server specifics, see the *Assembling or packaging* topic (welc_assembling) in the IBM WebSphere Application Server, Version 5 InfoCenter, which is available at http://www-3.ibm.com/software/webservers/appserv/infocenter.html.

Three-tier architectures

WebSphere Application Server provides the application logic layer in a three-tier architecture, enabling client components to interact with data resources and legacy applications.

Collectively, *three-tier architectures* are programming models that enable the distribution of application functionality across three independent systems, typically:

- Client components running on local workstations (tier one)
- Processes running on remote servers (tier two)
- A discrete collection of databases, resource managers, and mainframe applications (tier three)

These tiers are logical tiers. They might not run on the same physical server.

Tier 1 (Presentation) — Clients
Tier 2 (Business Logic) — Application Servers
Tier 3 (Data/Resource) — Mainframe, Resource Manager, Resources (for example databases)
LAN
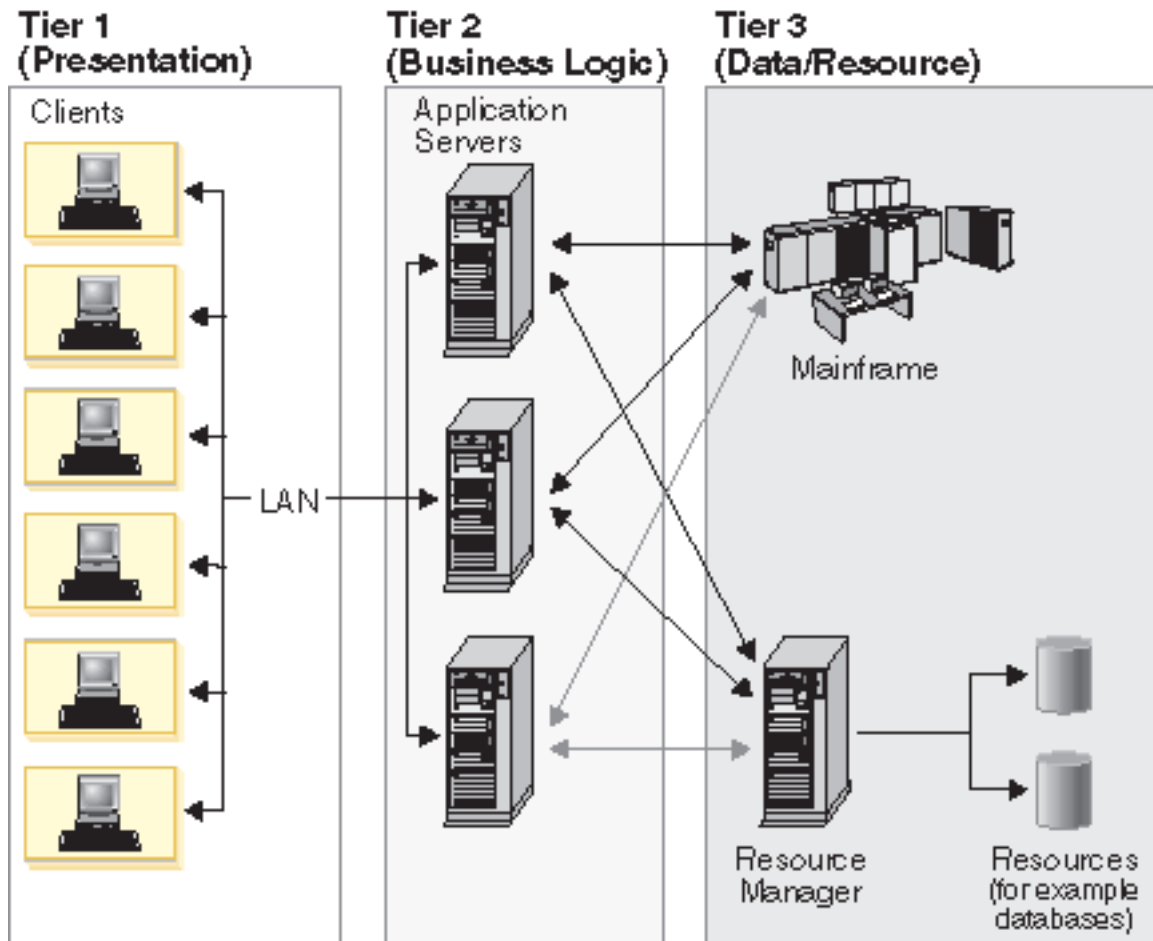
First tier. Responsibility for presentation and user interaction resides with the first-tier components. These client components enable the user to interact with the second-tier processes in a secure and intuitive manner. WebSphere Application Server supports several client types.

Clients do not access the third-tier services directly. For example, a client component provides a form on which a customer orders products. The client component submits this order to the second-tier processes, which check the product databases and perform tasks needed for billing and shipping.

Second tier (application logic layer). The second-tier processes are commonly referred to as the application logic layer. These processes manage the business logic of the application, and are permitted access to the third-tier services. The application logic layer is where the bulk of the processing work occurs. Multiple client components are able to access the second-tier processes simultaneously, so this application logic layer must manage its own transactions.

Continuing with the previous example, if several customers attempt to place an order for the same item, of which only one remains, the application logic layer must determine who has the right to that item, update the database to reflect the purchase, and inform the other customers that the item is no longer available. Without an application logic layer, client components access the product database directly. The database is required to manage its own connections, typically locking out a record that is being accessed. A lock can occur simply when an item is placed into a shopping cart, preventing other customers from even considering it for purchase. Separating the second and third tiers reduces the load on the third-tier services, can improve overall network performance, and allows more eloquent connection management.

Third tier. The third-tier services are protected from direct access by the client components by residing within a secure network. Interaction must occur through the second-tier processes.

Communication among tiers. All three tiers must be able to communicate with each other. Open, standard protocols and exposed APIs simplify this communication. Client components then can be written in any programming language, such as Java or C++, and can run on any operating system, as long as they can speak with the application logic layer. Likewise, the databases in the third tier can be of any design, as long as the application layer can query and manipulate them. The key to this architecture is the application logic layer.

Architectural features

This section examines the major components within IBM WebSphere Application Server.

HTTP server

IBM WebSphere Application Server works with an HTTP server to handle requests for servlets and other dynamic content from Web applications. (The terms HTTP server and Web server are used interchangeably throughout the documentation.)

The HTTP server and Application Server communicate using the WebSphere HTTP plug-in for the HTTP server. The HTTP plug-in uses an easy-to-read 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. It can also be configured to use secure HTTPS, if required. The HTTP plug-in is available for popular Web servers.

For more information, see *Configuring Web server plug-ins* (trun_plugin) in the IBM WebSphere Application Server, Version 5 InfoCenter, which is available at http://www-3.ibm.com/software/webservers/appserv/infocenter.html.

Application server

The Application Server collaborates with the Web server by exchanging client requests and application responses. You can define multiple Application Servers, each running in its own Java Virtual Machine (JVM).

- **EJB container**

  The EJB container provides the runtime services needed to deploy and manage EJB components, from here on known as enterprise beans. It is a server process that handles requests for both session and entity beans.

  The enterprise beans (inside EJB modules) installed in an Application Server do not communicate directly with the server; instead, an EJB container provides an interface between the enterprise beans and the server. Together, the container and the server provide the bean run-time 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 manage more than one EJB JAR file.

  For more information, see the *EJB containers* topic (cejb_ecnt) in the IBM WebSphere Application Server, Version 5 InfoCenter, which is available at http://www-3.ibm.com/software/webservers/appserv/infocenter.html.

- **Web container**

  Servlets and JavaServer Pages (JSP) files are server-side components used to process requests from HTTP clients, such as Web browsers. They handle presentation and control of the user interaction with the underlying application data and business logic. They can also generate formatted data, such as XML, for use by other application components.

The Web container processes servlets, JSP files and other types of server-side includes. Pre-J2EE servlets would run in a servlet engine. 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 destroy() method when appropriate and unloads the servlet, after which the JVM performs garbage collection.

A Web container handles requests for servlets, JavaServer Pages (JSP) files, and other types of files that include server-side code. The Web container creates servlet instances, loads and unloads servlets, creates and manages request and response objects, and performs other servlet management tasks. WebSphere Application Serve provides Web server plug-ins for supported Web servers. These plug-ins pass servlet requests to Web containers.

- **Application client container**

  Application clients are Java programs that typically run on a desktop computer with a graphical user interface (GUI). They have access to the full range of J2EE server-side components and services.

  The application client container handles Java application programs that accesses enterprise beans, Java Database Connectivity (JDBC), and Java Message Service message queues. The J2EE application client program runs on client machines. This program follows the same Java programming model as other Java programs; however, the J2EE application client depends on the application client run time to configure its execution environment, and uses the Java Naming and Directory Interface (JNDI) name space to access resources.

  For more information, see *Application clients* topic (ccli_appclients in the InfoCenter).

- **Applet container**

  An applet is a client Java class that typically executes in a Web browser, but can also run in a variety of other client applications or devices.

  Applets are often used in combination with HTML pages to enhance the user experience provided by a Web browser. They can also be used to shift some of the processing workload from the server to the client.

  The applet container handles Java applets embedded in a HyperText Markup Language (HTML) documents that reside on a client machine that is remote from the Application Server. With this type of client, the user accesses an enterprise bean in the Application Server through the Java applet in the HTML document.

  For more information, see *Application clients* (ccli_appclients in the InfoCenter).

- **Embedded HTTP server**

  A nice product feature is the HTTP handling capability embedded within the Application Server, enabling an HTTP client to connect directly to the Application Server. Or, as described earlier, an HTTP client can connect to a Web server and the HTTP plug-in can forward the request to the Application Server.

  For more information, see the *Web container* (cweb_aov4 in the InfoCenter).

- **Virtual host**

  A virtual host is a configuration enabling a single host machine to resemble multiple host machines. Resources associated with one virtual host cannot share data with resources associated with another virtual host, even if the virtual hosts share the same physical machine.

  Virtual hosts allow the administrator to associate Web applications with a particular host configured for the machine executing the application.

  For more information, see the *Virtual hosts* (crun_vhost in the InfoCenter).

Product family overview

A variety of product lines complementing IBM WebSphere Application Server are described below. To investigate product availability and pricing, see the www.ibm.com Web pages for each product.
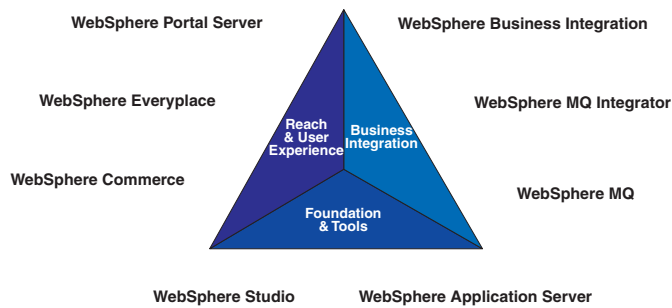
Foundation and tools

The following products collaborate to help you achieve scalability and productivity by growing and building e-business applications rapidly and reliably.

- **WebSphere Application Server**

  Packaging changes in Version 5 make it easier than ever to upgrade as needed, as your business grows. See also *WebSphere Application Server packages*.

  The product is available for several distributed operating systems, as well as specifically optimized for zSeries and iSeries. For more information, see http://www.ibm.com/webservers/appserv/.

- **WebSphere MQ**

  This is software for exchanging information among more than 35 platforms with assured delivery. For more information, see http://www.ibm.com/software/ts/mqseries/ .

- **WebSphere Studio**

  These are e-business professional tools based on a common workbench technology. Based on open standards, they exploit the WebSphere runtime environment. For more information, see http://www.ibm.com/software/webservers/studio/.

Reach and user experience

The following products collaborate to help you achieve customer loyalty by extending and personalizing user experiences.

- **WebSphere Portal**

  This is software for accessing widespread and diverse data sources from anywhere, anytime, by anyone you allow. For more information, see http://www.ibm.com/software/webservers/portal/.

- **WebSphere Everyplace**

  This is software infrastructure to support mobile solutions, addressing the challenge of extending e-business applications to mobile devices. For more information, see http://www.ibm.com/software/pervasive/.

- **WebSphere Commerce**

  These are powerful solutions designed to handle the broad range of challenges encountered when selling in Business-to-Business (B2B) and Business-to-Consumer (B2C) environments. For more information, see http://www.ibm.com/software/webservers/commerce/.

Business integration

The following products collaborate to help you achieve business agility by integrating applications and automating business processes.

- **WebSphere Business Integration**

This software creates the nimble infrastructure needed to support the business imperatives of your dynamic enterprise. For more information, see http://www.ibm.com/software/integration/ .

- **WebSphere MQ Integrator**

  This software helps you to flexibly connect and integrate assets within your enterprise and with trading partners. For more information, see http://www.ibm.com/software/integration/.

Product GUIs and tools

IBM WebSphere Application Server provides a variety of graphical and non-graphical user interfaces. The product GUIs are described, as well as some key scripting and command line tools. For a complete list of available commands, as well as more information about the commands referenced below, see the **Command line syntax** section of the **Quick reference** view in the InfoCenter .

Tools for installing, upgrading, and migrating

First Steps
> First Steps is a desktop GUI from which you can start or stop the Application Server. It also provides access to the administrative console and the Application Assembly Tool. See *First Steps tool tips*.

Launch Pad
> The Launch Pad is a graphical interface for launching the product installation. It also provides links to information you might need for installation. See *Using the LaunchPad to start the installation*.

Installation wizard
> This graphical interface leads you through the process of installing the product.

Migration tools
> Command line tools such as WASPreUpgrade and WASPostUpgrade are available to help you migrate from a previous product version.

Tools for developing applications

IBM WebSphere Application Server is not an application development tool, although it provides some Application Programming Interfaces (APIs) for improving the applications you deploy on the server. The WebSphere Studio Application Developer product line offers development environments and tools that complement each edition of the Application Server. For more information, see *Product family overview*.

Tools for assembling applications

As described in the *Assembling or packaging* topic (welc_assembling in the InfoCenter), assembly is a necessary packaging and configuration step prior to deploying an application onto the server.

Application Assembly Tool
> AAT is used to assemble enterprise applications for deployment. For more information, see the *Assembling applications* topic (uaatt_skel in the InfoCenter).

Deployment tool
> This is a command-line tool that the graphical Application Assembly Tool calls behind the scenes, to generate code for deployment. In specific cases, you might need to use it. See *The ejbdeploy tool* (raat_ejbdeploycmd in the InfoCenter).

Application Client Resource Configuration Tool
> This tool helps you configure deployment descriptors that define the resources needed by application clients. For more information, see *Deploying application clients* (ucli_tconfigclient in the InfoCenter).

clientUpgrade
> Use this tool to migrate client JAR files from J2EE 1.2 to J2EE 1.3.

Text editor
While it is recommended that you use the graphical AAT to edit deployment descriptors, these XML documents can be opened in your favorite text editor.

Tools for deploying and administering

As described in the *Deploying* topic (welc_deploying in the InfoCenter), deploying involves putting an application onto a particular server.

Systems administration tools
See the *Welcome to System Administration* topic (welc_configop in the InfoCenter), for a description of the available systems administration tools, including the graphical WebSphere Application Server administrative console, the wsadmin scripting client, and an assortment of special purpose command line tools.

A notable part of the administrative console is the **Security Center**.

LaunchClient command
This command line tool starts application clients. See the *launchClient tool* topic (rcli_javacmd in the InfoCenter).

XML-SOAP administrative tool
This graphical interface helps you manage deployed Web services. See the *Administering deployed Web services (XML-SOAP administrative tool)* topic (twbs_adminwbs in the InfoCenter).

UDDI user console
This is the graphical interface for interacting with the IBM WebSphere UDDI Registry. See the *UDDI user console* topic (twsu_uc in the InfoCenter).

NameSpaceDump tool
This command line interface displays the IBM WebSphere Application Server name space for debugging purposes. See *dumpNameSpace tool* topic (rnam_dump_utility in the InfoCenter).

Tools for monitoring and tuning

PMI Request Metrics in WebSphere Administrative Console
The product collects data by timing requests as they travel through components of the product. PMI Request Metrics, which is configurable through the administrative console, logs the time spent in major components, such as the Web container of the Application Server. These data points are recorded in logs and can be written to Application Response Time (ARM) agents used by Tivoli monitoring tools.

Tivoli Performance Viewer
This is a stand-alone program that monitors and helps analyze Application Server data. It is built on the Performance Monitoring Infrastructure (PMI) Client API, which also is exposed to third-party development tools. For more information, see the *Monitoring performance with Tivoli Performance Viewer (formerly Resource Analyzer)* topic (tprf_tpvmonitor in the InfoCenter) and the *Performance Monitoring Infrastructure* topic (cprf_pmidata).

Tools for troubleshooting

Application Server Toolkit
Included with IBM WebSphere Application Server, but on a separately installable CD, this kit includes debugging functionality that is built on the Eclipse workbench. See the *Debugging with the Application Server Toolkit* topic (ctrb_debug in the InfoCenter).

Collector tool
The Collector Tool gathers information about your WebSphere Application Server installation and packages it in a jar file that can be sent to IBM Customer Support to assist in problem determination and analysis. For more information, see the *Collector Tool* topic (ctrb_ct in the InfoCenter).

Log Analyzer

The Log Analyzer takes one or more service or activity logs, merges all of the data, and displays the entries. Based on its symptom database, it analyzes and interprets the error conditions in the log entries to help you debug problems. Log Analyzer has a special feature enabling it to download the latest symptom database from the IBM Web site. For more information, see the *Log Analyzer* topic (ctrb_jfla in the InfoCenter).

User roles and activities

Review the user role and activity descriptions to understand how someone with your organizational role might use WebSphere Application Server. If you are new to the product or Java 2 Platform, Enterprise Edition (J2EE), the simplified model presented here should help you with the basic flow of activities involved in seeing a J2EE application through to deployment.

Setting up and administering a production environment follows roughly these phases.

Simple timeline of activities:
Planning, Installer, and Administrator role

Planning the production environment

Installing the product, setting up multiple node environments

Migrating existing installations and configurations

Administering in preparation for application deployment

Obtaining **assembled** modules containing application code

Deploying modules onto test, production servers

Testing access to deployed modules

Administering deployed modules, servers, resources

Monitoring and tuning performance

Troubleshooting problems

Updating and re-deploying applications

Time

Using the Application Server in an application development environment follows roughly these phases.

Simple timeline of activities:
Application Developer role

Installing the product in application development environment

Planning application development approaches

Developing and unit-**testing** application components

Migrating existing application components and modules

Assembling code into modules for development

Deploying modules onto development servers

Testing access to deployed modules

Monitoring and tuning performance

Troubleshooting problems

Updating and re-deploying applications

Time

The product tools (described in *Product GUIs and tools*) and documentation are geared towards helping you with these activities. As you learn more, you will see ways to tailor the flow of activities to your specific needs.

Take advantage of task overviews. Task overviews are special sets of steps in this documentation set. Each outlines a feasible sequence of tasks for working with an area of product functionality, such as security. The tasks typically reflect the main activities, such as Migrating, Developing, Assembling, Deploying, and so on. Use task overviews to gain broad knowledge of the decisions and actions needed to accomplish your goals. From task overviews, you can drill down to more detailed sub-tasks.

For a list of available task overviews, see the **Task overviews** section of the **All topics by activity** view of the online IBM WebSphere Application Server, Version 5 InfoCenter (at http://www-3.ibm.com/software/webservers/appserv/infocenter.html).

Mapping to J2EE roles. The following is the mapping of WebSphere Application Server roles and activities to the roles defined by the J2EE 1.3 specification.

J2EE roles	Product roles	Product activities
Non-applicable	Planner, Installer, IT architect	Planning, Installing product environment
Application Component Provider	Developer / Programmer	Developing
Application Assembler	Developer / Programmer	Assembling
Deployer	Administrator (in production environment)	Deploying, Testing application deployment
System Administrator	Administrator	Administering
Non-applicable	All of the above	Migrating, Tuning, Troubleshooting
Tool Provider, J2EE Product Provider	WebSphere Application Server and product family	Non-applicable

Accessibility features

Accessibility features enable anyone with a physical disability, such as restricted mobility or limited vision, to operate software products successfully.

The administrative console is the primary interface for interacting with the product. This console is displayed within a standard Web browser. By using an accessible Web browser, such as Microsoft Internet Explorer or Netscape Navigator, administrators are able to:

- Use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Use voice recognition software, such as IBM ViaVoice, to enter data and to navigate the user interface
- Operate features by using the keyboard instead of the mouse

One can configure and administer product features by using standard text editors and scripted or command line interfaces instead of the graphical interfaces provided.

When appropriate, the documentation for specific product features contains additional information about the accessibility of the features.

Chapter 2. What is new in this release

IBM WebSphere Application Server, Version 5, offers a world-class infrastructure for the next chapter in open e-business platforms. As the foundation of the WebSphere software platform, WebSphere Application Server provides a rich, e-business application deployment environment with a complete set of application services including capabilities for transaction management, security, clustering, performance, availability, connectivity and scalability. Version 5 offer full J2EE specification support (Servlet 2.3, JSP 1.2, EJB 2.0, and others), as well as a variety of extensions.

Several new and improved Version 5 features are summarized here, with links to more information.

Features that are new or improved in fix packs are indicated as such. Technical updates are indicated similarly throughout the documentation. See also the Site Map, available from the banner of the online InfoCenter. From the **Site Map**, you can determine updates relative to the documentation at the V5 level. Version 3.5 and Version 4 users will know this as the documentation **Revision history** or **What's New**.

When you purchase IBM WebSphere Application Server, Version 5 you are entitled to one copy of IBM WebSphere Application Server Network Deployment Version 5.

With your purchase of IBM WebSphere Application Server, Version 5, you are entitled to use three features that ship with IBM WebSphere Application Server Network Deployment Version 5:
* Edge Components: For Load Balancing, Custom Advisors, Consultants, and Enhanced Caching
* Application Client: A stand-alone java program launched from the command line or desktop, and typically accesses EJB programs running on the J2EE Application Server
* Application Server Toolkit: Provides the tooling necessary to debug Web applications, JavaScript, Java, and compiled languages

If you currently own IBM WebSphere Application Server Network Deployment, you already have these three components. You can use the software and adopt the IBM WebSphere Application Server for z/OS license. If you do not have IBM WebSphere Application Server Network Deployment, contact your IBM representative, who will provide you with the necessary software.

Planning, installation, and product migration
* **5.0.2** Windows 2003 installation is now available. For instructions, see Installing WebSphere Application Server.
* **5.0.1** HP-UX installation is now available. For instructions, see Installing WebSphere Application Server.
* **5.0.1** Manage installation of fixes and FixPaks with the update installer application. For instructions, see Installing fixes and FixPaks.
* The product offers production-ready J2EE 1.3 standards and Web services support.
* Migration tools help you migrate existing configurations from certain previous versions during the installation process. The tools perform both pre-upgrade and post-upgrade activities. For more information, see Migrating and coexisting.
* **5.0.2** Use the WebSphere Application Server migration tools to migrate your WebSphere Application Server - Express configuration and applications to IBM WebSphere Application Server as described in the *Migrating from WebSphere Application Server - Express* in the base product InfoCenter.
* Multiple configuration instances are also available. Configuration instances appear to be separate installation instances on a single machine but actually share product binaries and have unique configurations and data only. For more information, see Creating multiple V5 configuration instances.
* The servlet redirector and remote OSE mechanisms are no longer supported. Instead, HTTP forwards Web requests from your Web server to an HTTP server running inside the appropriate Application Server.

- WebSphere Edge Server functionality has been integrated into IBM WebSphere Application Server, including a Demilitarized Zone (DMZ) CD containing software for load-balancing and content caching. The edge of the DMZ includes:
  - HTTP routers
  - HTTP server
  - Caching proxy
  - Deployment manager (further described in the Systems Administration section of this document)

  For more information, see *Configuring Edge Side Include caching* in the InfoCenter.

New and improved WebSphere Samples Gallery

- Technology centered samples, including EJB, J2EE client, JMS, JSP, and Servlet samples.
- The **Plants by WebSphere** ″super sample,″ demonstrating multiple technologies used to build realistic applications.
- Java Petstore sample
- ANT-based build scripts enabling you to run, modify, rebuild, and run the samples again.
- Use of Cloudscape rather than DB2 for the samples requiring a database. Cloudscape has a smaller footprint.
- Samples demonstrating each of the Enterprise product extensions.

For more information, see *Samples Gallery* in the InfoCenter.

Servers

- The product now has a single JVM run time, including containers, naming, security, administration, resources, ORB, and HTTP engine
- The new, flexible packaging structure means there is just one Application Server code base, with add-ons available for scale and function. The Application Server now runs with the high performing:
  - IBM Developer Kit, Java Technology Edition, Version 1.3.1 used on AIX, Windows, and Linux operating systems
  - Java virtual machine from Sun on Solaris operating system
  - Java virtual machine from HP-UX on the HP-UX operating system

  For more information, see the *Using the JVM* topic (trun_jvm in the InfoCenter).
- Workload management and clustering has improved:
  - Clusters now can be managed with browser based, scripting, and JMX administrative clients
  - The ability to configure member weights provides the administrator more control in tuning the system
  - A new load balancing algorithm has been incorporated to better optimize the distribution of client requests across cluster members
  - Enhanced failover support provides quicker failover in the case of network outages and other problems

  In Version 4, clusters were known as *server groups* or model and clones.

  For more information, see the *Balancing workloads with clusters* topic (trun_wlm in the InfoCenter).
- In Version 3.5.x and Version 4, the default bootstrap setting was 900. Now the default is 2809. This does not cause problems in applications unless there is a direct reference to port 900. You can reconfigure the port setting to 900 if you migrate your previous configuration.

  For more information, see Port number settings in WebSphere Application Server versions.
- The system name space structure provided by the name server has changed significantly since the last release, including:

– The Version 5 name space is distributed, meaning that objects are not all bound under a single context root as with previous versions.

– The name space consists of partitions. Some partitions in the name space contain transient bindings and some partitions contain persistent bindings.

These and other new naming features are summarized in the *New features for name space support* (rnam_new_features in the InfoCenter).

Applications and EJB modules

- EJB persistence manager has been re-architected to support EJB 2.0 CMP scheme, which differs greatly from the EJB 1.1 scheme

- EJB persistence manager has improved in modularity, maintainability, and performance. Maintenance focuses on configuration of server components, with less need to regenerate deployed artifacts

- EJB 2.0 specification support, including:
  – Local and remote beans
  – Message-driven beans
  – Container-managed relationships
  – A portable finder query language
  – All other aspects of the specification
  – Programming model
  – Abstract and concrete entity beans
  – Local home and local entity interfaces
  – Container-managed association relationships
  – Dependent values
  – EJB query language

- Plus these features that add high performance persistence beyond EJB 2.0:
  – Changing semantic behavior
  – Entity bean inheritance
  – Optimistic concurrency control
  – Read-ahead
  – Intent mechanism support
  – Support for different types of backend access mechanisms
  – Procedural access
  – SQLJ
  – Data caching

EJB specification extensions are described in the *WebSphere extensions to the Enterprise JavaBeans specification* topic (rejb_spcx in the InfoCenter).

- Powerful new features enhance Container-managed persistence (CMP) entity bean performance, including:
  – Caching of bean data at several levels
  – Long lifetime caching, for beans that change only infrequently and thus remain read-only across many transactions
  – Read-ahead, which pre-loads groups and working-sets of beans in a single datastore operation by following selected bean relationships
  – Optimistic concurrency control, which minimizes the amount of time data is actually locked during updates and thus increases overall throughput in heavily-used applications

- Activity sessions are a new feature of the Enterprise product. An activity session is a ″unit of work″ construct which can group work done by several transactions into a user-definable unit. Various properties and configurations can be associated with an Activity Session.

  For more information, see the *Using the ActivitySession service* topic (tas_ep in the InfoCenter).
- CMP beans and bean-managed persistence (BMP) beans can share datastore connections, allowing access to related data by both kinds of beans when in the same transaction.
- CMP beans can inherit from one another. (In other words, they can subclass one another.) The Application Server will recognize this during bean deployment and at run time will — for example — allow finders to return beans of that class or any subclass. Inheritance can be expressed in relational datastores in either *single-table* or *root-leaf* arrangements.

Applications and Web modules

New enhancements include:
- Servlet 2.3 with Filters and Events
- JSP 1.2 with XML Syntax
- Changes in autoRequestEncoding and autoResponseEncoding

  The web container no longer automatically sets request and response encodings and response content types. The programmer is expected to set these values using the methods available in the Servlet 2.3 API. If you want the Application Server to attempt to automatically set these values, set autoRequestEncoding=true in order to have the request encoding value set and set autoResponseEncoding=true in order to have the response encoding and content type set. These values can be found in the ibm-web-ext.xmi file for each web application.

  For more information, see the *autoRequestEncoding and autoResponseEncoding* topic (cweb_autoreq in the InfoCenter).
- *Filters* are Java classes that can be configured to operate on (filter) the request and response data of a requested resource.

  The resource to filter, and filter precedence is specified in the deployment descriptor information found in the web.xml file of a Web application. Initialization parameters for filters can also be specified in the web.xml. Filters can be chained and can be configured to work on a single resource or a group of resources. Typical usages for filters include logging filters, image conversion filters, encryption filters, and MIME-type filters (functionally equivalent to the old style servlet chaining).

  For more information, see the *Servlet filtering* topic (cweb_sflt in the InfoCenter).
- *Application lifecycle events* give the application developer greater control over interactions with ServletContext and HttpSession objects.

  Application event objects consist of application events and application listeners. Servlet context listeners are used to manage resources at an application level. Session listeners manage resources associated with a series of request from a single client. Listeners are available for lifecycle events and for attribute modification events. The listener developer creates a class that implements the javax listener interface corresponding to the desired listener functionality.

  For more information, see the *Application lifecycle listeners and events* topic (cweb_sctxl in the InfoCenter).
- The HttpUtils class is deprecated in 2.3 and its methods are replaced by new methods in the request object. The HttpUtils class will still be available for use by servlet writers until a future servlet specification directs its complete removal.
- The product no longer requires the JSP-enabling servlet:

  The *file serving enabled* check box in the *IBM extensions* tab of the Web module properties in the Application Assembly Tool controls this function. (It is selected by default.)

  Adding JSP files to the WAR file in the Application Assembly Tool, or adding them to the appropriate `application_name.war` directory of the installed enterprise application causes them to be served.

Adding HTML files to the Web archive (WAR) file in the Application Assembly Tool, or adding them to the appropriate `application_name.war` directory of the installed enterprise application causes them to be served.

- New and improved features pertaining to **HTTP session support** include:
  - Multiple mechanisms for HTTP session state management, plus configuration options based on scalability and failover requirements from simple, single server environments to large, high-load clusters:
    - In memory
    - Persistent to database
    - Memory-to-memory
  - Replacement of the Session Manager object, which resided underneath each servlet engine in the WebSphere Application Server Version 4 topology. Its properties are now part of each Application Server.
  - Enhanced support for HTTP session state failover, as described in the *Managing HTTP sessions* topic (tprs_sep1 in the InfoCenter).
  - With Version 5, a new option exists for saving HttpSession information for failure recovery purposes. In addition to a database, IBM WebSphere Application Server can save a HttpSession in more than one Application Server instance. Called ″in-memory session replication,″ this feature leverages the replication domain and replicator entry services provided in Network Deployment
  - Session support for Wireless Application Protocol devices, as described in the *Configuring session tracking for Wireless Application Protocol (WAP) devices* (tprs_config_wireless in the InfoCenter).

CORBA applications

CORBA clients can access WebSphere Application Server CORBA C++ servers and WebSphere Application Server EJB servers. In addition, the product provides a basic CORBA environment that can ″bootstrap″ into the J2EE name space and invoke J2EE transactions. New support includes SSL security for CORBA C++ clients.

Web services applications

The new Web services standards are developed for the Java language under the Java Community Process (JCP). These standards include the Java API for XML-based remote procedure call (JAX-RPC (JSR-101)) and Web services for J2EE.

The JAX-RPC standard covers the programming model and bindings for using Web Services Description Language (WSDL) for Web services in the Java language. The Web services standard for J2EE covers the use of JAX-RPC in a J2EE environment, as well as the deployment of Web services implementations in a J2EE server. Both standards are part of the J2EE 1.4 release.

- Web services enable businesses to connect applications to other business applications, to deliver business functions to a broader set of customers and partners, to interact with marketplaces more efficiently, and to create new business models dynamically. To that extent, the product provides four protocols that support Web services:
  - Web Services Description Language (WSDL), an XML-based description language that provides a way to catalog and describe services
  - Universal Discovery Description and Integration (UDDI), a global, platform-independent, open framework to enable businesses to discover each other, define their interaction, and share information in a global registry
  - Simple Object Access Protocol (SOAP), a lightweight protocol for exchange of information in a decentralized, distributed environment
  - eXtensible Markup Language (XML), which provides a common language for exchanging information.

- Enhanced Web Services, including WSIF, WS-Security, and a Technology Preview of JSR109. New and improved features in **Web services** support include:
  - An open source implementation of a Web Services Invocation Framework (WSIF), new in this release. It includes protocol isolation and dynamic invocation (no stubs).
  - Newly-enhanced Web services capabilities of WebSphere Studio (sold separately) for developing Web services and Web services gateway filters.
  - Web services security functionality that is based on standards included in the Web services security (WS-Security) specification. Web services security is a message-level standard, based on securing Simple Object Access Protocol (SOAP) messages through XML digital signature, confidentiality through XML encryption, and credential propagation through security tokens. See *Securing Web services* for information on securing Web services.

Application services

Application service enhancements include:

- Changes and improvements in naming support are described in *New features for name space support*. They include:
  - The way that the system binds objects into the name space has changed significantly.

    In previous product versions, all objects were bound relative to a single root context. Now they are bound to a context that is specific to the server associated with the object. This context is referred to as the server root context. Each server has its own server root context. An initial context can be any server root context. This means that jndiName values in deployment descriptors and lookup names in thin clients must be qualified when the object associated with the name is bound under a server root context different from the initial context. For more information, see *Name space logical view* and *Lookup names support in deployment descriptors and thin clients*.
  - In Version 3.5.x and Version 4.x, the Name Server runs in the same process as the administrative server. An administrative server is no longer running on every Version 5 installation. The Name Server configuration is included in the same configuration files as Application Servers. The Name Server runs in its own process.
- Changes and improvements in **dynamic caching** include:
  - Version 4 supported the configuration of dynamic servlet caching through the use of a `servletcache.xml` file. For migration purposes, this file is still supported by this release. In order to utilize the new and improved functionality of the dynamic cache service in this release, you must configure your cache policy using the new `cachespec.xml` format. For more information, see the *Configuring cacheable objects with the cachespec.xml file* topic (tprf_dynamiccacheconfig in the InfoCenter).
  - Sophisticated dynamic network caching follows these directives, meaning explicit cache APIs are not needed:
    - Cache within the context of J2EE (such as Servlet, JSP, and EJB patterns)
    - Describe caching behavior in the form of XML cache policy files, providing a more flexible cache policy deployment descriptor

    Features of the dynamic caching engine include:
    - Disk overflow of cached objects through Java Object store/access (put and get)
    - Least Recently Used (LRU) management
    - XML cache policy management (such as the use of ID generation)
    - Invalidation management
    - External cache support for caches such as:
      - IBM WebSphere Edge caching using Akamai ESI
      - IBM HTTP Server Fast Response Cache Accelerator (FRCA)

    Caching support includes:

- Servlet and JSP results caching (same as Version 4.)
- Command caching (new)
- Pattern caching (new)
- Web services caching (new)

For more information, see the *Improving performance through the dynamic cache service* topic (tprf_dynamiccache in the InfoCenter).

- Internationalization allows applications to become global by determining the client locale and changing all relevant attributes, such as currency, character sets, and so on.

  For more information, see the *Using the internationalization service* topic (tin_ep) in the InfoCenter.

- Classloaders are new and improved. For more information, see the *Classloading* topic (trun_classload) in the InfoCenter.

- User profile support is deprecated. For more information, see the *Managing user profiles* topic (tprs_sep2) in the InfoCenter.

Applications and service choreography

Service choreography (including the process choreographer) is a new service-oriented approach to application development that provides integrated J2EE workflow capabilities. With this feature, developers can easily compose and choreograph application interactions and dynamic workflows among J2EE components, Web services, existing applications and systems and human activities.

For more information, see the *Using the process choreographer* (twork in the InfoCenter).

Resources and messaging

Enhancements to messaging include:
- Java Message Service (JMS) through embedded provider:
  - Supports point-to-point and publish/subscribe styles of messaging
  - Used for message-driven bean support
  - Integrated with transaction manager (JMS with XA)
  - Used for messaging within a cluster or cell
- Support for plugging in other JMS providers, including MQ Series
- Messaging and e-mail interfaces through JavaBeans Activation Framework (JAF), Remote Method Invocation over Internet InterORB Protocol (RMI/IIOP), JavaMail, and JMS with the help of IBM MQ Series
- Integrated Java Message Service, as described in the *Using asynchronous messaging* topic (tm_ep in the InfoCenter).

Resources and data access

Enhancements to data access include:
- **5.0.1** The administrative console pages for configuring data sources now contain buttons for testing the data source connections. See the *Test connection* topic (cdat_testcon in the InfoCenter).
- All connector access is through J2C:
  - JDBC access managed via J2 relational resource connector
  - Legacy JDBC support is provided
- Data access support provides a complete implementation of the JCA 1.0 specification, including support for:
  - Connection sharing

This version fully supports the res-sharing-scope tag within the resource reference (resource-ref) element, so the product supports both shareable and unshareable connections.

– Get/use/close and get/use/cache programming models for connection handles

The product supports the Web Container. Both EJB and Web components can utilize the J2EE Connector Architecture.

– XA, Local, and No Transaction models of resource adapters, including XA recovery
– Security options A and C per the specification
– Res-auth settings of either Application or Container.

In Version 4, the res-auth setting was disregarded. That is, it was treated as if the value of res-auth was set to Application. If your existing applications had res-auth set to Container, you might get different behavior if you install them into the new environment without any changes.

Applications must be packaged as J2EE 1.3 applications. For more information, see the *Migrating a version 4.0 data access application to version 5.0* topic (tdat_migdaapp in the InfoCenter).

- Subpools were eliminated to provide better performance. You can no longer specify Pool and Subpool names. The Pool name is based on the data source or JNDI name of the connection factory.
- J2EE Connector Architecture or J2C is part of J2EE V1.3.

J2C is similar to the Common Connector Framework (CCF) but is implemented for the Java platform. It provides specialized access to Enterprise Resource Planning (ERP) and mainframe systems such as CICS and IMS from IBM. As part of J2C, the product provides these components:

– Common Client Interface API, which simplifies access to diverse back-end Enterprise Information Systems (EIS)
– Resource Adapter, which enables the product to communicate with the back-end EIS. One-phase commit resource adapters are available for:
  - Host On-Demand
  - CICS
  - IMS
  - SAP
  - J.D. Edwards
  - PeopleSoft
  - Oracle Financial
– Connection Factory, which connects an application to the Resource Adapter

Security

- **5.0.1** The z/OS SecureWay LDAP server is now supported, using the same configuration as that of IBM SecureWay LDAP. The z/OS SecureWay LDAP server functions just like any other LDAP server currently supported by IBM WebSphere Application Server.
- Four administrative roles are now available for securing the administrative console. For more information, see the *Administrative console and naming service authorization* topic (csec_adminconsole in the InfoCenter).
- Enhanced security features include:
  – JAAS
  – CSIv2 interoperability
  – Java2 security
  – Support for third party Security Providers

For more information, see the *Welcome to Security* topic (welc_security in the InfoCenter).
- Distributed Systems Management, Security, and Directory Support
- J2EE 1.3 security support, including JAAS programming model and CSIv2 for CORBA interoperability

- Web services security includes signatures and credential propagation
- Support for third party security providers (prior to JSR 115)
- A new UserRegistry interface, to which Version 4 users should consider migrating from the deprecated CustomRegistry interface that was introduced in Version 4.
- The Trust Association Interceptor interface remains backward compatible with that of Version 4
- The application login helper functions provided in Version 4 and prior releases are deprecated, but still supported.
- The login helper functions are replaced by the JAAS LoginContext and Subject based programming model in Version 5.
- **5.0.2** WebSphere Application Server, Version 5.0.2 is integrated with Federal Information Processing Standards (FIPS) 140-2 certified cryptographic modules including Java Secure Socket Extension (JSSE) and Java Cryptography Extension (JCE).

Environment

Enhancements to the environment include:

- Configurable plug-ins for popular Web servers

  The Web server (or HTTP server) plug-in enables communication between the HTTP server and the Application Server. It uses the industry-standard HTTP transport protocol for non-secure transports and HTTPS for secure transports.

- The `plugin-cfg.xml` file location has changed to `install_root/config/cells/plugin-cfg.xml`.

  For more information, including format changes, see the *Configuring Web server plug-ins* topic (trun_plugin in the InfoCenter).

- For testing purposes, there are HTTP serving capabilities embedded within the product. For more information, see the *Configuring transports* topic (trun_plugin_transport in the InfoCenter).

- New variable support. Variables are configuration properties that can be used to provide a parameter for any value in the system. For more information, see the *Variables* topic (crun_variable in the InfoCenter).

- New shared library support, as described in the *Shared library files* topic (crun_sharedlib in the InfoCenter).

System administration

Enhancements to system administration include:

- Terminology for distributed systems management:
  - A cell is a collection of machines that you are managing together
  - A node is a machine on which you are running an Application Server
  - A server is the Java virtual machine running the Application Server containing your applications
- New, scalable XML-based administrative infrastucture:
  - All configuration data is stored in XML for standard deployment descriptors, and XMI format for product-specific configuration documents. These documents are stored on each node. No relational database is required. See the *Working with server configuration files* topic (trun_data) in the InfoCenter.
  - WebSphere Common Configuration Model (WCCM) documented API is provided for manipulating product configuration files
  - Servers load directly off of documents.
  - Application binaries are managed as part of the configuration repository.
  - In clusters, the product manages synchronization of documents across machines. This feature is configurable, as described in the *File synchronization service settings* topic (urun_rsynchservice in the InfoCenter).
- JMX support:

- Multiple protocol support (SOAP by default, but also RMI/IIOP)
- Support for alerts
- Message routing between machines, providing cell-level view
- Support for MBeans defined and registered by you
- Runtime attributes and access to run-time operations, configurations, and performance data

For more information, see the *Deploying and managing using programming* topic (txml_programming in the InfoCenter).

- Scripting support:
  - Based on Bean Scripting Framework (BSF), supports multiple scripting languages
  - Parallel capabillity between scripting and Web-based administrative console (see below)
  - Interactive and script modes
  - Multiple connection styles (SOAP, RMI)
  - Remote administration support
  - Ability to access any MBean registered in any server in the cell
  - Runtime attributes and access to run-time operations, configurations, and performance data

  For more information, see the *Deploying and managing using scripting* topic (txml_script in the InfoCenter).
- WebSphere administrative console:
  - Access to configuration data and to operations, run-time state
  - Multi-user support, with ability to customize based on user preferences. Coarse-grain administrative security control and filtering for roles such as Administrator, Monitor, Configurator, and Operator as described in the *Administrative console and naming service authorization* topic (csec_adminconsole in the InfoCenter).
  - Filtering and search for collections
  - Inclusion in all product packages — Version 4.0 Java-based console no longer available. Additional features are added as additional product packages are installed
  - Struts and Tiles implementation
  - Display of run time and configuration exceptions. You can toggle between them.

  For more information, see the *Using the administrative console* topic (trun_console in the InfoCenter).
- A stable of command line tools are now available for specific tasks. For more information, see the *Managing using command line tools* topic (txml_command in the InfoCenter).

Monitoring and tuning performance

Enhancements to monitoring and performance tuning include:

- **5.0.2** The Runtime Performance Advisor and the Performance Advisor in Tivoli Performance Viewer are new tools that suggest configuration changes to help administrators optimize performance on WebSphere Application Server. For more information, see *Welcome to Monitoring and Troubleshooting* in the InfoCenter.
- **5.0.1** Tuning information has been enhanced to include the handy:
  - Parameter hotlist, key parameters you can tune for a high impact on performance
  - Parameter index, the comprehensive list of tuning parameters
  - Troubleshooting tips, a concise guide that traces various problems to their possible origins in poor tuning
- The Performance Monitoring Infrastructure (PMI):
  -

– Is now integrated with Java Management eXtensions (JMX), as shown in the *Performance Monitoring Infrastructure* topic (cprf_pmidata) in the InfoCenter.
– Has new counters including those for:
    - Dynamic caching
    - Workload management
    - Object Request Broker (ORB)
    - HTTP session size
    - JDBC time
    - **5.0.2** Web Services
    - CPU utilization

    For more information about data, see the *Performance data organization* topic (rprf_dataorg in the InfoCenter).
– Supports both the PmiClient interface and the JMX interface. The latter is supported through the AdminClient class described in the *Developing an administrative client program* topic (txml_develop in the InfoCenter).
– Continues to support the Version 4.0-style APIs, but the data hierarchy has been updated to match the Version 5 product structure
- The resource analyzer has been rebranded as Tivoli Performance Viewer and bundled with WebSphere Application Server. New features include logging and replay in XML format, and CPU utilization.

    For more information, see the *Monitoring performance with Tivoli Performance Viewer (formerly Resource Analyzer)* topic (tprf_tpvmonitor in the InfoCenter).
- Request metrics is instrumentation that tracks the time spent by selected requests in each WebSphere Application Server component in the system. The data can be written to a log file or sent to an ARM agent.

    For more information, see the *Measuring data requests (Performance Monitoring Infrastructure Request Metrics)* topic (tprf_requestmetrics in the InfoCenter).
- Performance features include:
    – Dynamic, multi-tier caching, which is set up per node or Application Server using XML files and is most effective for non-user specific output such as mutual fund prices
    – Dynamic reloading of enterprise beans
    – JNDI caching, which improves performance by caching expensive lookups. See the *JNDI caching* topic (cnam_naming_caching in the InfoCenter).
    – Caching of dynamic content, such as servlets and JSP files, to improve throughput
- The product can be tuned from the WebSphere Administrative Console.

Troubleshooting

- Enhanced Problem Determination features, including FFDC (First Failure Data Capture):
    – Collects data based on the first failure in the system
    – Filters out expected or recurring exceptions to reduce overhead in collecting data
    – Passes data to an analysis engine that searches a knowledge base of information about common errors, including their possible causes and solutions

    For more information, see the *Working with troubleshooting tools* topic (ttrb_trbtls in the InfoCenter).
- RAS Collector Tool gathers information to send to IBM Service personnel. For more information, see the *Collector Tool* topic (ctrb_ct in the InfoCenter).
- The RAS collector summary option is a lightweight version of the RAS collector tool. It is useful for initial problem reporting to IBM Service personnel. For more information, see *Collector summary option*.
- Improved messages. To view message documentation, click **Messages** in the **Quick reference** view of the InfoCenter.

Chapter 3. Installing WebSphere Application Server

Perform the following tasks to install the product on your machine.

Steps for this task

1. Plan to install an e-business network.

   This task helps you plan for installing an e-business network. It also describes interoperability considerations.

2. **(Optional)** Install a Web server on a separate machine.

   To install the IBM HTTP Server, or another Web server, on a separate machine, install it now. You can also install the IBM HTTP Server using the installation program in the `IHS` directory on the WebSphere Application Server product CD-ROM.

3. Install the product.

   This task helps you prepare your machine for installation and explains the different types of installation available to you. It includes information on:

   - Using the LaunchPad
   - Deciding whether to migrate applications and the configuration from a previous version
   - Deciding whether to coexist with a previous version
   - Using the silent installation method

   This task describes using the installation wizard. Through the installation wizard, you migrate applications and configurations from a previous version of WebSphere Application Server, coexist with the previous version, choose a typical or custom installation, and perform some initial configuration.

   The wizard lets you install and configure IBM HTTP Server. You can also select Web server plug-ins for IBM HTTP Server and other popular Web servers.

4. Verify the WebSphere Application Server installation.

   a. Use the First Steps tool to run the Installation Verification Test.

      The First Steps tool starts automatically at the end of the installation.

   b. Identify and correct any problems using the troubleshooting procedure.

5. **(Optional)** Examine the results of migration, or prepare to perform a manual migration.

   This task describes exactly what is migrated during the automatic migration. It also describes how to perform a manual migration using the migration tools.

6. **(Optional)** Prepare to migrate from an unsupported operating system.

   This task describes a basic procedure to follow when migrating from one operating system to another, taking into account the possibility that you might have to format a drive, for example.

7. Configure WebSphere Application Server after migration.

   This task explains how to check migrated application and configuration information, to understand and configure exactly what you migrated.

   - If you migrate from V3.5.x, examine the applications you are moving. Make any necessary changes to the applications, which were converted to Java 2 Platform, Enterprise Edition (J2EE) platform applications. The migration tools create the initial J2EE enterprise applications, based on V3.5.x configurations.
   - If you migrate from V4.x, there is little to review. J2EE 1.2 enterprise archive (EAR) files in V4 work in V5 of WebSphere Application Server, which also supports J2EE 1.3.

8. **(Optional)** Set up V3.5.x and V5 coexistence.

   This task describes running WebSphere Application Server V3.5.5 and later editions with V5, on the same machine.

9. **(Optional)** Set up V4.0.x and V5 coexistence.

   This task describes running WebSphere Application Server V4.0.2 and later editions with V5, on the same machine.

10. **(Optional)** Set up V5 coexistence.

    This task describes running multiple WebSphere Application Server V5 installations on the same machine.

11. **(Optional)** Manually configure a Web server, for example, to support multiple WebSphere Application Server versions or multiple V5 instances.

    This task describes migrating plug-ins and how to configure a plug-in for V5.

12. **(Optional)** Automatically restart WebSphere Application Server processes.

    This task describes how to set up WebSphere Application Server processes for the operating system to monitor and restart.

13. **(Optional)** Create multiple V5 instances on one machine.

    This task describes creating multiple configuration instances from one installation. It also describes creating multiple servers in a coexistence or multiple instance environment, and changing port settings for HTTP transport to avoid conflicts.

14. **(Optional)** Apply an interim fix or fix pack to an existing installation.

    This task helps you apply maintenance to WebSphere Application Server products after you download the maintenance from the Support page.

15. **(Optional)** Remove an interim fix or fix pack from an existing installation.

    This task helps you remove applied maintenance from WebSphere Application Server products.

You can install the WebSphere Application Server product.

Uninstalling and reinstalling: If you must uninstall the product, follow the steps in Uninstalling WebSphere Application Server. After uninstalling WebSphere Application Server, reinstalling into the same directory without first deleting all directory contents, results in invalid XML configurations because of old file retention. To delete all files so that you can reinstall with a clean system, perform a manual uninstall, as described inUninstalling WebSphere Application Server.

WebSphere Application Server packages

The WebSphere Application Server family of interoperable products provides a next-generation Application Server on an industry-standard foundation. The IBM WebSphere Application Server family is divided into four packages for V5, to better address requirements you might have. Each edition addresses a distinct set of scenarios and needs. WebSphere Application Server includes:

- **WebSphere Application Server Express** (not pictured)

  This edition is a lightweight server for static content, servlets, and JSP pages, but does not support enterprise beans.

- **WebSphere Application Server**

  This edition addresses the basic programming and execution needs of desktop developers and single-server production scenarios. The execution environment for this edition addresses standards-based programming for Web and component-based programming, as well as Web services.

  The administration model for this edition presumes a single-server environment - no clustering for failover or workload balancing, nor centralized administration of multiple server instances. However, you can add a stand-alone node to a centrally administered network (the cell) at any time after installing the next product, which controls the cell.

- **WebSphere Application Server Network Deployment**

  This edition addresses Application Server execution in departmental computing scenarios. It provides centralized administration of multiple server instances, as well as basic clustering and caching support.

- **WebSphere Application Server Enterprise**

This edition addresses large information technology (I/T) production scenarios for applications that are designed according to the core, standards-based programming model of Java 2 Platform, Enterprise Edition (J2EE) and Web services. It supports large-scale clustering, caching, content distribution, and dynamic workload management to help gain efficient utilization of shared resources in the I/T computing center. It also addresses complex programming requirements for integrating applications and lines of business, by introducing super-standard programming functions for business process management, integration adapters, rules management, and message transformations.

- **WebSphere Application Server for z/OS** (not pictured)

This edition integrates the WebSphere Application Server product and the Network Deployment product into a single package that runs on a z/OS platform. It addresses Application Server execution in departmental computing scenarios, and provides centralized administration of multiple server instances, as well as basic clustering and caching support.
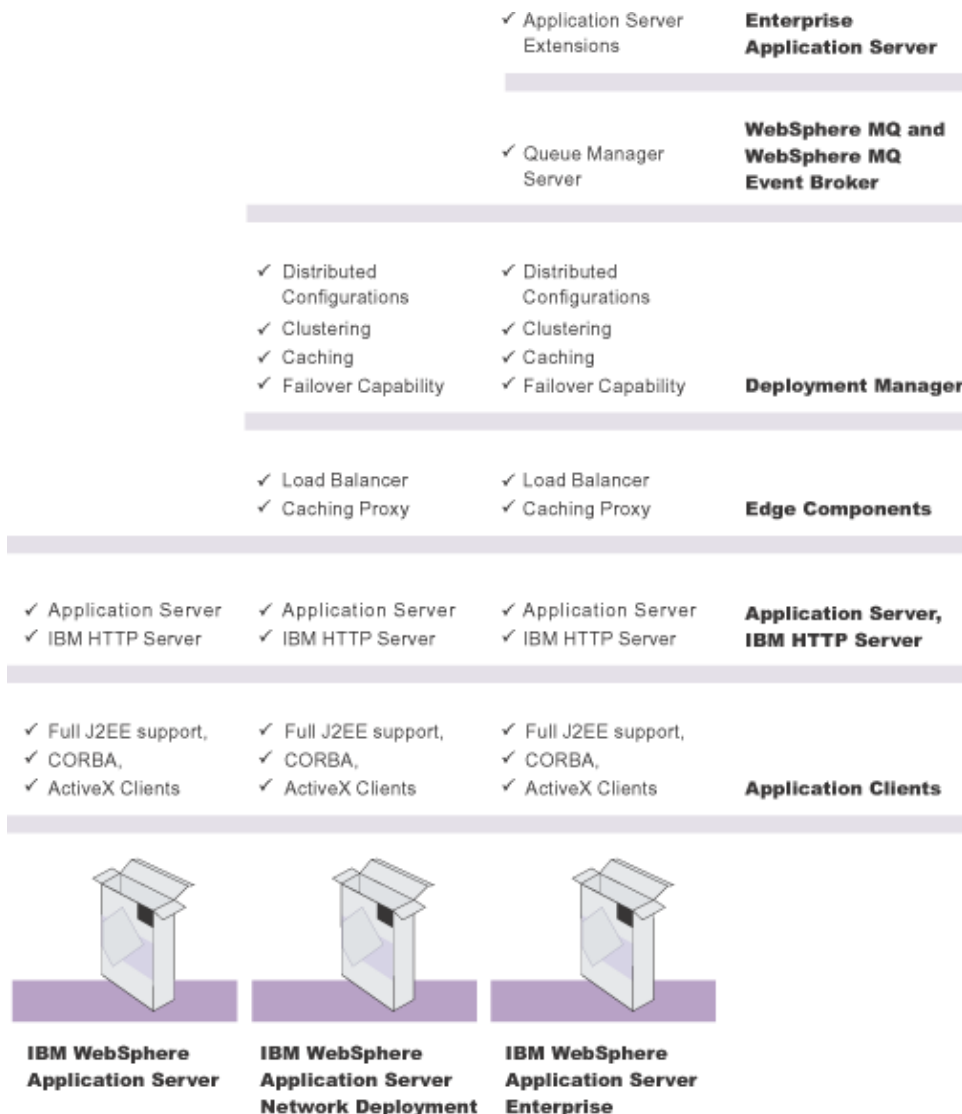
Installation images in WebSphere Application Server packages

The WebSphere software platform for e-business starts with a foundation formed from Web application serving and integration. The WebSphere Application Server family lets you quickly, reliably and flexibly enable your business for the Web. It provides the core software to deploy, integrate and manage your e-business applications. WebSphere Application Server supports custom-built applications, based on

integrated WebSphere software platform products, or on other third-party products. Such applications can range from dynamic Web presentations to sophisticated transaction processing systems.

IBM WebSphere Application Server, WebSphere Application Server Network Deployment, and WebSphere Application Server Enterprise are interoperable building blocks. The base product CD-ROMs are included in the Network Deployment package. The Network Deployment package CD-ROMs are included in the Enterprise package.

- The **Clients** installation image contains support for Java (J2EE) Application client 2, CORBA and ActiveX client run times.
- The **WebSphere Application Server** product installation image contains the core Application Server runtime, a native JMS provider (the embedded messaging feature), IBM HTTP Server, IBM Developer Kit, IBM Cloudscape, XML and XSL parsers, the Application Assembly Tool (AAT), the deployment tool, the node agent for communicating to the deployment manager when it is part of a cell, and the external adapter library for proxy caching enablement.
- The **Deployment Manager** product installation image contains the deployment manager configured for use in departmental production computing scenarios. It includes the embedded messaging client feature, the V2 compliant universal description, discovery, and identification (UDDI) registry feature, and the Web services gateway feature. Although the Network Deployment package includes the base Application Server CD-ROM, installing the Network Deployment product does not install the base Application Server product. You must use the base product CD-ROM to install the base product.
- The **Edge components** installation image contains IBM HTTP Server, and edge of network support for the Load Balancer (Dispatcher) and Caching Proxy (edge caching), as well as support for network authentication and single sign-on.
- The **Enterprise** installation image contains programming model extensions to the core Application Server, such as Business Rule Beans and Business Policy Management, and deployment manager extensions for administering functions included in the programming model extensions. The Enterprise product supports an *umbrella installation*, which automatically installs the base product in the same installation procedure.

  Although the Enterprise package includes the Network Deployment product CD-ROM, installing the Enterprise product does not install the Network Deployment product. You must use the Network Deployment product CD-ROM to install the Network Deployment product.

  If you install the Enterprise product on a machine with an installed Network Deployment product, the Enterprise product extends the Network Deployment administrative console.

  Although the Enterprise package includes a separate base WebSphere Application Server product CD-ROM, installing the Enterprise product on a clean machine automatically installs the base Application Server product in an umbrella installation.

  If you install the Enterprise product on a machine with an installed base Application Server product, the Enterprise product extends the base Application Server product, while automatically installing any base product features that the Enterprise product requires.
- The **WebSphere MQ and WebSphere MQ Event Broker** installation images provide the non-embedded full-function WebSphere MQ Queue Manager for reliable, dynamically load balanced asynchronous messaging for more than 35 platforms.

Included with all packages are the Data Direct Technologies JDBC Drivers and the Application Server Toolkit CD-ROMs.

5.0.1 Also included with the base product and the Network Deployment product are the WebSphere Application Server, V5 for iSeries CD-ROMs. The Enterprise product is not supported on iSeries servers at this time.

In addition, the WebSphere Application Server Network Deployment and Enterprise packages include the IBM Directory Server and the DB2 Universal Database Enterprise Edition products.

Planning to install an e-business network

Version 5 of the WebSphere Application Server family provides flexible configurations and deployment options for hosting Java 2 Platform, Enterprise Edition (J2EE) applications. The WebSphere Application Server package provides all required components:

- The WebSphere Application Server
- A Web server
- A J2EE client

Node

(o) = Application Server,
         IBM HTTP Server CD

[   ] = Application Servers

**IBM WebSphere Application Server
Package**
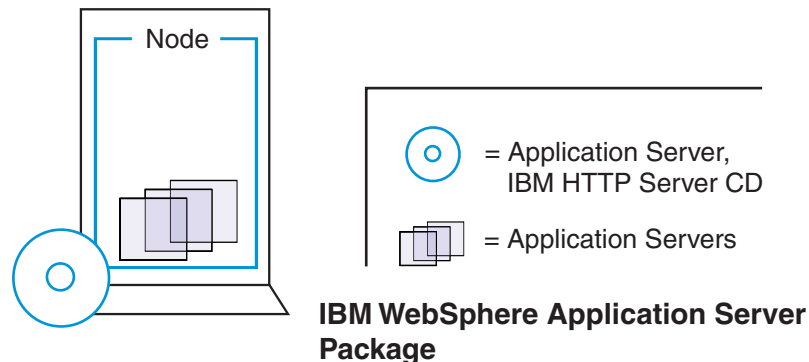
The WebSphere Application Server Network Deployment package includes the core package and provides a smooth transition to deploying and managing applications in a distributed network environment. It includes productivity and scalability enabling components (Edge Components). The deployment manager server lets you easily federate single nodes to its group, which it manages as a single image or *cell*, by using the nodeagent server process on each node. Open the InfoCenter for the Network Deployment product to learn how to establish a multimachine environment, and to learn about WebSphere Application Server, V5 product scalability.

Tasks in the installation planning process appear below.

Steps for this task

1. Plan the general scope of your network.

   Review single machine topologies to learn how you can install and use the base WebSphere Application Server product on a single machine. Open the InfoCenter for the Network Deployment product to learn how to establish a multimachine environment.

2. Plan for interoperability and coexistence.

   Plan to have WebSphere Application Server interoperate with your other e-business systems, including other versions of WebSphere Application Server. *Interoperability* provides a communication mechanism for WebSphere Application Server nodes that are at different versions. *Coexistence* is another term in common use in this InfoCenter. It describes multiple versions or instances running on the same machine, at the same time.

   Interoperability support enhances migration scenarios with more configuration options. It often is convenient or practical to interoperate during the migration of a configuration from an earlier Application Server version to a later one when some machines are at the earlier version and some at the later one. The mixed environment of machines and application components at different software version levels requires interoperability and coexistence.

   It is often impractical, or even physically impossible, to migrate all the machines and applications within an enterprise at the same time. Understanding multiversion interoperability and coexistence is therefore an essential part of a migration between version levels.

   a. Plan to run WebSphere Application Server across platforms.

Support of multiple versions is provided on all operating system platforms supported by WebSphere Application Server, V5.

  b. Plan to run WebSphere Application Server across versions.

WebSphere Application Server, V5 is generally interoperable with WebSphere Application Server Versions 3.5.x and 4.0.x, although each version has specific requirements. However, the ability to run different versions of an Application Server in a configuration does not let you include V5 Application Servers in an existing administrative domain, or let you include V3.5.x or V4.0.x Application Servers in a V5 cell.

Single server topology

The following illustrations show examples of single server topologies. Each WebSphere Application Server product can run in a single server environment. The most common topology is a stand-alone base WebSphere Application Server product.

Client HTTP requests — optional → HTTP server / Plug-in — optional → Application server
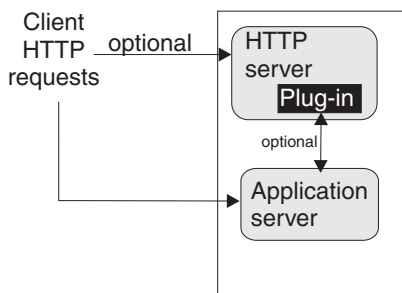
The WebSphere Application Server base product runs on a single machine. You can install the product in a stand-alone configuration or as part of a cell in a multimachine configuration. The stand-alone configuration is typically for developer desktops or stand-alone production computing, which involve a single Application Server instance operating independently of any other applications.

You can install an Application Server installation image on any supported machine. You can use the Application Server to host one or more applications, configuring it through the administrative console.

Multiple instances on one machine

There are three main topologies for the base WebSphere Application Server product:
- A single installation, as described above
- Multiple installations in a coexistence environment
- A single installation with multiple configuration instances

You can install the base WebSphere Application Server product multiple times in separate directories. Each resulting installation instance is a fully functional Application Server. The following illustration shows an example of such a configuration.

Client HTTP requests → HTTP server / Plug-in → Application server / HTTP server / Plug-in → Application server / Application server / Application server
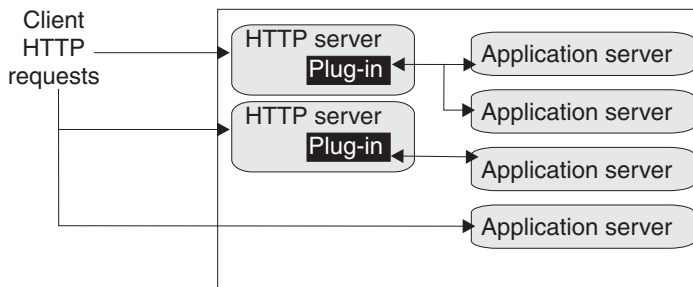
You can also install the base WebSphere Application Server product one time and use the **wsinstance** command to create multiple configuration instances. Configuration instances are fully functional Application Servers that share the run time and command binaries of the initial product installation. The following illustration shows an example of such a configuration.
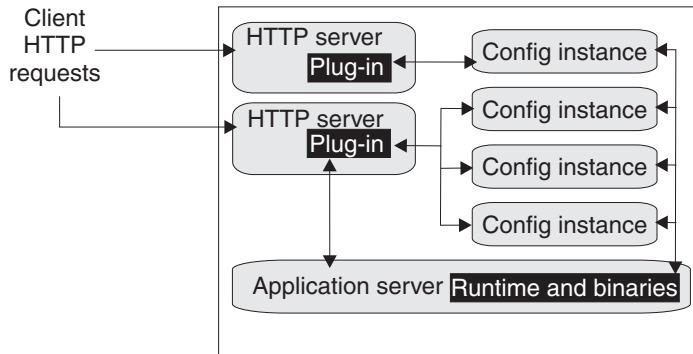
Example: Choosing a topology for better performance

WebSphere Application Server provides various Workload Management (WLM) topologies. The following case study uses two topologies, Topology A and Topology B to show how the type of topology you choose can affect performance. In the case study, Topology A can improve performance from 10%-20% more than Topology B. You can see performance increase by using the J2EE benchmark Trade, which is included in this release.

Topology A contains a Web server and a WebSphere Application Server plug-in to a cluster of WebSphere Application Servers. Each cluster member contains a Web container and EJB container. Topology B includes a Web server, a plug-in, and one Web container to a cluster of EJB containers. In both topologies, the Object Request Broker pass-by-reference is selected and the backend database is on a dedicated machine.

Topology A has an advantage because the Web container and EJB container are running in a single Java virtual machine (JVM). In Topology B, the Object Request Broker pass-by-reference option is ignored between the Web container cluster member and the EJB container member. In Topology A, the EJB container uses the same thread passed from the Web container. The request does not have to be passed from one thread in one JVM to another thread in another JVM.

Also, if the processor utilization of the cluster member machines is near 100% you can add more members. If the Web server box is not running at capacity and the Web container processing is not heavy, try freeing the processor on the other members by moving to Topology B.

In this test environment, Topology A had the advantage, however, many factors related to the application and environment can influence results.

Queuing network

WebSphere Application Server contains interrelated components that must be harmoniously tuned to support the custom needs of your end-to-end e-business application. These adjustments help the system achieve maximum throughput while maintaining the overall stability of the system. This group of interconnected components is known as a queuing network. These queues or components include the network, Web server, Web container, EJB container, data source, and possibly a connection manager to a custom back-end system. Each of these resources represents a queue of requests waiting to use that resource. Various queue settings include:

- IBM HTTP Server: MaxClients for UNIX and ThreadsPerChild for Windows systems.

- Web container: Thread pool Maximum size, HTTP transports MaxKeepAliveConnections and MaxKeepAliveRequests
- Object Request Broker
- Data source Connection pooling and Statement cache size
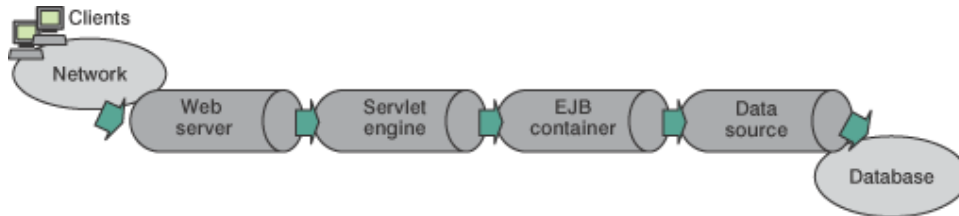
Figure Reference 1. WebSphere queuing network

Most of the queues that make up the queuing network are closed queues. A closed queue places a limit on the maximum number of requests present in the queue, while an open queue has no limit. A closed queue supports tight management of system resources. For example, the Web container thread pool setting controls the size of the Web container queue. If the average servlet running in a Web container creates 10MB of objects during each request, a value of 100 for thread pools limits the memory consumed by the Web container to 1GB.

In a closed queue, requests can be active or waiting. An active request is doing work or waiting for a response from a downstream queue. For example, an active request in the Web server is doing work, such as retrieving static HTML, or waiting for a request to complete in the Web container. A waiting request is waiting to become active. The request remains in the waiting state until one of the active requests leaves the queue.

All Web servers supported by WebSphere Application Server are closed queues, as are WebSphere Application Server data sources. You can configure Web containers as open or closed queues. In general, it is best to make them closed queues. EJB containers are open queues. If there are no threads available in the pool, a new one is created for the duration of the request.

If enterprise beans are called by servlets, the Web container limits the number of total concurrent requests into an EJB container, because the Web container also has a limit. The Web container limits the number of total concurrent requests only if enterprise beans are called from the servlet thread of execution. Nothing prevents you from creating threads and bombarding the EJB container with requests. Therefore, servlets should not create their own work threads.

Queuing and clustering

Cloning Application Servers can be a valuable asset in configuring highly-scalable production environments, especially when the application is experiencing bottlenecks that are preventing full CPU utilization of symmetric multiprocessing (SMP) servers. When adjusting the WebSphere Application Server system queues in clustered configurations, remember that when a server is added to a cluster, the server downstream receives twice the load.
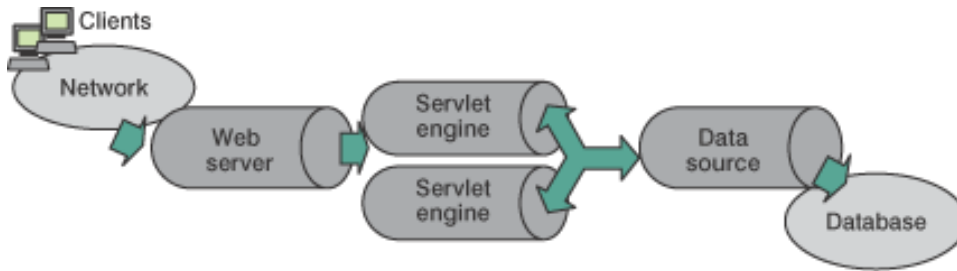
Figure Reference 1. Clustering and Queuing

Two servlet engines are located between a Web server and a data source. It is assumed that the Web server, servlet engines and data source, but not the database, are all running on a single SMP server. Given these constraints, the following queue considerations must be made:

- Double the Web server queue settings to verify ample work is distributed to each Web container.
- Reduce the Web container thread pools to avoid saturating a system resource like CPU or another resource that the servlets are using.
- Reduce the data source to avoid saturating the database server.
- Reduce Java heap parameters for each instance of the Application Server. For versions of the Java virtual machine (JVM) shipped with WebSphere Application Server, it is crucial that the heap from all JVMs remain in physical memory. For example, if a cluster of four JVMs is running on a system, enough physical memory must be available for all four heaps.

Queue configuration tips

The following section outlines a methodology for configuring the WebSphere Application Server queues. Moving the database server onto another machine or providing more powerful resources, for example a faster set of CPUs with more memory, can dramatically change the dynamics of your system.

There are four tips for queuing:

- **Minimize the number of requests in WebSphere Application Server queues.**

  In general, requests wait in the network in front of the Web server, rather than waiting in WebSphere Application Server. This configuration only supports those requests that are ready for processing to enter the queuing network. Specify that the queues furthest upstream or closest to the client are slightly larger, and queues further downstream or furthest from the client are progressively smaller.
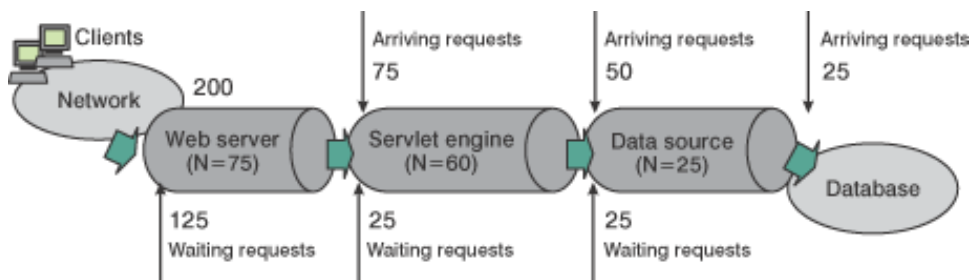
Figure Reference 1. Upstream queuing network

Queues in the queuing network become progressively smaller as work flows downstream. When 200 client requests arrive at the Web server, 125 requests remain queued in the network because the Web server is set to handle 75 concurrent clients. As the 75 requests pass from the Web server to the Web container, 25 requests remain queued in the Web server and the remaining 50 are handled by the Web container. This process progresses through the data source until 25 user requests arrive at the final destination, the database server. Because there is work waiting to enter a component at each point upstream, no component in this system must wait for work to arrive. The bulk of the requests wait in the

network, outside of WebSphere Application Server. This type of configuration adds stability, because no component is overloaded. You can then use the Edge Server to direct waiting users to other servers in a WebSphere Application Server cluster.

- **Draw throughput curves to determine when the system capabilities are maximized.**

  You can use a test case that represents the full spirit of the production application by either exercising all meaningful code paths or using the production application. Run a set of experiments to determine when the system capabilities are fully stressed or when it has reached the saturation point. Conduct these tests after most of the bottlenecks are removed from the application. The goal of these tests is to drive CPUs to near 100% utilization. For maximum concurrency through the system, start the initial baseline experiment with large queues. For example, start the first experiment with a queue size of 100 at each of the servers in the queuing network: Web server, Web container and data source. Begin a series of experiments to plot a throughput curve, increasing the concurrent user load after each experiment. For example, perform experiments with one user, two users, five, 10, 25, 50, 100, 150 and 200 users. After each run, record the throughput requests per second, and response times in seconds per request. The curve resulting from the baseline experiments resembles the following typical throughput curve shown as follows:

Throughput curve
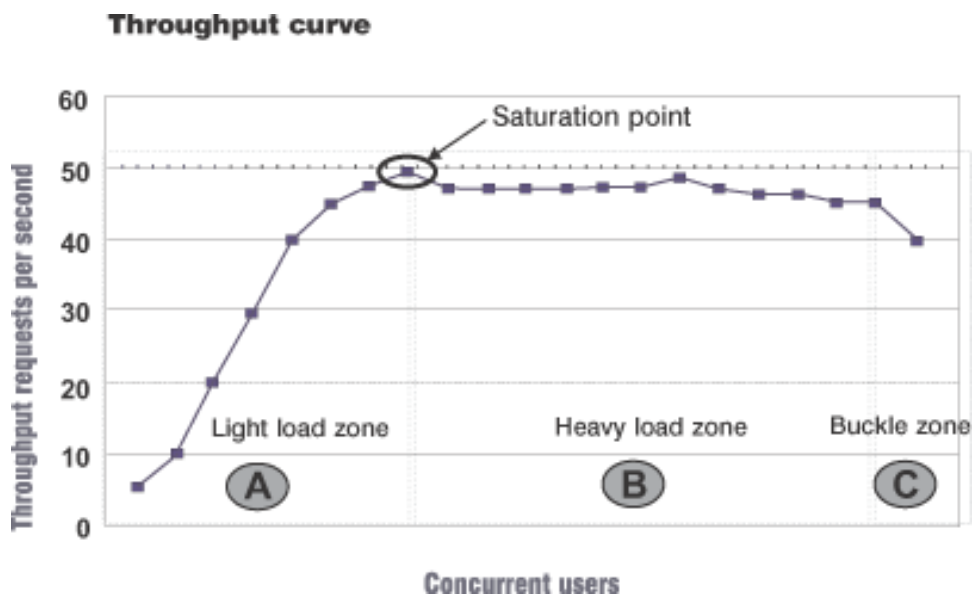
The WebSphere Application Server throughput is a function of the number of concurrent requests present in the total system. Section A, the light load zone, shows that the number of concurrent user requests increases, the throughput increases almost linearly with the number of requests. At light loads, concurrent requests face very little congestion within the WebSphere Application Server system queues. At some point, congestion starts to develop and throughput increases at a much lower rate until it reaches a saturation point that represents the maximum throughput value, as determined by some bottleneck in the WebSphere Application Server system. The most manageable type of bottleneck occurs when the WebSphere Application Server machine CPUs become fully utilized because adding CPUs or more powerful CPUs fixes the bottleneck.

In the heavy load zone or Section B, as the concurrent client load increases, throughput remains relatively constant. However, the response time increases proportionally to the user load. That is, if the user load is doubled in the heavy load zone, the response time doubles. At some point, represented by Section C, the buckle zone, one of the system components becomes exhausted. At this point, throughput starts to degrade. For example, the system might enter the buckle zone when the network connections at the Web server exhaust the limits of the network adapter or if the requests exceed operating system limits for file handles.

If the saturation point is reached by driving CPU utilization close to 100%, you can move on to the next step. If the saturation CPU occurs before system utilization reaches 100%, it is likely that another bottleneck is being aggravated by the application. For example, the application might be creating Java objects causing excessive garbage collection bottlenecks in the Java code.

There are two ways to manage application bottlenecks: remove the bottleneck or clone the bottleneck. The best way to manage a bottleneck is to remove it. You can use a Java-based application profiler, such as WebSphere Studio Application Developer, Performance Trace Data Visualizer (PTDV), Borland's Optimizeit, JProbe or Jinsight to examine overall object utilization.

- **Decrease queue sizes while moving downstream from the client.**

  The number of concurrent users at the throughput saturation point represents the maximum concurrency of the application. For example, if the application saturates WebSphere Application Server at 50 users, using 48 users might produce the best combination of throughput and response time. This value is called the Max Application Concurrency value. Max Application Concurrency becomes the preferred value for adjusting the WebSphere Application Server system queues. Remember, it is desirable for most users to wait in the network; therefore, queue sizes should increase when moving downstream farther from the client. For example, given a Max Application Concurrency value of 48, start with system queues at the following values: Web server 75, Web container 50, data source 45. Perform a set of additional experiments adjusting these values slightly higher and lower to find the best settings.

  To help determine the number of concurrent users, view the Servlet Engine Thread Pool and Concurrently Active Threads metric in the Tivoli Performance Viewer.

- **Adjust queue settings to correspond to access patterns.**

  In many cases, only a fraction of the requests passing through one queue enters the next queue downstream. In a site with many static pages, a number of requests are fulfilled at the Web server and are not passed to the Web container. In this circumstance, the Web server queue can be significantly larger than the Web container queue. In the previous example, the Web server queue was set to 75, rather than closer to the value of Max Application Concurrency. You can make similar adjustments when different components have different execution times.

  For example, in an application that spends 90% of its time in a complex servlet and only 10% of its time making a short Java database connectivity (JDBC) query, on average 10% of the servlets are using database connections at any time, so the database connection queue can be significantly smaller than the Web container queue. Conversely, if the majority of servlet execution time is spent making a complex query to a database, consider increasing the queue values at both the Web container and the data source. Always monitor the CPU and memory utilization for both the WebSphere Application Server and the database servers to verify that the CPU or memory are not saturating.

Running WebSphere Application Server across versions

WebSphere Application Server, V5 is generally interoperable with WebSphere Application Server, Versions 3.5.x and 4.0.x. However, there are specific requirements to address for each version. Make the following changes to support interoperability between versions.

Steps for this task
1. Apply required fixes.

Table 1. Fixes to apply for V3.5.x

Fix	V3.5.3	V3.5.4	V3.5.5	V3.5.6
PQ51387	Apply	Apply		
PQ60074	Apply	Apply	Apply	Apply
PQ60335			Apply	
PQ63548	Apply	Apply	Apply	Apply

Table 2. Fixes to apply for V4.0.x

Fix	V4.0.1	V4.0.2	V4.0.3
PQ60074	Apply	Apply	
PQ60336	Apply	Apply	
PQ63548	Apply	Apply	Apply

All fixes are available on the IBM WebSphere Application Server Support page.

Fix PQ51387:
A naming client fix that supports V3.5.x naming client access to the V5 name server.

Fix PQ60074:
An Object Request Broker (ORB) fix that supports V5 naming client access to the V3.5.x or 4.0.x name server. A down-level client has no problem accessing a V5 name server, even using corbaloc.

Fix PQ60335:
An ORB fix to reconcile `java.math.BigDecimal` and other class differences in IBM Software Development Kits 122 and 131.

Note: This fix does not apply to IBM Software Development Kits on the Solaris operating system.

Fix PQ60336:
An ORB fix to reconcile `java.math.BigDecimal` and other class differences in IBM Software Development Kits 130 and 131.

Note: This fix does not apply to IBM Software Development Kits on the Solaris operating system.

Fix PQ63548:
A fix to correct problems that might occur when passing embedded valueTypes between WebSphere Application Server releases.

The best solution is to upgrade all your installations to the latest release and PTF levels, such as Versions 3.5.7 or 4.0.4, which do not require this fix. If this solution is not possible, apply the fix to your version.

Symptoms include `org.omg.CORBA.MARSHAL` exceptions when passing embedded valueTypes across the versions. Sometimes, other symptoms might mask `org.omg.CORBA.MARSHAL` exceptions, which makes them difficult to identify.

If symptoms reoccur in spite of the fix, re-export existing IORs.

2. Follow required guidelines.

Table 3. Guidelines to apply for V3.5.x and V4.0.x

Guideline	V3.5.x	V4.0.x
1	Apply	
2	Apply	Apply
3	Apply	
4	Apply	
5	Apply	Apply
6	Apply	Apply

Guideline 1:
Use the context of the lowest common denominator when interoperating at the naming level.

For example, always use the 3.5.x context com.ibm.ejs.ns.jndi.CNInitialContextFactory when a client or server is at 3.5.x. For later versions, use the current com.ibm.websphere.naming.WsnInitialContextFactory context.

Guideline 2:

Make required naming changes to support V3.5.x or 4.0.x client access to V5 enterprise beans. This issue is new, introduced by V5. There are several ways to make it work, such as:

- Updating the `namebindings.xml` file if you do not use the V5 migration tools, but have installed V3.5.x or 4.0.x applications on V5. To allow V3.5.x or 4.0.x client access to the applications, add additional information to the bind information in the V5 namespace to make the old JNDI names work. Add the information to the `namebindings.xml` configuration file at the cell level using the administrative console.

  Applications that you migrate to V5 during installation, or that you manually migrate using the WASPreUpgrade and WASPostUpgrade migration tools, already have this update.

- Calling V5 enterprise beans directly using the naming path that includes the server on which the enterprise beans are running, such as `cell/node/server1/some/ejb/name`, for example.

- Using the V4.0.x client `java:/comp` location to find V5 enterprise beans. (You cannot use the command from a V3.5.x client.)

Guideline 3:

Verify that programs performing a JNDI lookup of the `UserTransaction` interface, use an InitialContext that resolves to a local implementation of the interface. Also verify that such programs use a JNDI location appropriate for the enterprise bean version.

Prior to the EJB 1.1 Specification, the JNDI location of the UserTransaction interface was not specified. Earlier versions up to and including V3.5.x do not use the EJB 1.1 Specification. They bind the UserTransaction interface to a JNDI location of `jta/usertransaction`.

Version 4, and later releases, bind the UserTransaction interface at the location defined by the EJB 1.1 Specification, which is `java:comp/UserTransaction`.

Version 5 no longer provides the earlier `jta/usertransaction` binding within Web and EJB containers to applications at a J2EE level of 1.3 or later, to enforce use of the newer UserTransaction interface. For example, EJB 2.0 applications can use only the `java:comp/UserTransaction` location.

Guideline 4:

Be aware of limitations when calling WorkLoad Management (WLM)-enabled enterprise beans.

Some clients cannot call WLM-enabled enterprise beans on remote clusters when there is a local WLM-enabled enterprise bean of the same name. If there is a cluster local to the client with the same enterprise bean as the remote cluster that the client is trying to call, the client ends up talking to the local cluster. The following table lists supported combinations of clients calling WLM-enabled enterprise beans on remote Application Servers.

All clients at Version:	Server at Version:	Supported interoperability
3.5.6	5	Yes
4.02, 4.03	5	Yes
5	3.5.x	No
5	4.02, 4.03	Yes

Guideline 5:

Be aware of administrative console limitations.

You cannot use the administrative interfaces for V5 to administer a V3.5.x or 4.0.x administrative server. Likewise, you cannot use a V3.5.x or V4.0.x administrative console to administer a V5 environment. If you use the administrative console on a remote machine to

administer V3.5.x or 4.0.x WebSphere Application Server domains, migrating any of the nodes or domains to V5 renders the remote administration console ineffective for administering any V5 environment.

Guideline 6:
Use Secure Socket Layer V3 (SSLv3) when interoperating with V3.5.x for secure connections. You cannot use Common Secure Interoperability V2 (CSIv2) for interoperability, because Versions 3.5.x and 4.0.x do not support CSIv2.

What to do next

This information is dynamic and might be augmented by information in Technical Articles that are available on the WebSphere Developer Domain. Be sure to check the site for the latest information. You can find a link to the site in Installation: Resources for Learning.

Preparing to install and configure a Web server

Install the IBM HTTP Server or another supported Web server, or install a plug-in for an existing Web server to enable it to work with the WebSphere Application Server, V5 product.

To use a Web server other than IBM HTTP Server V1.3.26.x on the product CD, install and configure the Web server before installing the WebSphere Application Server product. To install and configure IBM HTTP Server V2.0, use the installation and configuration instructions in Installing IBM HTTP Server powered by Apache 2.0.

The WebSphere Application Server installation wizard configures supported Web servers. You can also manually configure supported Web servers for WebSphere Application Server, V5, as described in Manually configuring supported Web servers. If you must migrate a Web server from supporting an earlier version of WebSphere Application Server, to supporting V5, see Migrating Web server configurations.

Web server configuration

To install the Web server and Application Server on the same machine, select the appropriate Web server plug-in for a supported Web server. There is no further migration. This scenario is described in step 1 below. Other scenarios might require you to perform some Web server migration.

When you install WebSphere Application Server, you can indicate which Web server supports each Application Server. The installation wizard uses three files to configure a plug-in for the Web server you select:

- The **Web server configuration file** on the Web server, such as the `httpd.conf` file for IBM HTTP Server.
- The **plug-in configuration file** on the Application Server, `plugin-cfg.xml`, either on the Application Server in a single machine configuration, or copied to the Web server from the Application Server, either during installation, or manually. This file must refresh on the Web server whenever an Application Server configuration change requires it to regenerate.
- The **binary plug-in program** file, either on the Application Server in a single machine configuration, or copied to the Web server from the Application Server, either during installation, or manually. An example of such a file is the `mod_ibm_app_server_http.dll` file for IBM HTTP Server on a Windows platform.

If you install a Web server plug-in, the installation wizard updates the configuration file on the Web server you select to define where the Web server can locate the plug-in binary file and the `plugin-cfg.xml` configuration file.

WebSphere Application Server provides a unique binary plug-in file for each supported Web server. Plug-in binaries for supported servers are located in the `bin` folder of the installation root. The plug-in configuration file is in the `config/cells` folder of the installation root.

The binary file reads the `plugin-cfg.xml` configuration file (generated by the Application Server), to retrieve the configuration of the Application Server. An example of a specification in the configuration file is how often the binary file reads the XML configuration file to determine if there are changes in the file. The XML file also identifies cluster members and applications on the Application Server.

The Web server configuration file is installed as part of whatever Web server you install. It must have entries to identify the locations of the WebSphere Application Server binary plug-in file and the `plugin-cfg.xml` configuration file, for every Application Server that hosts applications that the Web server accesses. The installation wizard configures the file if you install the plug-in on the Web server machine. You have to remove the old entries manually. You can also add the current entries manually, if necessary.

Procedure for installing and configuring a Web server

Steps for this task

1. **(Optional)** Install the IBM HTTP Server V1.3.26.x and its plug-in from the product CD-ROM.

   If you select the IBM HTTP Server plug-in during installation of the base WebSphere Application Server product, the installation wizard installs both the IBM HTTP Server, V1.3.26.x plug-in, and the plug-in for IBM HTTP Server V2.0. However, the installation wizard configures a V1.3.26.x installation only. To enable WebSphere Application Server, V5 to work with IBM HTTP Server 2.0, you must manually edit the IBM HTTP Server 2.0 `httpd.conf` file, as described in Installing IBM HTTP Server powered by Apache 2.0.

   Install the IBM HTTP Server and its plug-in on a different machine from the WebSphere Application Server by following this procedure:

   a. Insert the product CD-ROM labeled, **WebSphere Application Server, IBM HTTP Server**, into the machine.

   b. Select **Install the product** if the LaunchPad starts.

   c. Accept the product license agreement.

   d. Select **Custom** installation.

   e. Deselect all options but the IBM HTTP Server and the plug-in for the IBM HTTP Server.

   f. Complete the installation.

   Installing HTTP Server without the WebSphere Application Server

   Alternatively, you can install the IBM HTTP Server without using the WebSphere Application Server installation wizard. You do not get the binary plug-in for using the HTTP Server product with the WebSphere Application Server product:

   a. Verify you have the IBM Developer Kit, Java Technology Edition Version 1.3, installed on your machine.

      Install the IBM Developer Kit for the Java platform, V1.3, if you plan to use the Key Management (IKEYMAN) utility to create server certificates for Secure Sockets Layer (SSL). This Developer Kit ships on the WebSphere Application Server CD-ROM and is available from the following IBM Web site.

   b. Insert the product CD-ROM labeled, **WebSphere Application Server, IBM HTTP Server**, into the machine.

   c. Close the LaunchPad if it starts automatically.

   d. Change directory to the `IHS` directory on the product CD-ROM.

   e. Click the installation script for your platform, to install the IBM HTTP Server:

      - **Windows platforms:** InstallIHS.bat
      - **UNIX-based operating platforms:** InstallIHS.sh

      This script installs the plug-in you need and makes the necessary configuration changes for the supported Web server.

You can use the WebSphere Application Server update installer program to apply fixes to the IBM HTTP Server 1.3.26.x product if you install the IBM HTTP Server with or without using the WebSphere Application Server installer. Installing the HTTP Server product using the WebSphere Application Server installation wizard also lets you use the WebSphere Application Server update installer in the future to apply interim fixes and fix packs to the binary plug-in.

IBM HTTP Server, V1.3.26.x can coexist with earlier versions, or can upgrade V1.3.19 and later, to V1.3.26.x. Upgrading relieves you from having to uninstall and reinstall HTTP Server. Install V1.3.26.x into the same directory structure as V1.3.19 and later, to upgrade the earlier version. If you install HTTP Server into a different directory, V1.3.26.x coexists with the previous version. By default, the IBM HTTP Administration Server and the IBM HTTP Server use the same ports as the previous version, although you can change the port assignments on the coexistence panel during the installation of the WebSphere Application Server product or by editing the `httpd.conf` file:

- Change the port number assignments for the new installation if you install into a separate directory. You can do this on the coexistence panel during the installation of the WebSphere Application Server product. You can back track through the installation wizard and change the port settings if you have not already done so. Or, you can change the port settings after installation, in the `conf\httpd.conf` file in the HTTP Server directory.

- Update the IBM HTTP Server `httpd.conf` configuration entries to remove entries for earlier WebSphere Application Server versions if you install into the same directory as an earlier version.

   Both V4 and V5 of the WebSphere Application Server use the same HTTP Transport plug-in binary module. If the Web server configuration file contains WebSphere Application Server V4.0 plug-in information, you must manually remove it. Otherwise, when HTTP Server attempts to start the second (V5) plug-in binary module, there is an error. The error indicates that the module is already loaded.

   The configuration file might contain duplicate entries for accessing WebSphere Application Server Samples as shown in the following example. Remove any aliases for previous versions and retain the V5 entries:

```
Version 3.5.x installations
Alias /IBMWebAS/  "C:\WebSphere\AppServer35\web\"

Version 4.0.x installations:
Alias /IBMWebAS/  "C:\WebSphere\AppServer40\web\"
Alias /WSsamples  "C:\WebSphere\AppServer40\WSsamples\"

Version 5 installation:
Alias /WSsamples "c:\Program Files\WebSphere\AppServer\WSsamples"
Alias /IBMWebAS/ "c:\Program Files\WebSphere\AppServer\web\"
```

   If you get an `/opt/IBMHttpServer/bin/httpd: error while loading shared libraries:libdb.so.2:cannot open shared object file:no such file or directory ./apachect1 sart: httpd could not be started` error when starting IBM HTTP Server Version 1.3 or Version 2.0 on RedHat Version 8, install the **compat-db rpm** from the RedHat CD. This lets IBM HTTP Server start successfully.

2. **(Optional)** Install the IBM HTTP Server powered by Apache 2.0.

   If you get an `/opt/IBMHttpServer/bin/httpd:error while loading shared libraries:libdb.so.2:cannot open shared object file:no such file or directory ./apachect1 sart: httpd could not be started` error when starting IBM HTTP Server Version 1.3 or Version 2.0 on Red Hat Version 8, install the **compat-db rpm** from the Red Hat CD. This will enable IBM HTTP Server to start successfully.

3. **(Optional)** Install the plug-in for another supported Web server from the product CD-ROM.

   Select the plug-in for a supported Web server on the feature selection page of the installation wizard.

4. **(Optional)** Manually configure the plug-in for supported Web servers.

5. **(Optional)** Migrate plug-ins to work with WebSphere Application Server, Version 5.

Migrating plug-ins to support multiple versions of WebSphere Application Server is described in Migrating and coexisting.

6. **(Optional)** Install more than one instance of the IBM HTTP Server on an operating system image.

   If you install multiple instances of the WebSphere Application Server product, you can also install multiple images of the IBM HTTP Server, V1.3.26.x:

   a. Use the **InstallIHS** command script in the `IHS` directory on the product CD to install IBM HTTP Server, V1.3.26.x for a second or later instance.

   b. Install the multiple installation instance to a new directory.

   c. Set a unique port for each multiple instance by editing the ″port″ directive in the `/conf/httpd.conf` file in the IBM HTTP Server installation root.

   d. Apply any service to each multiple instance of the IBM HTTP Server using the silent interface of the WebSphere Application Server update installer:

   ```
   updateSilent -ihsOnly -install -ihsInstallDir dir
   ```

   e. Copy the current binary plug-in from the `install_root\bin` directory of the WebSphere Application Server product, to the HTTP Server machine and update the `LoadModule` `ibm_app_server_http_module` directive in the `httpd.conf` file accordingly.

   f. Copy the `plugin-cfg.xml` file from the `install_root\config\cells` directory of the WebSphere Application Server product, to the HTTP Server machine and update the WebSpherePluginConfig directive in the `httpd.conf` file accordingly.

   On some installations of the Sun One (iPlanet) Web server on an HP-UX machine, it is necessary to manually set the `SHLIB_PATH` variable to `/usr/lib` before starting the Sun One Web server with a plug-in configured for SSL. For example, in the korn shell, issue the following command before invoking the command to start the Sun One Web server:

   ```
   'export SHLIB_PATH=/usr/lib:$SHLIB_PATH'
   ```

   On some installations of Apache on a Linux machine, it is necessary to manually set the `LD_LIBRARY_PATH` variable to `/usr/lib` before starting Apache with a plug-in configured for SSL. For example, in the korn shell, issue the following command before invoking the command to start Apache:

   ```
   'export $LD_LIBRARY_PATH=/usr/lib:$LD_LIBRARY_PATH'
   ```

Installing IBM HTTP Server powered by Apache 2.0

To install and configure IBM HTTP Server V2.0, use the installation and configuration instructions in this topic. Select any Web server plug-in when you install the base WebSphere Application Server product. The installation wizard installs all plug-ins if one is selected. The plug-in for IBM HTTP Server 2.0, `mod_was_ap20_http.so` (and `mod_was_ap20_http.dll` for Windows platforms), is among the plug-ins that the installation wizard installs.

If you select the HTTP Server feature or plug-in during installation of the base IBM WebSphere Application Server, V5 product, the installation wizard installs both the HTTP Server, V1.3.26.x plug-in, and the plug-in for HTTP Server V2.0. However, the installation wizard configures a V1.3.26.x installation only. To enable WebSphere Application Server, V5 to work with HTTP Server V2.0, you must manually edit the HTTP Server V2.0 `httpd.conf` file as described in this topic.

HTTP Server V2.0 is available on all supported WebSphere Application Server V5 platforms. Based on the new Apache V2.0 Web server, this version of IBM HTTP Server includes:

- An Apache V2.0 foundation
- IBM-added InstallShield for MultiPlatforms installation
- Secure Sockets Layer (SSL)
- Lightweight Directory Access Protocol (LDAP)

The HTTP Server V2.0 product is fully supported by IBM when deployed with WebSphere Application Server V5. You can download HTTP Server V2.0 from http://www-3.ibm.com/software/webservers/httpservers/

Use this procedure to perform the following tasks:
- Install HTTP Server V2.0
- Set up HTTP Server V2.0 with WebSphere Application Server Version
- Install WebSphere Application Server on a different machine from HTTP Server V2.0 (remote configuration)

<u>Steps for this task</u>

1. Install HTTP Server V2.0.

   Go to the directory where you downloaded and unpacked the file and issue this command:

   `java -jar setup.jar`

   To install HTTP Server V2.0 on UNIX-based platforms without using an X-Windows GUI:
   - To use all installation defaults instead of dialogs:

     `java -jar setup.jar -silent`
   - To use text-based dialogs:

     `java -jar setup.jar -console`

2. Set up HTTP Server V2.0 to support WebSphere Application Server, V5.

   To use HTTP Server V2.0 with WebSphere Application Server, locate the HTTP Server V2.0 `httpd.conf` file (in the `conf` directory under the HTTP Server installation root directory) and add the following directives to allow the HTTP transport plug-in to work correctly with HTTP Server V2.0:

   ```
   LoadModule   was_ap20_module   /full/path/to/binary_module
   WebSpherePluginConfig   /full/path/to/plugin-cfg.xml_file
   ```

   When WebSphere Application Server installs on the same machine as HTTP Server V2.0, file paths to the binary module and plug-in configuration module are as follows:
   - **Paths to the binary module** (`mod_was_ap20_http.dll` or `mod_was_ap20_http.so`):
     - Windows platforms: `install_root\bin`
     - Linux and UNIX-based platforms: `install_root/bin`
   - **Paths to the WebSphere Application Server, V5 plug-in configuration file** (`plugin-cfg.xml`):
     - Windows platforms: `install_root\config\cells`
     - Linux and UNIX-based platforms: `install_root/config/cells`

3. **(Optional)** Configure IBM HTTP Server V2.0 to work with a remote installation of IBM WebSphere Application Server, V5 (on a different machine).

   Use the following procedure:

   a. Copy the Application Server plug-in for HTTP Server V2.0 from the `install_root/bin` directory of the Application Server to the machine where you installed HTTP Server V2.0.

   If HTTP Server is installed on a machine without WebSphere Application Server V5, you must get the plug-in from a machine where WebSphere Application Server is installed. The Application Server installation program installs the plug-in for HTTP Server V2.0 in the `install_root/bin` directory of the Application Server. The plug-in file name is `mod_was_ap20_http.so` (for UNIX and Linux)and `mod_was_ap20_http.dll` (for Windows). You must copy the plug-in to the machine where you installed HTTP Server V2.0. Put the plug-in into the HTTP Server module directory:
   - Windows platforms: `drive\Program Files\IBM HTTP Server 2.0\modules`
   - AIX: `/usr/IBMIHS/modules`
   - Linux and other UNIX-based platforms: `/opt/IBMIHS/modules`

You must also copy the Application Server `plugin-cfg.xml` file to the machine where you installed HTTP Server. Put the Application Server `plugin-cfg.xml` file in the directory with the HTTP Server configuration files. Suggested paths to the Application Server `plugin-cfg.xml` configuration file are:

- Windows platforms: `drive\Program Files\IBM HTTP Server 2.0\conf`
- AIX: `/usr/IBMIHS/conf`
- Linux and other UNIX-based platforms: `/opt/IBMIHS/conf`

b. Edit the `plugin-cfg.xml` file to point to valid paths on the machine.

After copying the `plugin-cfg.xml` file to the HTTP Server machine, you must edit the file to point to valid paths on the machine. For example:

```
Log LogLevel="Error"
    Name="/opt/WebSphere/AppServer/logs/http_plugin.log"/
```

The path `/opt/WebSphere/AppServer/logs/http_plugin.log` must be a valid path on the HTTP Server machine.

c. **(Optional)** Prepare the installation for Secure Socket Layer (SSL) operation.

If secure socket layer (SSL) is set up on WebSphere Application Server, V5, you must:

1) Copy the `plugin-key.kdb` and `plugin-key.sth` files from the IBM WebSphere Application Server, V5 machine, to the IBM HTTP Server V2.0 machine.

2) Edit the `plugin-cfg.xml` file to point to the appropriate paths on the HTTP Server machine. For example:

```
Property name="keyring" value="/opt/WebSphere/AppServer/etc/plugin-key.kdb"/
Property name="stashfile" value="/opt/WebSphere/AppServer/etc/plugin-key.sth"/
```

The `/opt/WebSphere/AppServer/etc/plugin-key.kdb` and `/opt/WebSphere/AppServer/etc/plugin-key.sth` file specifications must be valid paths on the HTTP Server machine.

Manually configuring supported Web servers

This topic describes manual configuration settings for:

- Apache HTTP Server V1.3
- Domino Web server V6
- Domino Web server V5
- IBM HTTP Server powered by Apache 2.0
- IBM HTTP Server powered by Apache 1.3
- Internet Service Manager Web server
- Sun ONE (iPlanet) Web server

Each subsection is presented as an optional step in the overall procedure, as if you might have more than one Web server that you must configure.

Steps for this task

1. **(Optional)** Configure Apache HTTP Server 1.3 `httpd.conf` file entries.

Use the following entries as a model for configuring your file.

- **UNIX-based platform:**

```
LoadModule app_server_http_module/opt/WebSphere/AppServer/bin/mod_app_server_http.so
WebSpherePluginConfig /opt/WebSphere/AppServer/config/cells/plugin-cfg.xml
```

Most Apache Web servers are not compiled with extended API (EAPI) support. If you see a message similar to the following, when starting the Web server with the `mod_app_server_http.so` plug-in module, use the EAPI version of the module.

```
[Tue Jul  2 13:20:14 2002] [warn] Loaded DSO /opt/WebSphere/AppServer/bin/mod_app_server_http.so
uses plain Apache 1.3 API, this module might crash under EAPI!
(please recompile it with -D EAPI)
```

To use the EAPI module, configure the `httpd.conf` file to point to `mod_app_server_http_eapi.so` instead of to `mod_app_server_http.so`:

```
LoadModule app_server_http_module/opt/WebSphere/AppServer/bin/mod_app_server_http_eapi.so
WebSpherePluginConfig /opt/WebSphere/AppServer/config/cells/plugin-cfg.xml
```

> **Note:** The installation root can vary per operating system platform. For example, the AIX installation root is `/usr/WebSphere/AppServer/` for V5.

- **Windows platform:**

```
LoadModule ibm_app_server_http_module <drive>:\WebSphere\AppServer\bin\mod_app_server_http.dll
WebSpherePluginConfig <drive>:\WebSphere\AppServer\config\cells\plugin-cfg.xml
```

Most Apache Web servers are not compiled with extended API (EAPI) support. If you see a message similar to the following, when starting the Web server with the `mod_app_server_http.dll` plug-in module, use the EAPI version of the module.

```
[Tue Jul  2 13:20:14 2002] [warn] Loaded DSO C:\WebSphere\AppServer\bin\mod_app_server_http.so
uses plain Apache 1.3 API, this module might crash under EAPI!
(please recompile it with -D EAPI)
```

To use the EAPI module, configure the `httpd.conf` file to point to `mod_app_server_http_eapi.dll` instead of to `mod_app_server_http.dll`:

```
LoadModule ibm_app_server_http_module <drive>:\WebSphere\AppServer\bin\mod_app_server_http_eapi.dll
WebSpherePluginConfig <drive>:\WebSphere\AppServer\config\cells\plugin-cfg.xml
```

2. **(Optional)** Configure Domino Web Server V6 entries.

   Follow this procedure to enable the HTTP transport plug-in to work with Domino Version 6.02. The installation wizard does not perform this modification. Perform this manual procedure.

   a. Start the Domino server.

   b. Access the file `/names.nsf` using your Web browser (for example, `http://hokie2ks.raleigh.ibm.com/names.nsf`). The browser prompts you for a password. Give the administrator name and password.

   c. Click **Configuration** on the left side of the page.

   d. Click **Servers** on the top-left center of the page.

   e. Click **All Server Documents** on the left side of the page.

   f. Click the server that you intend to have work with WebSphere Application Server, V5.

   g. Click **Edit Server** on the top-left of the center window.

   h. Click **Internet Protocols** in the middle of the page.

   i. Under **DSAPI Section** in the middle-right of the page, and add the path to the Domino plug-in. The plug-in is installed in the `bin` directory of the WebSphere Application Server installation root.

      If there are already Domino Web Server Application Programming Interface (DSAPI) filter files specified, use a space to delimit the WebSphere Application Server plug-in file.

   j. Click **Save and Close** on the upper-left of the center window.

   k. Define the location of the `plugin-cfg.xml` configuration file.

      - UNIX-based platform:

        Set the `WAS_HOME` environment variable to point to the WebSphere Application Server installation root directory.

      - Windows platform:

        Add the variable **Plugin Config** to the registry under the path **HKEY_LOCAL_MACHINE > SOFTWARE > IBM > WebSphere Application Server > 5.0.0.0**. Set the value for this variable to the location of the `plugin-cfg.xml` file, which is `install_root\config\cells\plugin-cfg.xml`.

l. Restart the Domino server. When the server starts, information similar to the following appears:

```
02/12/2001 03:05:09 PM  JVM: Java virtual machine initialized
WebSphere Application Server DSAPI filter loaded
02/12/2001 03:05:10 PM  HTTP Web Server started
```

3. **(Optional)** Configure Domino Web server entries.

   Follow this procedure to enable the HTTP transport plug-in to work with Domino V5.05 or 5.06. The installation wizard does not perform this modification. Perform this manual procedure.

   a. Start the Domino server.

   b. Access the file `/webadmin.nsf` using your Web browser (for example, `http://hokie2ks.raleigh.ibm.com/webadmin.nsf`). The browser prompts you for a password. Give the short name for the administrator and the administrator password.

   c. Click **Configuration** on the left side of the page.

   d. Click **Servers** on the top-left center of the page.

   e. Double-click the server that you intend to have work with WebSphere Application Server, V5.

   f. Click **Edit Server** on the top-left of the center window.

   g. Click **Internet Protocols** in the middle of the page.

   h. Under **DSAPI** in the middle-right of the page, add the path to the Domino plug-in, which is installed in the WebSphere Application Server `bin` directory.

      Note: If there are already Domino Web Server Application Programming Interface (DSAPI) filter files specified, use a space to delimit the WebSphere Application Server plug-in file.

   i. Click **Save** and **Close** on the upper-left of the center window.

   j. Define the location of the `plugin-cfg.xml` configuration file.

      • UNIX-based platform:

        Set the `WAS_HOME` environment variable to point to the WebSphere Application Server installation root directory.

      • Windows platform:

        Add the variable **Plugin Config** to the registry under the path **HKEY_LOCAL_MACHINE > SOFTWARE > IBM > WebSphere Application Server > 5.0.0.0**. Set the value for this variable to the location of the `plugin-cfg.xml` file, which is `install_root\config\cells\plugin-cfg.xml`.

   k. Restart the Domino server. When the server starts, information similar to the following appears:

```
02/12/2001 03:05:09 PM  JVM: Java virtual machine initialized
WebSphere Application Server DSAPI filter loaded
02/12/2001 03:05:10 PM  HTTP Web Server started
```

4. **(Optional)** Configure IBM HTTP Server 2.0 `httpd.conf` file entries.

   Use the following entries as a model for configuring your file.

   • **UNIX-based platform:**

```
LoadModule was_ap20_module/opt/WebSphere/AppServer/bin/mod_was_ap20_http.so
WebSpherePluginConfig /opt/WebSphere/AppServer/config/cells/plugin-cfg.xml
```

   Note: The installation root can vary per operating system platform. For example, the AIX installation root is `/usr/WebSphere/AppServer/` for V5.

   • **Windows platform:**

```
LoadModule was_ap20_module <drive>:\WebSphere\AppServer\bin\mod_was_ap20_http.dll
WebSpherePluginConfig <drive>:\WebSphere\AppServer\config\cells\plugin-cfg.xml
```

   Note: If the IBM HTTP Server 1.3.26.x directive, LoadModule ibm_app_server_http_module, is present in an IBM HTTP Server 2.0 `httpd.conf` file, the IBM HTTP Server cannot start. You must comment or delete the directive, to start the V2 server.

5. **(Optional)** Configure IBM HTTP Server 1.3.26.x `httpd.conf` file entries.

   Use the following entries as a model for configuring your file.

- **UNIX-based platform:**

```
LoadModule ibm_app_server_http_module/opt/WebSphere/AppServer/bin/mod_ibm_app_server_http.so
WebSpherePluginConfig /opt/WebSphere/AppServer/config/cells/plugin-cfg.xml
```

> **Note:** The installation root can vary per operating system platform. For example, the AIX installation root is `/usr/WebSphere/AppServer/` for V5.

- **Windows platform:**

```
LoadModule ibm_app_server_http_module <drive>:\WebSphere\AppServer\bin\mod_ibm_app_server_http.dll
WebSpherePluginConfig <drive>:\WebSphere\AppServer\config\cells\plugin-cfg.xml
```

6. **(Optional)** Configure Internet Service Manager Web server entries.

    Follow this procedure to manually reproduce how the installation wizard configures the Internet Service Manager (IIS) Web server:

    a. Start the IIS application.

    b. Create a new virtual directory for the Web site instance that you intend to work with WebSphere Application Server. To create this directory with a default installation, expand the tree on the left until you see **Default Web Site**. Right-click **Default Web Site** and select **New > Virtual Directory**. In the wizard for adding a virtual directory, do the following:

        1) Type **sePlugins** in the **Alias to be used to Access Virtual Directory** field.

        2) Browse to the WebSphere Application Server `bin` directory in the **Enter the physical path of the directory containing the content you want to publish** field.

        3) Select the **Allow Execute Access** check box in the **What access permissions do you want to set for this directory** field.

        4) Click **Finish** to add the `sePlugins` virtual directory to your default Web site.

    c. Add the Internet Services Application Programming Interface (ISAPI) filter into the IIS configuration. Right-click on the host name in the tree on the left and click **Properties**. In the Properties dialog, do the following:

        1) Go to the **Internet Information Services** tab.

        2) Click **WWW Service** in the **Master Properties** window.

        3) Click **Edit** to open the **WWW Service Master Properties** window.

        4) Click **ISAPI Filters > Add** to open the **Filter Properties** window.

        5) Type iisWASPlugin in the **Filter Name** field.

        6) Click **Browse** in the **Executable** field.

        7) Browse to the `bin` directory of the WebSphere Application Server installation root.

        8) Click the `iisWASPlugin_http.dll` file.

        9) Click **OK** until all open windows close.

    d. Add the variable `Plugin Config` to the registry under the path **HKEY_LOCAL_MACHINE > SOFTWARE > IBM > WebSphere Application Server > 5.0.0.0**. Set the value to the location of the configuration file, which is `install_root\config\cells\plugin-cfg.xml`.

7. **(Optional)** Configure Sun ONEt (iPlane) Web server entries.

    Use the following entries as a model for configuring your file.

    - **UNIX-based platform for Sun ONE 4.1:**

        Include these directives in the `obj.conf` file:

```
Init fn="load-modules" funcs="as_init,as_handler,as_term" shlib="
                            /opt/WebSphere/AppServer/bin/libns41_http.so"
Init fn="as_init" bootstrap.properties=" /opt/WebSphere/AppServer/config/cells/plugin-cfg.xml"
Service fn="as_handler"
```

    (The first line of the example is split to fit on the page. Type it as one line.)

Note: The installation root can vary per operating system platform. For example, the AIX installation root is `/usr/WebSphere/AppServer/` for V5.

- **UNIX-based platform for Sun ONE 6.0:**

  Add one directive to the `obj.conf` file after ″<Object name=default>″:

  ```
  Service fn="as_handler"
  ```

  Add two directives to the end of the `magus.conf` file:

  ```
  Init fn="load-modules" funcs="as_init,as_handler,as_term" shlib="
                          /opt/WebSphere/AppServer/bin/libns41_http.so"
  Init fn="as_init" bootstrap.properties="/opt/WebSphere/AppServer/config/cells/plugin-cfg.xml"
  ```

  (The first line of the example is split to fit on the page. Type it as one line.)

  Note: The installation root can vary per operating system platform. For example, the AIX installation root is `/usr/WebSphere/AppServer/` for V5.

- **Windows platform for Sun ONE 4.1:**

  Include these directives in the `obj.conf` file:

  ```
  Init fn="load-modules" funcs="as_init,as_handler,as_term" shlib="
              <drive>:\WebSphere\AppServer\bin\libns41_http.dll"
  Init fn="as_init" bootstrap.properties=" <drive>:\WebSphere\AppServer\config\cells\plugin-cfg.xml"
  Service fn="as_handler"
  ```

  (The first line of the example is split to fit on the page. Type it as one line.)

- **Windows platform for Sun ONE 6.0:**

  Add one directive to the `obj.conf` file after ″<Object name=default>″:

  ```
  Service fn="as_handler"
  ```

  Add two directives to the end of the `magus.conf` file:

  ```
  Init fn="load-modules" funcs="as_init,as_handler,as_term" shlib="
              <drive>:\WebSphere\AppServer\bin\libns41_http.dll"
  Init fn="as_init" bootstrap.properties=" <drive>:\WebSphere\AppServer\config\cells\plugin-cfg.xml"
  ```

  (The first line of the example is split to fit on the page. Type it as one line.)

Installing the product

This topic describes how to install the WebSphere Application Server product. It provides enough detail to guide you through preparing for, choosing, and installing the variety of options and features provided. To prepare for installation and to make yourself familiar with installation options, read through this topic and its related topics, before you start to use the installation tools. In particular, read these topics before installing the product:

- Platform-specific tips for installing and migrating
- Tips for installing the embedded messaging feature
- Migrating and coexisting
- Using the LaunchPad to start the installation
- Installing with the installation wizard GUI
- Installing silently

As a general rule, if you launch an installation and there is a problem such as not having enough temporary space or not having the right packages on your Linux or UNIX-based system, for example, you must cancel the installation, make the required changes, and restart the installation to pick up changes you made.

This topic describes installing the base WebSphere Application Server product, using the installation image on the product CD-ROM labeled, **Application Server, IBM HTTP Server**. This CD-ROM is available in the base WebSphere Application Server product package, the Network Deployment product package, and the Enterprise product package. Use this topic to install the base WebSphere Application Server installation image, regardless of which package the CD-ROM comes from, unless you are doing an umbrella installation while installing the Enterprise product.

Open this topic in the InfoCenter for the Network Deployment to learn how to install the Network Deployment installation image and establish a multimachine environment. The CD-ROM for the Network Deployment installation image is labeled, **Deployment Manager**.

Open this topic in the InfoCenter for the Enterprise product to learn how to install the Enterprise installation image and extend the multimachine environment. The CD-ROM for the Enterprise installation image is labeled, **Enterprise Application Server**.

In addition to being available in this Adobe PDF format on the product CD-ROM, this topic is also available in the online InfoCenter. When possible, access the most current version of this information by selecting the InfoCenter version. The official version of the InfoCenter is in English, but it is also available in other national language versions.

Although it is not supported or recommended, you can install this product as a non-root user on a UNIX-based operating system, or from a user ID that is not part of the Administrator group on a Windows platform. However, there are certain components, such as the embedded messaging feature, that require you to install as root or as part of the Administrator group.

The LaunchPad tool lets you access the product overview, `readme.html` file, and installation guides. After starting the LaunchPad, click its **Install the product** option to walk through the installation wizard.

The installation wizard:
- Automatically checks prerequisites
- Looks for a previous WebSphere Application Server installation, to determine whether to display the migration and coexistence panel during the installation
- Installs the IBM HTTP Server and other features, if you select them

This section contains:
- Platform-specific tips for installing and migrating
- Tips for installing the embedded messaging feature
- Using the LaunchPad to start the installation
- Installing with the installation wizard GUI
- Installing silently
  - Customizing the options response file
  - responsefile

Platform-specific tips for installing and migrating

This topic is a collection of platform-specific tips that can help you install and migrate the base WebSphere Application Server product.

As a general rule, if you launch an installation and there is a problem such as not having enough temporary space or not having the right packages on your Linux or UNIX-based system, for example, you must cancel the installation, make the required changes, and restart the installation to pick up the changes you made.

For information related to installing the embedded messaging feature, refer to ″Tips for installing the embedded messaging feature″.

This topic is divided into sections to make it easier for you to find applicable tips:
*   All platforms
*   All Linux and UNIX-based platforms
*   AIX platforms
*   **5.0.1** HP-UX platforms
*   Linux platforms
*   Solaris platforms
*   Windows platforms

All platforms
Summary of tips that apply to all platforms
*   Restart the server after a configuration change
*   Port updates for co-existence require a WebSphere Application Server installation
*   Manually uninstall all Beta products before installing WebSphere Application Server V5.
*   Installing from a directory with a name beginning with the word disk fails.
*   Migration tools are in the migration subdirectory on the WebSphere Application Server, V5 CD-ROM.
*   License files can contain bad characters in certain languages.
*   Update the XMLConfig utility on WebSphere V4.0 Advanced Edition before migration.
*   The WebSphere Application Server Network Deployment product does not support the launchClient command.
*   **5.0.1** Save the jdbc-resource-provider-templates.xml file before you install Fix Pack 1 for WebSphere Application Server
*   **5.0.1** Uninstall fixes for IBM HTTP Server and MQ Series (Embedded Messaging) before installing fix pack updates to these features

Tips that apply to all platforms
*   **Restart the server after a configuration change**

    If you make any changes to the configuration, restart the server as noted in the messages section of the administrative console.
*   **Port updates for co-existence require a WebSphere Application Server installation**

    Port updates for co-existence require the installation of WebSphere Application Server. This requirement affects port updates for IBM HTTP Server co-existence. Port updates do not occur if only the IBM HTTP Server is installed. In this case, manually update the httpd.conf files.
*   **Note to Beta participants: Manually uninstall all Beta products before installing WebSphere Application Server V5.**

    You might experience problems if portions of the beta product remain.

    To make sure you get a clean uninstall, follow these steps:
    1. Uninstall the beta version of WebSphere Application Server.

        Refer to ″Uninstalling WebSphere Application Server″ for more information.
    2. Confirm that WebSphere MQ and WebSphere Embedded Messaging Publish and Subscribe (WEMPS) have been uninstalled.

If not, uninstall them by using the Windows Add/Remove programs tool or Smitty/installp, whichever is appropriate. Refer to "Uninstalling WebSphere Application Server" for more information about manually uninstalling on specific platforms.

3. Remove all previous WebSphere Application Server install directory trees from your system.

If the stand-alone WebSphere MQ V5.3 is installed, remove the WEMPS directory only. Do not uninstall or remove other WebSphere MQ V5.3 items.

- **Installing from a directory with a name beginning with the word `disk` fails.**

  Installing WebSphere Application Server V5 from a folder that begins with the word `disk` results in an error. Provide another name for the folder.

- **Migration tools are in the migration subdirectory on the WebSphere Application Server, V5 CD-ROM.**

  A `migration` subdirectory on the CD-ROM installation image contains the WASPreUpgrade command line migration tool. It is intended for scenarios where you might save the currently installed configuration manually before installing the V5 product. One example of this situation is where you must upgrade the operating system as part of the V5 installation. You could migrate the earlier version, copy the migrated files in the backup directory to another system, update the operating system, restore the migrated files in their backup directory, install V5, and complete the migration.

  You can also use the migration directory on the CD-ROM to back up a V5 configuration in the event of an operating system upgrade. After the upgrade, you could restore the V5 configuration using the WASPostUpgrade tool.

- **License files can contain bad characters in certain languages.**

  Use the graphical installation interface to avoid this problem.

- **Update the XMLConfig utility on WebSphere V4.0 Advanced Edition before migration.**

  The migration tools use the XMLConfig utility to export the WebSphere 4.0 Advanced Edition configuration. There are some fixes available to the WebSphere 4.0 Advanced Edition XMLConfig utility that you can apply to fix the following problems:

  – PQ52555 - XMLConfig doesn't export clone property configuration
  – PQ55064 - XMLConfig does not export EJB to DataSource level mappings
  – PQ58038 - Performing an XMLConfig export produces an extra CRLF
  – PQ62103 - XMLConfig full export fails with NullPointerException in a Multinode Environment
  – PQ62471 - Security AdminRoles are not getting exported during XML export
  – PQ63815 - "=" is not a valid character for the value string in XMLConfig

- **The WebSphere Application Server Network Deployment product does not support the launchClient command.**

  The **launchClient** command works on a base WebSphere Application Server node and on a WebSphere Application Server Enterprise node, where Enterprise extends the base product.

- **5.0.1** **Save the jdbc-resource-provider-templates.xml file before you install Fix Pack 1 for WebSphere Application Server**

  In WebSphere Application Server, Version 5.0.1, the jdbc-resource-provider-templates.xml file is modified to include a new DB2 Universal JDBC provider and other JDBC provider name changes. If you have modified this configuration file in WebSphere Application Server, Version 5, save the jdbc-resource-provider-templates.xml file before installing the WebSphere Application Server, Version 5.0.1. Otherwise, the new XML file overwrites the jdbc-resource-provider-templates.xml file and your configuration data is lost.

- **5.0.1** **Uninstall fixes for IBM HTTP Server and MQ Series (Embedded Messaging) before installing fix pack updates to these features**

  If you have installed fixes for IBM HTTP Server or MQ Series Embedded Messaging, you need to uninstall these fixes prior to updating these features using the Update Installer. If you do not uninstall

these fixes, the updates to IBM HTTP Server or Embedded Messaging made during the fix pack installation might fail or the installation might be faulty. You can choose to skip the updates IBM HTTP Server or Embedded Messaging if they are not required during fix pack installation. See the Update Installer documentation for more information.

All Linux and UNIX-based platforms
Summary of Linux and UNIX-based platform tips

- Move all core dump files from the var/sadm/pkg directory.
- Start the JMS server before running some Samples on UNIX platforms.
- Use an option on the install or uninstall command to identify a temporary directory other than the default /tmp directory.
- **5.0.1** An Error 500 message displays when you click Name Space Bindings
- **5.0.1** IBM HTTP Server 1.3.26.x requires the compat-db-3.3.11-2.i386.rpm package on Red Hat 8.0

Linux and UNIX-based platform tips

- **Move all core dump files from the var/sadm/pkg directory.**

  The InstallShield for MultiPlatforms (ISMP) installation wizard iterates through all the directories in /var/sadm/pkg, assuming that each entry it finds is a directory. It then tries to open the `pkginfo` file underneath the directory. The ISMP wizard fails when it cannot find an entry under the core file. Remove the core file from the directory to avoid the problem.

- **Start the JMS server before running some Samples on UNIX platforms.**

  To run Samples that use JMS APIs, you must manually start the JMS server (`jmsserver`) before running the samples on UNIX platforms.

  To start the JMS server, complete the following steps:

  1. Open the administrative console: `http://localhost:9090/admin`.
  2. Expand **Servers > Application Servers > server1 > Server Components > JMS Servers**.
  3. Change the `Initial State` from **STOP** to **START** .
  4. Save the configuration and log out of the administrative console.
  5. Stop and restart the Application Server from the command line. For example,

     `stopServer.sh server1 followed by startServer.sh server1`

- **Use an option on the install or uninstall command to identify a temporary directory other than the default /tmp directory.**

  If the tmp disk does not have a large enough allocation, this message appears:

  ```
  Error writing file =  There may not be enough temporary disk space.
  Try using -is:tempdir to use a temporary directory on a partition with more disk space.
  ```

  Use the `-is:tempdir` installation option to specify a different temporary disk. For example, the following command uses the `/swap` file system as a temporary disk during installation:

  `./install -is:tempdir /swap`

- **5.0.1** **An Error 500 message displays when you click Name Space Bindings**

  On UNIX platforms, when you click **Name Space Bindings**, you receive an Error 500 message on the browser. When you click Show Details, a NullPointerException displays. This only occurs with the administrative ID that was configured as the server ID.

  To solve the problem, add a different user to **System Management > Console Users** and login with that user to add additional Name Space Bindings.

- **5.0.1** **IBM HTTP Server 1.3.26.x requires the compat-db-3.3.11-2.i386.rpm package on Red Hat 8.0.**

Install the package on Red Hat 8.0 platforms. Otherwise the HTTP Server cannot start.

AIX platforms
Summary of AIX platform tips

- Space requirements for installation.

- **5.0.2** Package requirements for Fix Pack 2.

- **5.0.1** A core dump might occur when running WebSphere Application Server with DB2 V7.2 FP8 client in AIX 5.2

- Installing or uninstalling causes a problem.

- Avoid a null pointer exception during the interactive installation of the IBM HTTP Server product on AIX platforms.

- Ignore DBCS messages when you do not require DBCS support. Otherwise, install the necessary patches.

- Apply a fix package to AIX 4.3.3 before installing the Simplified Chinese (GBK) version of WebSphere Application Server.

- Use UNIX line-end characters (0x0D0A) to terminate each line of the options response file for silent installation.

- The scroll bar disappears on the installation feature panel.

- Ignore the following error messages that appear in the log.txt installation log.

- Avoid a potential port conflict between the administrative console and the AIX 5.1, with maintenance level 2, WebSM system management console.

- Cancel the installation and update your operating system before restarting the installation, if you install on an AIX machine and receive a message that a file set is missing, such as file set X11.fnt.ucs.ttf.

- Set up the display environment for a real root login on AIX systems.

- Avoid a DSAPI filter-loading error when the Lotus Domino Server starts.

AIX platform tips

- **Space requirements for AIX platforms**

Table 4. Installed sizes of installation directories for AIX platforms

	Base product	Network Deployment product	IBM HTTP Server	Tivoli Global Security Kit (base product only)	Temporary space requirement
Installation directory	`/usr/ WebSphere/ AppServer`	`/usr/ WebSphere/ DeploymentManager`	`/usr/ WebSphere/ DeploymentManager`	`/usr/ ibm/ gsk5`	`/tmp`
Minimum free space for installation	512 MB	512 MB	25.5 MB	32.5 MB	At least 375 MB

Space requirements for the embedded messaging feature are described in "Installing WebSphere embedded messaging as the JMS provider".

- **5.0.2** **AIX V5 package requirements for Fix Pack 2.**
  Fix Pack 2 requires the bos.perf package located on CD1. The package provides these filesets:
  – PerformanceStatistics Library, bos.perf.libperfstat, version 5.1.0.35, which supports various AIX performance metrics
  – Performance Statistics, bos.perf.perfstat, version 5.1.0.36, which supports the libperfstat library

Some PMI functionality is not available if the package is not installed.

- **5.0.1**   **A core dump might occur when running WebSphere Application Server with DB2 V7.2 FP8 client in AIX 5.2**

  A core dump might occur when you are running WebSphere Application Server with DB2 V7.2 FP8 client in AIX 5.2 having all of the following configurations:

  – You installed WebSphere Application Server in 64-bit AIX 5.2.

  – You installed 64-bit DB2 V7.2 FP8.

  – You created a 32-bit client instance to connect to the DB2 server.

  – You installed applications which use DB2 data source in WebSphere Application Server.

  A java core dumped error displays when you start the WebSphere Application Server. This is due to a DB2 problem. A defect has been opened to DB2.

  To solve this problem, rename the `libdb2lai.a` file on your DB2 client machine as follows:

  – Change to the `/usr/lpp/db2_07_01/lib` directory.

  – Rename the file `libdb2lai.a` to `libdb2lai.a.orig`.

- **Installing or uninstalling causes a problem.**

  While installing or uninstalling, a core dump causes the desktop to disappear, leaving only the login prompt. Run the DBX utility on the resulting core file. This problem produce messages similar to the folling example:

  ```
  dbx /usr/bin/X11/X core
  Type 'help' for help.
  reading symbolic information ...warning: no source compiled with -g

  [using memory image in core]

  Segmentation fault in . at 0x10040aac
  0x10040aac (???) 931d0000        stw   r24,0x0(r29)
  (dbx) where
  SetFontPathElements(??, ??, ??, ??) at 0x10040aac
  SetFontPath(??, ??, ??, ??) at 0x10042e58
  SetFontPath(??, ??, ??, ??) at 0x10042e80
  ProcSetFontPath(??) at 0x10015ce4
  DeleteClientFromAnySelections(??) at 0x1001e8a8
  ```

  To solve this problem, apply the AIX PTF fix for PMR:82869,004.

- **Avoid a null pointer exception during the interactive installation of the IBM HTTP Server product on AIX platforms.**

  When installing IBM HTTP Server as a stand-alone GUI install on the AIX platform, in certain cases you might receive a null pointer Java error on the installation panels, when the installer begins to copy files to your machine. This is caused by an unstable AIX ODM registry. To work around this problem:

  1. Exit the installer.

  2. Issue the command to see if there are remaining IBM HTTP Server registry entries that should not be there:

     ```
     lslpp -l|grep IHS
     ```

  3. If there are remaining IBM HTTP Server registry entries, do a silent install of IBM HTTP Server by running:

  ```
  java -jar setup.jar -silent -P ihs.installLocation=the desired install location
  ```

  Note: To remove the installation silently, use the -silent parameter on the uninstall command. For example, run this command from the IBM HTTP Server installation root:

  ```
  java -jar _uninst/uninstall.jar -silent
  ```

- **Ignore DBCS messages when you do not require DBCS support. Otherwise, install the necessary patches.**

  While installing any WebSphere Application Server product, you might see the following messages on AIX 5.1 and AIX 4.3.3 until you install the missing filesets.

  ```
  Operating system patches of particular concern:

  fileset X11.fnt.ucs.ttf_KR was not found on the system.
  v5.1.0.0 - Font required for Korean character display

  fileset X11.fnt.ucs.ttf_TW was not found on the system.
  v5.1.0.0 - Font required for Taiwanese character display

  fileset X11.fnt.ucs.ttf_CN was not found on the system.
  v5.1.0.0 - Font required for Chinese character display
  ```

- **Apply a fix package to AIX 4.3.3 before installing the Simplified Chinese (GBK) version of WebSphere Application Server.** You can download the fix from the AIX Fix Distribution Service Web site at URL: `http://techsupport.services.ibm.com/rs6k/fixdb.html`. Search the AIX V4 database for APAR number IY22531. You can find a link to the site in Installation: Resources for Learning.

- **Use UNIX line-end characters (0x0D0A) to terminate each line of the options response file for silent installation.**

  During a silent installation on AIX machines, the response file passed to the installer program must not contain ASCII line-end characters (0x0D0A). The response file must contain UNIX line-end characters only. When the options response file contains ASCII line-end characters, the **install** command is unsuccessful and does not log or display an error. To verify the cause of failure, use the Java argument `-Dis.debug=1` on the **install** command. The debug information describes a service exception about invalid characters in the options response file.

- **The scroll bar disappears on the installation feature panel.**

  If the scroll bar disappears, use the up and down arrow keys to navigate the features in the Feature panel. Use the tab key to move the focus to the navigation. You can also use the mouse.

- **Ignore the following error messages that appear in the log.txt installation log.**

  Ignore:

  ```
  WASBase, com.ibm.wizard.platform.aix.AixProductServiceImpl,
          wrn,   - WARNING:
  Got invalid size of 0 for file:
          /usr/WebSphere/AppServer/config/cells/BaseApplicationServerCell/nodes/DefaultNode/spi.policy:
  WASBase, com.ibm.wizard.platform.aix.AixRegistryServiceImpl, wrn, AixRegistryServiceImpl:
  Error attempting to modify AIX VPD.
  ```

- **Avoid a potential port conflict between the administrative console and the AIX 5.1, with maintenance level 2, WebSM system management console.**

  The AIX WebSM system management server listens on port 9090 by default. If you suspect you have a port conflict, verify it by shutting down WebSphere Application Server. Then issue this command:

  ```
  netstat -an |grep 9090
  ```

  If you get a match, another process is already listening on port 9090. If you want the WebSM server and WebSphere Application Server to coexist, change the WebSphere Application Server administrative console port when installing WebSphere Application Server, or after installation. Although not recommended, you can also disable the WebSM server. To disable the WebSM server, issue this command:

  ```
  /usr/websm/bin/wsmserver -disable
  ```

  The command permanently disables WebSM server startup.

- **Cancel the installation and update your operating system before restarting the installation, if you install on an AIX machine and receive a message that a file set is missing, such as file set `X11.fnt.ucs.ttf`.**

- **Set up the display environment for a real root login on AIX systems.**

In a normal root login, issue the command **su**. For a real root login, issue the command **su -**.

Display settings for a normal root login are automatic. For a real root login, you must set your display environment properly to successfully view the GUI installation wizard. Otherwise, you see a message about `Preparing Java(tm) Virtual Machine...`, and seven rows of dots, but no installation GUI and no further messages. Refer to the documentation for AIX machines to determine proper display settings.

- **Avoid a DSAPI filter-loading error when the Lotus Domino Server starts.**

  On a UNIX-based operating system, if the Lotus Domino Web server starts using a non-root user, you are likely to generate a DSAPI filter-loading error:

  ```
  Error loading DSAPI filter.

  Filter not loaded: /usr/WebSphere/AppServer/bin/libdomino5_http.a
  ```

  Manually change the WebSphere Application Server `bin` directory permissions from `750` to `755` to run Lotus Domino Server as a non-root user and not generate the error. This change does, however, pose a security risk.

  You must also change permissions on the WebSphere Application Server `logs` directory to `777` to let Lotus Domino Server write to the log.

  If the Lotus Domino Server is started as root, the problem does not occur.

HP-UX platforms
5.0.1

Summary of HP-UX platform tips
- Space requirements for installation.
- Configure HP-UX kernel settings before installing.
- Avoid using NetScape 4.79 on HPUX 11 in Japanese, to avoid problems in viewing the administrative console.
- Configure the converter.properties file to use EUC-JP (Japanese) encoding on HP-UX.

HP-UX platform tips
- **Space requirements for HP-UX platforms**

Table 5. Installed sizes of installation directories for HP-UX platforms

	Base product	Network Deployment product	IBM HTTP Server	Tivoli Global Security Kit (base product only)	Temporary space requirement
Installation directory	/opt/ WebSphere/ AppServer	/opt/ WebSphere/ DeploymentManager	/opt/ WebSphere/ DeploymentManager	/opt/ ibm/ gsk5	/tmp
Minimum free space for installation	512 MB	512 MB	18.6 MB	15.9 MB	At least 375 MB

Space requirements for the embedded messaging feature are described in "Installing WebSphere embedded messaging as the JMS provider".

- **Configure HP-UX kernel settings before installing.** To set kernel parameters, perform the following steps:
  1. Log into the host machine with superuser (root) privileges.
  2. Determine the physical memory, which you must know to avoid setting certain kernel parameters above the physical capacity:

a. Start the HP-UX System Administration Manager (SAM) utility.

b. Select **Performance Monitors > System Properties > Memory**.

c. Note the value for Physical Memory and click **OK**.

d. Exit from the SAM utility.

3. Set the maxfiles and maxfiles_lim parameters to at least 4096. (The following table recommends 8000 and 8196, respectively. You must first edit the `/usr/conf/master.d/core-hpux` file, to allow the SAM utility to set values greater than 2048:

   a. Open the file `/usr/conf/master.d/core-hpux` in a text editor.

   b. Change the line," *range maxfiles<=2048″ to ″*range maxfiles<=60000″

   c. Change the line, ″*range maxfiles_lim<=2048″ to ″*range maxfiles_lim<=60000″

   d. Save and close the file. Old values might be stored in the `/var/sam/boot.config` file. Force the SAM utility to create a new `boot.config` file:

      1) Move the existing version of the `/var/sam/boot.config` file to another location, such as the `/tmp` directory.

      2) Start the SAM utility.

      3) Select **Kernel Configuration > Configurable Parameters**. When the Kernel Configuration window opens, a new `boot.config` file exists.

         Alternatively, rebuild the `boot.config` file with this command:

         `# /usr/sam/lbin/getkinfo -b`

4. Set new kernel parameter values:

   a. Start the SAM utility.

   b. Select **Kernel Configuration > Configurable Parameters**.

   c. For each of the parameters in the following table, perform this procedure:

      1) Highlight the parameter to change.

      2) Select **Actions > Modify Configurable Parameter**.

      3) Type the new value in the **Formula/Value** field.

      4) Click **OK**.

Some kernel values for WebSphere Application Server with the embedded messaging feature are higher than those shown in the following table. See ″Installing WebSphere embedded messaging as the JMS provider″ for more information.

Some kernel values for WebSphere Application Server IBM DB2 on the same machine, are higher than those shown in the following table.

– Recommended HP-UX kernel configuration parameters for DB2 V8

– Recommended HP-UX kernel configuration parameters for DB2 V7

Typical kernel settings for running WebSphere Application Server appear in the following table:

Parameter	Value
dbc_max_pct	25
maxdsiz	805306358
maxdsiz	2048000000 (when the base and Network Deployment products are on one machine)
maxfiles_lim	8196 (Change this one before maxfiles.)
maxfiles	8000
maxssiz	8388608
maxswapchunks	8192
max_thread_proc	3000
maxuprc	512
maxusers	512

Parameter	Value
msgmap	2048
msgmax	65535
msgmax	131070 (when the base and Network Deployment products are on one machine)
msgmnb	65535
msgmnb	131070 (when the base and Network Deployment products are on one machine)
msgmni	50
msgseg	32767
msgssz	32
msgtql	2046
nfile	58145
nflocks	3000
ninode	60000
nkthread	7219
nproc	4116
npty	2024
nstrpty	1024
nstrtel	60
semmap	514
semmni	2048
semmns	2048
semmnu	1024
semume	200
shmmax	2147483647
shmmni	1024
shmseg	1024
STRMSGSZ	65535

5. Select **Actions > Process New Kernel**.

6. Click **Yes** on the information window to confirm your decision to restart the machine.

   Follow on-screen instructions to restart your machine and to enable the new settings.

7. To redirect displays to non-HP machines, do the following before running the WebSphere Application Server installation wizard:

   a. Issue the following command to obtain information on all public locales accessible to your application:

   `# locale -a`

   b. Choose a value for your system from the output that is displayed and set the LANG environment variable to this value. Here is an example command that sets the value of LANG to en_US.iso88591

   `# export LANG=en_US.iso8859`

- **Avoid using NetScape 4.79 on HP-UX 11 in Japanese, to avoid problems in viewing the administrative console.**

  It is possible that you might be unable to view the menu or areas of the console that scroll. Only certain portions of the administrative console are visible. It is not possible to scale the administrative console to view it all. It is difficult to see what is displayed.

  The workaround for the problem is to use another supported browser, or another browser and operating system platform. You can also open the menu frame in a separate window, to avoid the problem.

- **Configure the converter.properties file to use EUC-JP (Japanese) encoding on HP-UX**

  The `install_root/java/jre/lib/i18n.jar` file on HP-UX platforms does not have the coverters for Cp33722C, but does have the converter for Cp33722. To use EUC-JP encoding on HP-UX platforms, change the EUC-JP=Cp33722C entry in the `converter.properties` file to EUC-JP=Cp33722 or EUC-JP=EUC_JP.

Linux platforms
Summary of Linux platform tips

- Providing adequate disk space for installation

- **5.0.1** Preparing the SuSE Linux Enterprise Server 8.0 - Powered by UnitedLinux 1.0 operating platform for WebSphere Application Server installation

- Avoiding the certificate revocation list (CRL) function, which is not supported for IBM HTTP Server on Linux for S/390 at this time

- Using the ikeycmd command line interface for ikeyman from IBM HTTP Server on Linux for S/390

- Migrating from WebSphere Application Server, V3.5 on Linux platforms

- Updating the kernel version in Red Hat Linux 7.2 from V2.4.9-37 to V2.4.9-38 and obtaining OCO device drivers for the new kernel

- Installing compat-libstdc++ before starting the IBM HTTP Server using RedHat 7.2

- Avoiding utility hangs.

- **5.0.1** Accessing First Steps items on Linux for S/390 systems

Linux platform tips

- **Providing adequate disk space for installation**

Table 6. Installed sizes of installation directories for Linux platforms

	Base product	Network Deployment product	IBM HTTP Server	Tivoli Global Security Kit (base product only)	Temporary space requirement
Installation directory	/opt/ WebSphere/ AppServer	/opt/ WebSphere/ DeploymentManager	/opt/ WebSphere/ DeploymentManager	/opt/ ibm/ gsk5	/tmp
Minimum free space for Linux/Intel installation	375 MB	375 MB	20.4 MB	11.8 MB	At least 375 MB
Minimum free space for Linux for S/390 installation	512 MB	512 MB	20.4 MB	13.2 MB	At least 375 MB

Space requirements for the embedded messaging feature are described in "Installing WebSphere embedded messaging as the JMS provider".

- **5.0.1** **Preparing the SuSE Linux Enterprise Server 8.0 - Powered by UnitedLinux 1.0 operating platform for WebSphere Application Server installation.**

Perform each of these tasks to prepare the platform:

  – Install SP2 for the UnitedLinux 1.0 operating platform to let you use the WebSphere Application Server LaunchPad.

  – Use the IBM Developer Kit that WebSphere Application Server provides to support the Java 2 SDK on the SuSE SLES 8.0 operating system to avoid potential problems when uninstalling an interim fix or a fix pack.

    To use the IBM Developer Kit, remove the java2-jre-1.3.1-524 and java2-1.3.1-524 RPMs from the machine before installing WebSphere Application Server.

- **Avoiding the certificate revocation list (CRL) function, which is not supported for IBM HTTP Server on Linux for S/390 at this time.**

Do not use the certificate revocation list (CRL) function on Linux for S/390 at this time.

- **Using the ikeycmd command line interface for ikeyman from IBM HTTP Server on Linux for S/390.**

  You could see a Java core dump after executing an **ikeyman** command function, such as creating the stash file. This has occurred on both RedHat and SuSe releases. It is the result of a conflict in library routines caused by the default loading sequence.

  To work around this problem, set the LD_PRELOAD environment variable before running the command:

  ```
  LD_PRELOAD=/usr/lib/libstdc++-libc6.1-2.so.3
  ```

  This loads the library first when the application is started. Setting this environment variable is also necessary to allow Secure Socket Layer to work correctly on Linux for S/390.

- **Migrating from WebSphere Application Server, Version 3.5 on Linux platforms.**

  The error can occur when the migration tools cannot find a Java home. The **WASPreUpgrade** command reports the resulting error during backup of the WebSphere 3.5 environment. The error appears in the WASPreUpgrade log as:

  ```
  MIGR0257E:Environment variable JAVA_HOME was not set is generated
  ```

  To work around the problem:

  1. Create a directory link to one of the following directories:
     - $WAS_HOME/IBMJavaLink
     - /opt/IBMJava2-122
     - $WAS_HOME/IBMJava2-122

     WAS_HOME is the WebSphere Application Server 3.5 WAS_HOME.

  2. Run the **WASPreUpgrade** command again, followed by the **WASPostUpgrade** command, to migrate the environment correctly to WebSphere Application Server, V5.

- **Updating the kernel version in Red Hat Linux 7.2 from Version 2.4.9-37 to Version 2.4.9-38 and obtaining OCO device drivers for the new kernel.**

  The kernel version provided with Red Hat Linux 7.2 is 2.4.9-37. If this is a new installation you must be sure that you have the right level of object code only (OCO) modules for this kernel version. You need the OCO modules if you are using hardware that requires IBM object code only device drivers. You can obtain the OCO code at http://www10.software.ibm.com/ developerworks/opensource/Linux for S/390/special_oco_rh_2.4.shtml.

  To run IBM SDK 1.3.1 on your Red Hat Linux 7.2 distribution, install the Red Hat-supplied RPM that provides kernel V2.4.9-38. This requires installing the correct level of IBM-supplied OCO modules, if you require the device drivers. You can obtain the RPM file that installs the OCO modules at http://www10.software.ibm.com/ developerworks/opensource/Linux for S/390/special_oco_rh_2.4.shtml.

  The Red Hat supplied kernel errata 2.4.9-38 and installation instructions are at http://rhn.redhat.com/errata/RHBA-2002-198.html. You must install the kernel errata and OCO modules at the same time.

- **Installing compat-libstdc++ before starting the IBM HTTP Server using RedHat 7.2.**

  If you get the following message when starting IBM HTTP Server on RedHat 7.2, you must install compat-libstdc++:

  ```
  /opt/IBMHttpServer/bin/httpd: error while loading shared libraries: libstdc++-libc6.1-1.so.2:
    cannot open shared object file: No such file or directory
    ./apachectl start: httpd could not be started
  ```

  To check if compat-libstdc++ is installed, run the command **rpm -qa | grep compat-libstdc++**. If it is not installed, install this rpm from the RedHat 7.2 CD. If it is installed and you are still having problems starting IBM HTTP Server, run the **ldconfig system** command. This lets the IBM HTTP Server start successfully.

- **Avoiding utility hangs.**

The default Red Hat installation creates an association between the hostname of the machine and the loopback address, 127.0.0.1. In addition, the /etc/nsswitch.conf file is set up to use /etc/hosts before trying to look up the server using a name server (DNS). This loopback processing can hang utilities that start and stop a server, such as startServer.sh and others, even though the server might successfully start or stop.

Verify that the host name is defined properly. The default configuration has localhost defined in the `/etc/hosts` file. The default `/etc/nsswitch.conf` looks only at the host file and not the DNS server.

To correct this problem, remove the 127.0.0.1 mapping to localhost in the `/etc/hosts` file or edit the name service configuration file (`/etc/nsswitch.conf`) to resolve the proper host name by using the name server.

For example, remove the 127.0.0.1 mapping from the `/etc/hosts` file, which might look like this example:

```
# IP Address              name of machine
n.n.n.n                   hostname.domain.com     hostname
127.0.0.1                 localhost
```

Otherwise, change the `etc/nsswitch.conf` file to search DNS before searching the hosts file.

For example, `hosts : dns files`

- **5.0.1** **Accessing First Steps items on Linux for S/390 systems**

When you select the following features from the First Steps GUI, the Web browser window does not open:

  – WebSphere InfoCenter
  – Register the Product
  – Samples Gallery
  – Administrative Console

To access these features:

1. Open a Web browser window on another machine.
2. Type the browser URL that displays in the First Steps status window into the address field of the Web browser.
3. Click **Enter** to open the page in the Web browser.

Solaris Operating Environment
Summary of Solaris platform tips

- Space requirements for installation.

- **5.0.1** System calls are not restarted correctly within the Java virtual machine
- Install the Sun Java Development Kit, 1.3.1_07 to fix a problem restarting system calls correctly within the Java virtual machine.
- Use the unzip function, not the jar command, to uncompress downloaded files.
- Double-byte character set (DBCS) characters are not supported in the name of the installation directory on Solaris systems.
- Spaces are not supported in the name of the installation directory on Solaris systems.
- Close the terminal window that remains open after the installation finishes.
- Configure the converter.properties file to use EUC-JP (Japanese) encoding on Solaris.

- **5.0.1** Updating GSKit from level 5.0.5.48 to level 5.0.5.70 on the Solaris operating environment to eliminate a security exposure

- **5.0.1** The Domino Server plug-in fails to configure on Solaris Operating Environment Solaris platform tips

-

Solaris platform tips

- **Space requirements for Solaris platforms**

Table 7. Installed sizes of installation directories for Solaris platforms

	Base product	Network Deployment product	IBM HTTP Server	Tivoli Global Security Kit (base product only)	Temporary space requirement
Installation directory	`/opt/ WebSphere/ AppServer`	`/opt/ WebSphere/ DeploymentManager`	`/opt/ WebSphere/ DeploymentManager`	`/opt/ ibm/ gsk5`	`/tmp`
Minimum free space for installation	532 MB	532 MB	21.2 MB	23.1 MB (GSKit 5); 19.6 MB (GSKit4)	At least 375 MB

Space requirements for the embedded messaging feature are described in ″Installing WebSphere embedded messaging as the JMS provider″.

- **5.0.1** **System calls are not restarted correctly within the Java virtual machine**

A problem exists in the Sun Java Development Kit on the Solaris Operating Environment, which is fixed in 1.3.1_07 (FCS Jan 2003), when the socket reads and writes randomly thrown InterruptedIOException or java.net.SocketException exceptions. The root problem is that system calls are not restarting as they should within the Java virtual machine.

See bugs numbers 4425033 and 4178050 at http://developer.java.sun.com/developer/bugParade for further details. If previous exceptions are encountered and are having a significant impact on your operation before availability of 1.3.1_07, contact IBM for support.

- **Install the Sun Java Development Kit, 1.3.1_07 to fix a problem restarting system calls correctly within the Java virtual machine.**

A problem exists in the Sun Java Development Kit on the Solaris Operating Environment, which is fixed in 1.3.1_07 (FCS Jan 2003). The problem is random socket reads and writes, which throw InterruptedIOException or java.net.SocketException exceptions. The result is that system calls do not restart as they should within the Java virtual machine.

The descriptions of Java bugs 4425033 and 4178050 at http://developer.java.sun.com/developer/bugParade provide further details. If you encounter similar symptoms and they have a significant impact on your operation before 1.3.1_07 is available, contact IBM support.

- **Use the unzip function, not the jar command, to uncompress downloaded files.**

To uncompress downloaded installation files for the Solaris operating system, use the **unzip** function and not the **jar** function. Using the **jar** function sets file permissions incorrectly, which causes the installation to fail.

- **Double-byte character set (DBCS) characters are not supported in the name of the installation directory on Solaris systems.**

Do not use double-byte characters in the installation directory name on a Solaris system.

- **Spaces are not supported in the name of the installation directory on Solaris systems.**

Do not use spaces in the installation directory name on a Solaris system. This is a limitation of the IBM Software Development Kit.

- **Close the terminal window that remains open after the installation finishes.**

When installing WebSphere Application Server from the product CD onto a Solaris system, the ISMP installation wizard launches a terminal window, which remains open after the installation is complete. This window contains the following text:

```
InstallShield Wizard
Initializing InstallShield Wizard...

Searching for Java(tm) Virtual Machine...
.....
```

Close the window after the installation completes.

- **Configure the converter.properties file to use EUC-JP (Japanese) encoding on Solaris**

  The `install_root/java/jre/lib/i18n.jar` file on Solaris platforms does not have the coverters for Cp33722C, but does have the converter for Cp33722. To use EUC-JP encoding on Solaris platforms, change the EUC-JP=Cp33722C entry in the `converter.properties` file to EUC-JP=Cp33722 or EUC-JP=EUC_JP.

- **5.0.1** **Updating GSKit from level 5.0.5.48 to level 5.0.5.70 on the Solaris operating environment to eliminate a security exposure**

  After you apply WebSphere Application Server, Version 5.0.1, the GSKit does not get updated on the Solaris operating environment. To update the GSKit from level 5.0.5.48 to level 5.0.5.70, perform the following steps:

  1. Change the directory to the gsk5install directory located in the `/usr/WebSphere/AppServer1` directory.
  2. Issue a **pkgadd -n -d . gsk5bas** command from the command prompt or install using the Solaris administrative tool utility.
  3. Issue the **gsk5ver** command to verify the update to level 5.0.5.70.

- **5.0.1** **The Domino Server plug-in fails to configure on Solaris Operating Environment**

  During installation, a dsapi_stderr.txt file is created in the logs directory and you can get the following error messages:

```
lotus.notes.NotesException: Could not load dll for system name SUNOS
        at lotus.notes.NotesThread.load(NotesThread.java:210)
        at lotus.notes.NotesThread.<clinit>(NotesThread.java:24)
java.lang.UnsatisfiedLinkError: NnotesInitThread
        at lotus.notes.NotesThread.NnotesInitThread(Native Method)
        at lotus.notes.NotesThread.initThread(NotesThread.java:99)
        at lotus.notes.NotesThread.run(NotesThread.java:133)
```

  You can configure the IBM WebSphere Application Server or Domino Server plug-in manually using the Domino Server Web administration tool. Here is a work-around procedure:

  1. Start the Domino Server.
  2. Enter the URL for the Domino Server Web Administration site using a browser. For example, http://*hostname*//names.nsf.
  3. Enter the administrator user name and password.
  4. Double-click **Server-Servers**.
  5. Double-click **WebServer** to configure.
  6. Double-click **Edit Server**.
  7. Double-click **Internet Protocol**.
  8. Add the IBM WebSphere Application Server DSAPI plug-in to the DSAPI field. For example, /opt/WebSphere/AppServer/bin/libdomino5_http.so

     If there are already DSAPI filter files specified, use a space to delimit the WebSphere plug-in file.
  9. Double-click **Save and Close**.
  10. Restart the Domino Server.

Windows platforms
Summary of Windows platform tips

- Space requirements for installation.

- **5.0.1** Cannot install the pluggable client from the IBM WebSphere Application Server client CD on a Windows system

- Migration panels on Windows platform shows corrupted characters.

- If you are installing by downloading WebSphere Application Server from IBM, use another unzip product, such as WINZIP, instead of the PKWARE pkunzip utility to decompress the product archive.

- Avoid spaces in the names of the installation directory and the migration backup directory on Windows NT systems.

- The installation hangs or there are font problems that are often fixed by editing the vpd.properties file.

- Prepare for Adaptive Fast Path Architecture (AFPA) driver availability when migrating from IBM HTTP Server V1.3.19.x, or earlier.

- Reboot the machine after uninstalling the embedded messaging feature.

- ImagePath entry in registry not updated when you change the document root

- **5.0.1** You might receive a WUPD0248E exception when installing Fix Pack 1 on Windows NT platforms

Windows platform tips

- **Space requirements for Windows platforms**

Table 8. Installed sizes of installation directories for Windows platforms

	Base product	Network Deployment product	IBM HTTP Server	Tivoli Global Security Kit (base productonly)	Temporary space requirement
Installation directory	`C:\ Program Files\ WebSphere\ AppServer`	`C:\ Program Files\ WebSphere\ DeploymentManager`	`C:\ Program Files\ IBMHttpServer`	`C:\ Program Files\ IBM\ gsK5`	`C:\ temp`
Minimum space for installation	520 MB	520 MB	17.3 MB	16.8 MB	At least 375 MB

Space requirements for the embedded messaging feature are described in ″Installing WebSphere embedded messaging as the JMS provider″.

- **5.0.1** **Cannot install the pluggable client from the IBM WebSphere Application Server client CD on a Windows system**

You cannot install the pluggable client from the IBM WebSphere Application Server client CD on a Windows system.

To work around this problem, do one of the following:

1. Copy the IBM WebSphere Application Server client CD for Windows installation image to your local temp directory and install it from there.

2. Run the installer manually from a command line with the following options:

```
cd drive\nt\install.exe
    -P OrbPropertiesFileCopy.installLocation="SunJREPath/lib"
```

Where *cd drive* is the CD drive and *SunJREPath* is the location of the Sun operating environment JRE file installed on your machine.

For example:

```
F:\nt\install.exe
    -P OrbPropertiesFileCopy.installLocation=
       "C:/JavaSoft/JRE/1.3.1_04/lib"
```

- **Migration panels on Windows platform shows corrupted characters.**

  When migration is running during installation, the information displayed in the pre-migration and post-migration processing might display corrupted national characters. Use the PreUpgrade and PostUpgrade logs in the `<backup>/logs` directory, where `<backup>` is the backup directory specified during installation.

- **If you are installing by downloading WebSphere Application Server from IBM, use another unzip product, such as WINZIP, instead of the PKWARE pkunzip utility to decompress the product archive.**

  If you are using the downloadable archive file to install the WebSphere Application Server product, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

- **Avoid spaces in the names of the installation directory and the migration backup directory on Windows NT systems.**

  Do not use spaces in the installation directory name or the migration backup directory name on a Windows NT system. This avoids a problem during migration, in the call to ejbdeploy or wsadmin, which each fail with the following error when there is a space in the WAS 5.0 base install root or the backup directory:

  ```
  The name specified is not recognized as an
  internal or external command, operable program or batch file.
  ```

- **The installation hangs or there are font problems that are often fixed by editing the vpd.properties file.**

  Installing WebSphere Application Server on some international computers might result in installation failure or font problems. To work around this problem, follow these steps:

  1. Locate the `vpd.properties` file in the operating system installation directory.

     For example, `C:\WINDOWS` or `C:\WINNT` on a Windows system.

  2. Remove all lines containing one of these strings:
     - WSE
     - WSN
     - WSB
     - WSM

     Do not delete or rename the vpd.properties file because the InstallShield for MultiPlatforms (ISMP) program uses it for other products that it installs.

- **Prepare for Adaptive Fast Path Architecture (AFPA) driver availability when migrating from IBM HTTP Server V1.3.19.x, or earlier.**

  The AFPA driver controls the fast response cache accelerator function, which is also known as the *cache accelerator.* The version of IBM HTTP Server installed with WebSphere Application Server shares the AFPA driver with any coexisting IBM HTTP Server. Uninstalling a coexisting V1.3.19.x (or earlier) IBM HTTP Server also uninstalls the common AFPA driver.

  When configured to use an AFPA driver that is no longer present, IBM HTTP Server generates errors as it starts and there is no response improvement from the cache accelerator. For example:

  ```
  [error] (9) Bad file descriptor: Afpa Device Driver open failed.
  ```

  You can either restore the driver or disable the cache accelerator configuration. Restore the driver by reinstalling the later version of IBM HTTP Server after you uninstall the earlier version. To verify that the AFPA driver is installed and working, see if it is listed in device manager under `Non-Plug and Play Drivers`. Be sure to select the `Show hidden devices` option in the device manager view on Windows 2003 systems.

You can disable AFPA, by commenting out the following directives in the `httpd.conf` configuration file:

```
AfpaEnable
AfpaCache on
AfpaLogFile "C:\Program Files\IBM HTTP Server\log\afpalog" V-ECLF
```

- **Reboot the machine after uninstalling the embedded messaging feature.**

  If you install and then subsequently manually uninstall the embedded messaging feature on a Windows machine, you must reboot the machine before reinstalling the embedded messaging feature.

- **ImagePath entry in registry not updated when you change the document root**

  When you change the document root in the IBM HTTP Server 1.3.26.x administrative server, the ImagePath entry in the registry, corresponding to the HTTP server, is not updated. There have been no reported functionality problems caused by this incorrect registry entry.

- **5.0.1** **You might receive a WUPD0248E exception when installing the fix pack**

  When Fix Pack 1 on Windows NT platforms, you might receive an error similar to the following:

```
    Exception: WUPD0248E: Fix pack update failure:
        The processing of fix pack was50_fp1_win,
            component prereq.jsse failed.
```

  If you receive the error message, go to the Java directory under the directory where WebSphere Application Server, Version 5 is installed, for example, the WebSphere/AppServer/java directory, and run the **attrib -r /s** command.

  This problem is related to the read-only attribute being set for a number of files in the `WebSphere/AppServer/java` directory and its subdirectories.

Tips for installing the embedded messaging feature

When installing IBM WebSphere Application Server, you can install the embedded messaging feature (selected by default) for use as the Java message service (JMS) provider. To install the embedded messaging feature successfully depends on you completing several other actions first. These actions include considering whether or not you also want to use WebSphere MQ on the same host and, on UNIX, defining the groups and users needed for embedded messaging.

You can either complete the required actions before installing the WebSphere Application Server product, or deselect the embedded messaging feature, which installs the WebSphere Application Server product without embedded messaging.

If you already have WebSphere MQ installed, you can configure it as the JMS provider. Otherwise, you can install the embedded messaging feature, the WebSphere MQ product, or another JMS provider later, after you install the WebSphere Application Server product.

The WebSphere Application Server embedded messaging feature provides a JMS provider that supports both queues (for point-to-point messaging) and topics (for publish and subscribe messaging). You can install the WebSphere Application Server with the embedded messaging feature on the same host with the WebSphere MQ product. To support this combination, several WebSphere MQ product features must be at a certain level.

After installing the WebSphere Application Server product with the embedded messaging feature, you can install the WebSphere MQ product and use it as the JMS provider instead.

Applying fixes and fix packs to the embedded messaging feature

Always apply any outstanding corrective service to the stand-alone IBM WebSphere MQ product if you have it, before using the WebSphere Application Server update installer to update the embedded messaging feature with service in an interim fix or fix pack. For example:

- If you have the embedded messaging server and client features, apply service to the embedded messaging feature that is included in any WebSphere Application Server fix pack.
- If you have the embedded messaging client feature and a stand-alone IBM WebSphere MQ product providing the JMS, apply any outstanding corrective service to the IBM WebSphere MQ product before applying service to the embedded messaging feature.
- If you have the embedded messaging client feature and a stand-alone IBM WebSphere MQ product at the latest corrective service level, apply service to the embedded messaging feature that is included in any WebSphere Application Server fix pack.

Summary of tips for using the embedded messaging feature
The following descriptions are in this topic:
- General space requirements
- Default installation path
- Installs to fixed locations on UNIX-based operating systems
- Before installing WebSphere embedded messaging, create and mount a journalized file system called /var/mqm
- No need to install server or client subfeature twice
- InvalidExecutableException while starting jmsserver
- No support for terminal services with the embedded messaging feature
- Define the UNIX or Linux operating system groups, mqm and mqbrkrs, before installing the embedded messaging feature
- After installing WebSphere embedded messaging, restrict access to the /var/mqm/errors directory and messaging logging files
- Install Java130.rte.lib V1.3.0 on AIX V4.3.3 or AIX V5 before installing the embedded messaging feature
- Correct directory permissions on AIX platforms before reinstalling WebSphere Application Server with the embedded messaging feature
- Choose whether to install the embedded messaging server feature if WebSphere MQ V5.3 is already installed
- Reboot the machine after uninstalling the embedded messaging feature
- Coexistence problem between embedded messaging, IBM WebSphere Studio Application Developer Integration Edition and IBM WebSphere Application Server
- Preparing to uninstall WebSphere Application Server, Version 5 Fix Pack (with WebSphere Embedded Messaging) on Solaris
- More information

Tips for using the embedded messaging feature
Tips for using the embedded messaging feature include:
- **General space requirements**

  On Windows platforms, the size of the embedded messaging feature code for both the client and the server subfeatures is approximately 121 MB. Linux/Intel platforms require approximately 188 MB. AIX platforms require approximately 101 MB. HP-UX platforms require approximately 193 MB. Solaris platforms require approximately 180 MB.

  ″Installing WebSphere embedded messaging as the JMS provider″ describes space requirements in detail.
- **Default installation path**

  On Windows platforms, the default installation path is `drive:\Program Files\IBM\WebSphere MQ\`. You can specify a different directory during installation.
- **Installs to fixed locations on UNIX-based operating systems**

  On UNIX-based operating systems, the embedded messaging feature installs to fixed locations that you cannot override. The default locations are:

- Linux platforms: `/var/mqm`, `/var/wemps`, `/opt/mqm`, `/opt/wemps`
- AIX platforms: `/usr/opt/mqm`, `/usr/opt/wemps`, `/var/mqm`
- Solaris platforms: `/var/mqm`, `/var/wemps`, and `/usr`

"Installing WebSphere embedded messaging as the JMS provider" describes directory locations in detail.

- **Before installing WebSphere embedded messaging, create and mount a journalized file system called /var/mqm**

  On UNIX platforms, use a partition strategy with a separate volume for the messaging data. This means that other system activity is not affected if a large amount of messaging work builds up in /var/mqm.

  The /var file system is used to store all the security logging information for the system, and is used to store the temporary files for email and printing. Therefore, it is critical that you maintain free space in /var for these operations. If you do not create a separate file system for messaging data, and /var fills up, all security logging will be stopped on the system until some free space is available in /var. Also, email and printing will no longer be possible until some free space is available in /var.

- **No need to install server or client subfeature twice**

  Though you might install the base product in combination with the Network Deployment or Enterprise products, there is no need to install the server or client subfeatures more than once.

  To install the base and Network Deployment products, or the base and the Enterprise products on a single Windows machine, install the base product embedded messaging subfeatures instead of the Network Deployment client feature, or instead of the Enterprise server and client subfeatures. Using the base product features avoids a potential problem caused by a bug in the Microsoft Software Installer utility.

- **InvalidExecutableException while starting jmsserver**

  You might get an exception while starting the jmsserver when you install Network Deployment first and then install the base WebSphere Application Server product and its embedded messaging feature on the same node. The error message is recorded in the `install_root/logs/jmsserver/SystemOut.log` file:

```
[9/5/02 14:35:37:818 EDT] 36349b90 JMSService
E MSGS0001E: Starting the Server failed with exception: com.ibm.ws.process.exception.
InvalidExecutableException: Error creating new process.
  002: No such file or directory
```

  In addition, although `mq_install.log` files might appear to run without error, the `createMQ.nodeName_jmsserver.log` file can contain I/O exceptions. These exceptions result from a corrupted installation of the embedded messaging feature caused by installing the Network Deployment product before the base WebSphere Application Server product. The workaround is to uninstall both products, reinstall the base WebSphere Application Server product, and then reinstall the Network Deployment product.

- **No support for terminal services with the embedded messaging feature**

  Terminal services is not supported as a valid installation scenario when installing WebSphere Application Server and the embedded messaging feature.

- **Define the UNIX or Linux operating system groups, mqm and mqbrkrs, before installing the embedded messaging feature**

  Before you install the embedded messaging component on UNIX or Linux platforms, you must define the operating system groups **mqm** and **mqbrkrs**, and the user IDs needed for WebSphere embedded messaging. For detailed information, see Installing WebSphere embedded messaging as the JMS provider.

- **After installing WebSphere embedded messaging, restrict access to the /var/mqm/errors directory and messaging logging files**

  After installing WebSphere embedded messaging, you must restrict access to the /var/mqm directories and log files needed for WebSphere embedded messaging, such that only the user ID mqm or

members of the mqm user group have write access. For detailed information, see *Installing WebSphere embedded messaging as the JMS provider* and the *Securing messaging directories and log files* (tmj_secmqm) topic in the InfoCenter.

- **Install Java130.rte.lib V1.3.0 on AIX V4.3.3 or AIX V5 before installing the embedded messaging feature**

  On AIX V4.3.3 and AIX V5, you must install `Java130.rte.lib` V1.3.0 to verify that the embedded messaging feature installs correctly. To download a copy of Java 1.3.0:

  1. Go to `www.ibm.com\java: Java Technology Zone`.
  2. Go to the bottom of the page: **Most popular links.**
  3. Click **IBM Developer Kit for AIX**.
  4. Click **Download**.
  5. Click **Java 1.3.0** from the **Java Version** column in the table.
  6. (Optional) Register for a user ID and password.

  To correct an existing problem:

  1. Uninstall the following components, if present:
     - WebSphere Application Server
     - MQSeries
     - MQSeries classes for Java and MQSeries for Java Message Service
     - IBM HTTP Server
     - WebSphere Embedded Messaging Publishing and Subscribe Edition
  2. Edit the `vpd.properties` file and remove any entry related to WebSphere Application Server. WebSphere Application Server-related entries begin with:
     - **WSE** for the Enterprise product
     - **WSB** for the base product
     - **WSN** for the Network Deployment product
     - **WSM** for the WebSphere MQ product
  3. Reinstall the WebSphere Application Server product.

- **Correct directory permissions on AIX platforms before reinstalling WebSphere Application Server with the embedded messaging feature**

  After manually uninstalling the WebSphere Application Server product and before reinstalling the product with the embedded messaging component, look for the following directory: `/var/mqm/log/WAS_system_name_server_name`. If this directory exists, confirm that the directory is empty and that the user, mqm, can open and write to it. If it is not accessible, the embedded messaging installation program throws the following error:

  ```
  AMQ7064: Log path not valid or inaccessible is written in the
           createMQ.system_name_server_name.log file.
  ```

  If this error occurs, the remaining embedded messaging component installation fails.

- **Choose whether to install the embedded messaging server feature if WebSphere MQ V5.3 is already installed**

  You have a choice if you already have WebSphere MQ V5.3 installed:

  - You can install only the embedded messaging client feature on a machine that already has WebSphere MQ V5.3.

    To use WebSphere MQ V5.3 as the JMS provider, install IBM WebSphere Application Server with only the embedded messaging client feature. Installing and using the WebSphere Application Server embedded messaging client feature is recommended with either the server feature or the full WebSphere MQ V5.3 product.

WebSphere Application Server messaging applications can use the WebSphere MQ V5.3 product as the JMS provider. Using the client feature, however, requires that you install the WebSphere MQ V5.3 Java messaging feature.

– You can install the embedded messaging server and client features on a machine that already has WebSphere MQ V5.3.

To install the embedded messaging server feature when WebSphere MQ V5.3 is already installed, upgrade WebSphere MQ V5.3:

- Apply the CSD01 update to the original WebSphere MQ V5.3 release, or move to the WebSphere MQ V5.3 refresh release (which includes CSD01).

- Install the WebSphere MQ V5.3 features, server and Java messaging, which the WebSphere Application Server embedded messaging server feature requires.

If you install WebSphere MQ V5.3 without the required features, the installation of either IBM WebSphere Application Server embedded messaging feature is unsuccessful because of prerequisite check errors. The WebSphere Application Server Enterprise package includes installation images of the WebSphere MQ V5.3 product and the WebSphere MQ Event Broker product, with restricted licensing. You can use the products to install the required WebSphere MQ V5.3 features or to install the refresh release of WebSphere MQ V5.3 for use with WebSphere Application Server Enterprise.

- **Reboot the machine after uninstalling the embedded messaging feature**

If you install and then subsequently manually uninstall the embedded messaging feature on a Windows machine, you must reboot the machine before reinstalling the embedded messaging feature.

- **Avoid a coexistence problem between embedded messaging, IBM WebSphere Studio Application Developer Integration Edition and IBM WebSphere Application Server**

The IBM WebSphere Studio Application Developer Integration Edition and IBM WebSphere Application Server both include an option to install embedded messaging. The embedded messaging option in these two products is incompatible.

To avoid this problem, do not install embedded messaging for both products on the same machine.

- **Preparing to uninstall WebSphere Application Server, Version 5 Fix Pack (with WebSphere Embedded Messaging) on Solaris**

If you installed WebSphere Application Server with both the embedded messaging client and server features, you can ignore this item because it does not apply to your installation. This item applies when installing the embedded messaging client feature only.

If you have a WebSphere Application Server or Network Deployment installation that includes the embedded messaging client without the server component, complete the following steps before uninstalling Version 5 Fix Pack 1 on the Solaris Operating Environment:

– **WebSphere Application Server and embedded messaging client (only):**

1. Locate and delete the mqVer.properties file in the `/ opt / WebSphere / AppServer / properties / version / backup / external.mq / ptfs / was50_fp1_solaris / components / external.mq` directory. (Spaces added around each slash for formatting clarity.)

2. Manually uninstall the embedded messaging feature -`/usr/sbin/pkgrm mqm-upd03`.

3. Continue with uninstalling WebSphere Application Server, V5 Fix Pack 1.

– **Network Deployment and embedded messaging client:**

1. Locate and delete the mqVer.properties file in the `/ opt / WebSphere / DeploymentManager / properties / version / backup / external.mq / ptfs / was50_nd_fp1_solaris / components / external.mq` directory. (Spaces added around each slash for formatting clarity.)

2. Manually uninstall the embedded messaging feature - `/usr/sbin/pkgrm mqm-upd03`.

3. Continue with uninstalling V5 Fix Pack 1.

If you do not complete this procedure, when you uninstall the Version 5 Fix Pack 1 on Solaris, the following error messages are added to the fix pack uninstallation log. To recover from these errors, complete the procedure, then repeat the fix pack uninstallation.

```
PTF Component Activity:
==========================================================
Component Name       : external.mq
Action               : uninstall
Time Stamp (Start)   : 2003-04-09T10:41:12-05:00
==========================================================
Log File Name        : /opt/WebSphere/DeploymentManager/properties/version/log/20030
409_154112_was50_nd_fp1_solaris_external.mq_uninstall.log
Backup File Name        : /opt/WebSphere/DeploymentManager/properties/version/backup/20
030409_154112_was50_nd_fp1_solaris_external.mq_undo.jar
==========================================================

Saving History ...
Saving History ... Done

Running component update ( external.mq ).
Processing MQ CSD Uninstall Script
Processing MQ CSD version information.
Performing CSD uninstall.

Failed to perform uninstall.

Exception: WUPD0246E: Fix pack update failure: An exception occurred while
preprocessing the content of fix pack was50_nd_fp1_solaris, component external.mq

Saving History ...
Saving History ... Done

Results:
==========================================================
Time Stamp (End)      : 2003-04-09T10:41:14-05:00
PTF Component Result  : failed
PTF Component Result Message:
==========================================================
WUPD0246E: Fix pack update failure: An exception occurred while preprocessing the
content of fix pack was50_nd_fp1_solaris, component external.mq
==========================================================

PTF Component Installation ... Done

Exception: WUPD0243E: Fix pack uninstall failure: The update for component {1} for
fix pack external.mq could not be uninstalled.
```

- **More information**

  For more information about the actions to take before installing the embedded messaging feature, refer to Installing the WebSphere Application Server embedded messaging feature as the JMS provider. For more information about installing JMS providers, refer to Installing and configuring a JMS provider.

  For information about installing the WebSphere MQ V5.3 product, or migrating to WebSphere MQ V5.3 from an earlier release, refer to the appropriate WebSphere MQ *Quick Beginnings* book:

  – *WebSphere MQ for Windows, V5.3 Quick Beginnings*, GC34-6073
  – *WebSphere MQ for AIX, V5.3 Quick Beginnings*, GC34-6076
  – *WebSphere MQ for Solaris, V5.3 Quick Beginnings*, GC34-6075
  – *WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3 Quick Beginnings*, GC34-6078

  These books are available at the WebSphere MQ messaging platform-specific books Web page at http://www-3.ibm.com/ software/ts/ mqseries/ library/ manualsa/ manuals/ platspecific.html.

Installing WebSphere embedded messaging as the JMS provider

Use this task to install the embedded messaging options of IBM WebSphere Application Server for use as the JMS provider.

Abstract:

If you want to use embedded messaging, you can install the following options of IBM WebSphere Application Server:

Embedded messaging server

> This option installs the messaging server functions of the WebSphere JMS provider. The WebSphere JMS provider supports both queues (for point-to-point messaging) and topics (for publish/subscribe messaging).

Embedded messaging client

> This option installs the messaging client functions that enable applications running in WebSphere Application Server to communicate with the WebSphere JMS provider.

Before you install the embedded messaging options of IBM WebSphere Application Server, you must complete the following steps:

1. If you want to install embedded messaging on a machine where you already have WebSphere MQ installed, you must verify that you have upgraded to WebSphere MQ 5.3 with the required MQ features:

   a. If you have the original WebSphere MQ 5.3 release installed, verify that you have applied the CSD03 update.

   To determine if your WebSphere MQ 5.3 installation is at the required level, run the **mqver** utility provided by WebSphere MQ. The required level as indicated by mqver is shown below:

   ```
   Name:        WebSphere MQ
      Version:     530.3  CSD03
      CMVC level:  p530-CSD03J
      BuildType:   IKAP - (Production)
   ```

   In comparison, earlier levels of the WebSphere MQ 5.3 release have indicated levels such as p000-L020617 and p000-L021011.

   For more information about CSD03 and other WebSphere MQ product support, see the WebSphere MQ support page for your platform.

   b. Verify that you have installed the following WebSphere MQ features:

      - For a WebSphere Application Server **embedded messaging server** installation, the required MQ features are ″Server″ and ″Java Messaging″.
      - For a WebSphere Application Server **embedded messaging client** installation, the only required MQ feature is ″Java Messaging″.

   If you have not installed WebSphere MQ 5.3 with the required MQ features, then installation of IBM WebSphere Application Server embedded messaging options fails with prerequisite check errors.

   The WebSphere Application Server Enterprise package includes copies of the WebSphere MQ 5.3 and Event Broker installation packages, with restricted licensing. (WebSphere MQ Event Broker is not available for Linux.) You can use the provided packages to install the required MQ features or WebSphere MQ 5.3 for use with WebSphere Application Server Enterprise.

   For information about installing WebSphere MQ 5.3, or migrating to WebSphere MQ 5.3 from an earlier release, see the appropriate WebSphere MQ *Quick Beginnings* book, as follows:

   - *WebSphere MQ for Windows, V5.3 Quick Beginnings*, GC34-6073
   - *WebSphere MQ for AIX, V5.3 Quick Beginnings*, GC34-6076
   - *WebSphere MQ for Solaris, V5.3 Quick Beginnings*, GC34-6075
   - **5.0.1** *WebSphere MQ for HP-UX, V5.3 Quick Beginnings*, GC34-6077
   - *WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3 Quick Beginnings*, GC34-6078

   You can get these books from the WebSphere MQ messaging platform-specific books Web page.

2. Verify that there is enough space in the file systems where you want to install the embedded messaging options and store associated messaging data. On UNIX platforms, you also need to create the required file systems before installing the embedded messaging feature.

   - **(Windows only)** You can specify the file system into which the embedded messaging options are installed. The following table lists the default locations for the base messaging functions and the messaging broker functions (for publish and subscribe messaging). The table also provides figures for the file system sizes on which you can base your own calculations.

   Installation directory and space needed for embedded messaging - Windows platform

	Base messaging	Messaging broker
Installation directory	`C:\Program Files\IBM\WebSphere MQ`	`C:\Program Files\IBM\WebSphere MQ\WEMPS`
Typical space needed	70 MB (server) or 15 MB (client)	45 MB (server)

   If you are using the Installation wizard to install IBM WebSphere Application Server, you can specify an install location for the embedded messaging options during either a Full or Custom install. If you want to use a silent install for IBM WebSphere Application Server, you can specify an install location for the embedded messaging options when you customize the options response file before issuing the command to install silently. For more information about performing a silent installation (including editing the options response file), see ″Installing silently″.

   - **(UNIX platforms only)** The file system into which the embedded messaging options are installed is fixed. The following table lists the default locations for the base messaging functions and the messaging broker functions (for publish/subscribe messaging). The table also provides figures for the file system sizes on which you can base your own calculations.

   Installation directory and space needed for embedded messaging - UNIX platforms

	Base code	Broker code	Base data	Broker data
	/usr/mqm	/usr/opt/wemps	/var/mqm	/var/wemps
AIX	40MB (server) or 15MB (client)	80MB (server) or 15MB (client)	8MB (server) or 5MB (client)	5MB (server)
	/opt/mqm	/opt/wemps	/var/mqm	/var/wemps
Linux/Intel	40MB (server) or 15MB (client)	100MB (server) or 15MB (client)	8MB (server) or 5MB (client)	5MB (server)
HP-UX	40MB (server) or 15MB (client)	105MB (server) or 15MB (client)	8MB (server) or 5MB (client)	5MB (server)
Solaris	40MB (server) or 15MB (client)	70MB (server) or 15MB (client)	20MB (server) or 15MB (client)	5MB (server)

   Before you install WebSphere embedded messaging, create and mount a journalized file system called `/var/mqm` for your messaging working data. Use a partition strategy with a separate volume for the WebSphere MQ data. This means that other system activity is not affected if a large amount of messaging work builds up in `/var/mqm`. You can also create separate file systems for your log data (`var/mqm/log`) and error files (`var/mqm/errors`). Store log files on a different physical volume from the embedded messaging queues (`var/mqm`). This verifys data integrity in the case of a hardware failure. If you are creating separate file systems, allow a minimum of 30 MB of storage for `/var/mqm`, 20 MB of storage for `/var/mqm/log`, and 4 MB of storage for `/var/mqm/errors`.

   Note: The `/var` file system is used to store all the security logging information for the system, and is used to store the temporary files for email and printing. Therefore, it is critical that you maintain free space in `/var` for these operations. If you do not create a separate file system for messaging data, and `/var` fills up, all security logging will be stopped on the system until some free space is available in `/var`. Also, email and printing will no longer be possible until some free space is available in /var.

You have the same options for creating file systems for embedded messaging as you do for WebSphere MQ. For example:

- If you cannot install the embedded messaging options in the required file system (for example, if it is too small), you can do one of the following *before* installing the embedded messaging options:
  - Create and mount a new file system for the installation directory.
  - Create a new directory anywhere on your machine, and create a symbolic link from the required installation directory to the new directory. For example, on AIX:

    ```
    mkdir /bigdisk/mqm
    ln -s /bigdisk/mqm /usr/mqm
    ```

3. Define the operating system groups and users needed for embedded messaging:
   - **(UNIX platforms only)**
     a. If you have not already done so, create the groups **mqm** and **mqbrkrs**.
     b. Add the users **mqm** and **root** to the **mqm** group.
     c. Add the user **root** to the **mqbrkrs** group.
        Notes:
        1) You are recommended to run the JMS server process under the root user ID. If you run the JMS server process under another user ID, add that user ID to the **mqm** and **mqbrkrs** groups

           For more information about running servers under a non-root user ID, see the *Running an Application Server with a non-root user ID* (trun_svr_nonroot) topic in the InfoCenter.
        2) User IDs longer than 12 characters cannot be used for authentication with the embedded WebSphere JMS provider.
   - **(Windows only)** Define the process user ID with these authorizations:
     - Assign the user ID to the Administrator group.
     - Give the user ID the advanced user right, Act as part of the operating system.
     - Give the user ID the advanced user right, Log on as a service.

     User IDs longer than 12 characters cannot be used for authentication with the embedded WebSphere JMS provider. For example, the default Windows NT user ID, Administrator, is not valid for use with embedded WebSphere messaging, because it contains 13 characters.

     The IBM WebSphere Application Server installation wizard GUI grants your Windows user ID the advanced user rights, if the user ID belongs to the Administrator group. The silent installation does not. If you create a new user ID on a Windows platform to perform the silent installation, you must restart the system to activate the proper authorizations for the user ID, and to perform a successful silent installation.

     This user ID, the WebSphere Application Server process user ID, is used to start the JMS server (for general JMS support) and the WebSphere Embedded Broker (for WebSphere topic connections).

     Verify that these operating system security settings are used when you next start IBM WebSphere Application Server; for example, either log off and then on again with the process user ID, or open a new shell in which to start IBM WebSphere Application Server.

     You have the same options for creating user IDs and groups for embedded messaging as you do for WebSphere MQ. For more information about creating user IDs and groups for WebSphere MQ, see the section ″Preparing for Installation: Setting up the user ID and group″ in the appropriate WebSphere MQ *Quick Beginnings* book, as listed above.
4. **(Solaris only)** Several Solaris kernel values are typically too small for the embedded messaging options. Starting the internal JMS server or client with insufficient kernel resources produces a First Failure Support Technology (FFST) file in the `/var/mqm/errors` directory.

Before installing embedded messaging, review the machine's configuration. To do this type the following command:

```
sysdef -i
```

The kernel values are set in the `/etc/system` file, as shown in the following example.

```
set shmsys:shminfo_shmmax = 4294967295
set shmsys:shminfo_shmseg = 1024
set shmsys:shminfo_shmmni = 1024
set semsys:seminfo_semaem = 16384
set semsys:seminfo_semmni = 1024
set semsys:seminfo_semmap = 1026
set semsys:seminfo_semmns = 16384
set semsys:seminfo_semmsl = 100
set semsys:seminfo_semopm = 100
set semsys:seminfo_semmnu = 2048
set semsys:seminfo_semume = 256
set msgsys:msginfo_msgmap = 1026
set msgsys:msginfo_msgmax = 65535
set rlim_fd_cur=1024
```

You can change kernel values by editing the `/etc/system` file then rebooting the operating system. For more information about setting up the Solaris system, see the Solaris System Administration documentation; for example, the *Solaris Tunable Parameters Reference Manual*.

Note: Queue managers are generally independent of each other. Therefore system kernel parameters, for example shmmni, semmni, semmns, and semmnu need to allow for the number of queue managers in the system.

5. **(HP-UX only)** Several HP-UX kernel values are typically too small for the embedded messaging options. Starting the internal JMS server or client with insufficient kernel resources produces a First Failure Support Technology (FFST) file in the `/var/mqm/errors` directory.

Before installing embedded messaging, review the machine configuration and, if needed, set appropriate HP-UX kernel settings.

You should set new values for the following messaging-related kernel parameters (along with the base set of kernel parameters required by IBM WebSphere Application Server):

Parameter	Value
sema	1
semaem	16384
semmns	16384
semvmx	32767
shmem	1

This table lists only those messaging-related kernel parameters that either are extra to the base set or need values that are greater than given in the base set.

For information about how to review and set the base set of kernel parameters for IBM WebSphere Application Server, see *Configure HP-UX kernel settings before installing*.

Note: Queue managers are generally independent of each other. Therefore system kernel parameters, for example shmmni, semmni, semmns, and semmnu need to allow for the number of queue managers in the system.

To install the embedded messaging options of WebSphere Application Server for use as the WebSphere JMS provider, complete the following steps:

Steps for this task

1. Login with the WebSphere Application Server process user ID (defined as part of the prerequisites)

   Note: (UNIX platforms only) Login as root.

2. On a machine where you want to host queues or topics, install IBM WebSphere Application Server (installing the product) with the **embedded messaging server** option.

   If you also want Application Servers on the host to run messaging applications, install the **embedded messaging client** option.

   Both options are selected by default.

3. On a machine where you want Application Servers to run messaging applications that use a JMS provider on another host, install IBM WebSphere Application Server with the **embedded messaging client** option.

This task has installed WebSphere Application Server with its embedded messaging server feature as the JMS provider.

You can configure JMS resources to be provided by embedded messaging, by using the Application Server administrative console to define JMS resources.

(UNIX platforms only) Restrict access to the messaging errors directories and logging files; for example, by using the following commands:

1. For the /var/mqm/errors directory:

```
chmod 3777 /var/mqm/errors
chown mqm:mqm /var/mqm/errors

touch /var/mqm/errors/AMQERR01.LOG
chown mqm:mqm /var/mqm/errors/AMQERR01.LOG
chmod 666 /var/mqm/errors/AMQERR01.LOG

touch /var/mqm/errors/AMQERR02.LOG
chown mqm:mqm /var/mqm/errors/AMQERR02.LOG
chmod 666 /var/mqm/errors/AMQERR02.LOG

touch /var/mqm/errors/AMQERR03.LOG
chown mqm:mqm /var/mqm/errors/AMQERR03.LOG
chmod 666 /var/mqm/errors/AMQERR03.LOG
```

2. For the /var/mqm/qmgrs/@SYSTEM/errors directory:

```
chmod 3777 /var/mqm/qmgrs/@SYSTEM/errors
chown mqm:mqm /var/mqm/qmgrs/@SYSTEM/errors

touch /var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG
chown mqm:mqm /var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG
chmod 666 /var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG

touch /var/mqm/qmgrs/@SYSTEM/errors/AMQERR02.LOG
chown mqm:mqm /var/mqm/qmgrs/@SYSTEM/errors/AMQERR02.LOG
chmod 666 /var/mqm/qmgrs/@SYSTEM/errors/AMQERR02.LOG

touch /var/mqm/qmgrs/@SYSTEM/errors/AMQERR03.LOG
chown mqm:mqm /var/mqm/qmgrs/@SYSTEM/errors/AMQERR03.LOG
chmod 666 /var/mqm/qmgrs/@SYSTEM/errors/AMQERR03.LOG
```

This is part of the procedure to secure the directories and log files needed for WebSphere embedded messaging, as described in the *Securing messaging directories and log files* (tmj_secmqm) topic in the InfoCenter.

If you have installed the embedded messaging server option on top of WebSphere MQ, the MQ command **setmqcap** is set to use parameter 0 instead of -1, which results in:

• Issuing a license-unit message to the MQ console window whenever a queue manager starts

• Writing a message to the MQ error log

To prevent this, after you have completed the installation of IBM WebSphere Application Server, issue the `setmqcap -1` command from a command line.

Using the LaunchPad to start the installation

Use the LaunchPad to access a product overview, the ReadMe file, and installation guides, and to install the product.

Steps for this task

1. Start the LaunchPad.

   On Windows systems, insert the product CD to automatically run the LaunchPad.

   You can also start the LaunchPad manually:

   - On Windows platforms, run the `LaunchPad.bat` command.
   - Mount the CD-ROM drive on a UNIX-based system, if necessary. This procedure varies per platform. Consult your operating system documentation for instructions on mounting and dismounting CD-ROM drives. After accessing data on the CD, run the `LaunchPad.sh` shell script.

   The LaunchPad program is in the `operating-system platform` directory on the product CD.

2. Select a language for the LaunchPad.

   On Windows systems, you might have to move the command window that opens, to reveal the language selection dialog box.

3. Use the LaunchPad to access the product overview, the ReadMe file, and installation guides.

4. Click **Install the product** to launch the installation wizard.

Installing with the installation wizard GUI

Steps for this task

1. **(Optional)** 5.0.1 Prepare the SuSE Linux Enterprise Server 8.0 - Powered by UnitedLinux 1.0 operating platform for WebSphere Application Server product installation.

Operating platform	Tip
SuSE SLES 8.0	Preparing the SuSE Linux Enterprise Server 8.0 - Powered by UnitedLinux 1.0 operating platform for W

2. **(Optional)** 5.0.2 Perform the pre-installation task for IBM WebSphere Application Server, Version 5 installation on Version 9 of the Solaris Operating Environment.

   Perform this step only if you are installing on Solaris 9.

   Click **Support downloads** and limit the downloads search to Version **5.0.2** to locate the article about preparing to install WebSphere Application Server products on Version 9 of the Solaris Operating Environment. The article contains the `mqpreinst.tar` file.

   Download and unpack the `mqpreinst.tar` file from the Support Web site.

   After using the `mqpreinst.sh` script, increase the kernel settings and create the user groups for the embedded messaging feature, as described in step 3.

   When starting the installation in step 5, verify that you are starting from your local copy of the installation image instead of from the product CD-ROM.

3. **(Optional)** Prepare a Linux or UNIX operating platform for the embedded messaging feature.

   Perform this step only if you are installing the WebSphere Application Server Java Message Service (JMS) provider on a Linux or UNIX-based operating system. You must install the embedded messaging feature to use the WebSphere Application Server JMS provider.

   If you are installing the embedded messaging feature, you must create two operating system groups as described in Installing WebSphere embedded messaging as the JMS provider.

The Solaris Operating Environment and HP-UX also require you to increase kernel settings as described in Installing WebSphere embedded messaging as the JMS provider.

For other platform-specific information about using the embedded messaging feature, see Tips for installing the embedded messaging feature.

4. **5.0.1** Download and use the current `prereqChecker.xml` and `prereqChecker.dtd` files.

   The *platform_specific_directory*`/waspc/prereqChecker.xml` file on the product CD-ROM configures prerequisite checking for supported operating systems and required patches.

   5.0.2 This step is not necessary on Solaris V9 if you prepared the operating environment for IBM WebSphere Application Server, Version 5 installation.

   This step is necessary when installing Version 5.0.1 on the following operating systems:

   - Installing WebSphere Application Server 5.0.1 on SuSE Linux SLES 8
   - Installing WebSphere Application Server 5.0.1 on Red Hat Linux 8.0
   - Installing WebSphere Application Server 5.0.1 on AIX 5.2

   Earlier versions of the WebSphere Application Server product required you to edit prerequisites checking, to correct situations where operating system patch levels, the Web server software level, or the database level were more recent that what was defined in the `prereq.properties` file. This was necessary before you could continue the installation of the earlier product. Passing the prerequisites test is no longer required to install the WebSphere Application Server product because Version 5 does not store configuration data in a database. It stores the information in XML configuration files on the node. WebSphere Application Server V5 products do not require a database during installation.

   Installing WebSphere Application Server on new operating platforms can produce a ″Not supported platform″ error during the installation. Downloading and using current `prereqChecker.xml` and `prereqChecker.dtd` files removes the error and, more importantly, lets the installation program check the operating platform to verify that all required patches are installed.

   Download the files from the Support page and use the command method to start the installation, as described in the next step.

   Although you can install the product without the current `prereqChecker.xml` file, you risk not knowing of required patches for your operating system.

5. Start the installation.

   The default installation method for all operating platforms but Linux on a zSeries server, is to click **Install the product** on the LaunchPad tool to launch the InstallShield for MultiPlatforms installation wizard. This action launches the installation wizard GUI.

   The LaunchPad on SuSE Linux 390z does not start the installer. To install, use the **install.sh** program located in the `/linuxs390` directory, instead of the `install` executable. The readme documentation that the LaunchPad links to, is the `readme.html` file in the CD root directory. The `readme` directory off the root of the CD has more detailed readme files. Likewise, the *Installation Guide* is in the `/docs` directory off the root of the CD.

   When you install, the current working directory must be the directory where the installer binary is located. This placement is important because the resolution of the class files location is done in the current working directory.

   For example:

   ```
   cd INSTALL_ROOT/

   ./install
   ```

5.0.2 **Or, when installing on Version 9 of the Solaris Operating Environment:**

```
cd /tmp/WebSphereTemp/SUN
/tmp/WebSphereTemp/SUN/install
```

Or, when installing from the product CD-ROM:

```
cd CD_mount_point
```

```
./install
```

Failure to use the correct working directory can cause ISMP errors that abort installing.

Although the installation wizard checks for prerequisite operating system patches, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site at http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html .

You can also open the page by selecting the Prerequisites option from the LaunchPad. Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to the supported version.

For migration, coexistence, and other reasons, the `prereqChecker.xml` file also checks for existing versions of WebSphere Application Server, as well as for existing Version 5 and Version 5 feature installations. (The installation wizard does not migrate WebSphere Application Server - Express automatically, however. You must use the migration tools to perform the migration manually as described in Migrating from WebSphere Application Server - Express.)

You can start the installation wizard from the product CD-ROM, using the command line. The installation program is in the `operating-system platform` directory on the product CD-ROM.

- On Linux for S/390 platforms, run the **./install.sh** command.
- On other Linux platforms and UNIX-based platforms, run the **./install** command.
- On Windows platforms, run the **Install.exe** command.

If you downloaded a current prereqChecker.xml file to a temporary directory, use the osLevelCheckActionBean.configFilePath parameter. For example, use these commands to start the installation from the operating-system platform directory on the product CD-ROM:

- On Linux for S/390 platforms:
  ```
  ./install.sh -W osLevelCheckActionBean.configFilePath="/tmp/prereqChecker.xml"
  ```
- On other Linux platforms and on UNIX-based platforms:
  ```
  ./install -W osLevelCheckActionBean.configFilePath="/tmp/prereqChecker.xml"
  ```
- On Windows platforms:
  ```
  Install.exe -W osLevelCheckActionBean.configFilePath="C:\temp\prereqChecker.xml"
  ```

You can also perform a silent installation using the `-options responsefile` parameter, which causes the installation wizard to read your responses from the options response file, instead of from the interactive GUI. You must customize the response file before installing silently. After customizing the file, you must issue the command to silently install. Silent installation is particularly useful if you install the product often.

The rest of this procedure assumes that you are using the installation wizard. There are corresponding entries in the response file for every prompt that is described as part of the wizard. Review the description of the response file for more information. Comments in the file describe how to customize its options.

6. Select a language for the wizard GUI and click **Next**.

   The installation wizard opens and a welcome page appears.

7. Click **Next** to continue.

   The license agreement appears for you to read. The license displayed during the GUI installation can contain characters that display incorrectly in Japanese. For example, the section labeled **Part 1** does not show the number **1**. These missing characters do not significantly affect the content of the license agreement.

8. Click the radio button beside the **I accept the terms in the license agreement** message if you agree to the license agreement and click **Next** to continue.

   After you accept the licensing terms, the installation wizard checks for prerequisites and for previous versions, which it can migrate, or coexist with.

   As the WebSphere Application Server product version changes, its prerequisites and corequisites change. It is probably necessary to update your database, Web server, Software Development Kit (SDK), and other software.

   The base WebSphere Application Server product simplifies migrating product prerequisites, by providing the option to install a complimentary Web server and SDK on your supported operating system. You can uninstall back-level prerequisites and let the installation wizard install current versions.

   If the wizard finds a previous version of WebSphere Application Server, it prompts you to migrate applications and the configuration from the previous version, or to coexist with it. If it finds more than one previous version, the installation wizard lists them for you to select which one to migrate. The installation wizard does not list WebSphere Application Server - Express as a candidate for automatic migration, however. You must use the migration tools to perform the migration manually as described in Migrating from WebSphere Application Server - Express.

9. Choose whether to migrate applications and the configuration from a previous version, or to coexist with another version, to do both, or to do neither, and click **Next** to continue.

   The first rule of migration is to migrate after you install the Enterprise product, if you are planning to do so:

If you are planning to install these products:		Select the migration option while installing this product:	
Base product	**Enterprise**	**Base product**	**Enterprise**
X		X	
X	X		X

You can also perform a silent migration or configure for coexistence during a silent installation. Refer to ″Installing silently″ for a description of performing a silent installation, including options you can specify.

The migration prompt appears only when the installation wizard detects a previous version. The coexistence prompt appears when the installation wizard detects any other installation, including another V5 installation.

If you choose to coexist, the wizard displays a port selection panel, where you can select non-conflicting ports. For example, you can change the HTTP transport port for coexistence, from 9081 (one more than the default V5 port number) to 9085 or higher, to avoid potential conflicts with port numbers that previous versions of WebSphere Application Server commonly use.

In some cases, such as when installing a non-English version, the installation wizard might not detect a previous version. You can force the migration and coexistence panel to appear, by starting the installation with an option on the **Install.exe** or **install** command.

For example, on UNIX machines, use this command:

```
./install -W previousVersionDetectedBean.previousVersionDetected="true"
```

You can also force the appearance of the coexistence panel to change conflicting port number assignments. For example, the AIX WebSM system management server listens on port 9090 by default. To avoid a conflict with the WebSphere HTTPS Administrative Console Secure Port (HTTP_TRANSPORT_ADMIN) assignment, which is also 9090 by default, force the coexistence panel to appear using this command:

```
./install -W coexistenceOptionsBean.showCoexistence="true"
```

If you choose neither the migration option nor the coexistence option, you can run V5 and the earlier version, but not at the same time. Although it is possible that both might coexist without port conflicts, you can verify that both versions run together by selecting the coexistence option and checking for conflicting port assignments.

For example, if you plan to install IBM HTTP Server into a separate directory than an earlier version, it coexists with the other version instead of upgrading it. Change the **IBM HTTP Server Port** assignment to something other than **80** and the **IBM HTTP Server Admin Port** assignment to something other than 8008, to verify that they do not conflict with the earlier version port assignments. The installation wizard uses the values you enter when it configures the HTTP Server configuration file, `httpd.conf`.

The migration panel lists all known previous releases that it can identify. If you highlight each release that is shown, the **select previous version** text boxes show the location of the previous product. Select the product that you intend to migrate. If you do not see the previous version that you intend to migrate, click **Select previous version** to enter a location and configuration file name if you are migrating a WebSphere Application Server Advanced Edition Single Server Edition, V4.x installation. The **Configuration file** field is valid only for WebSphere Application Server Advanced Edition Single Server Edition, V4.x.

For the other versions of WebSphere Application Server that are supported by migration (V3.5 Standard Edition, V3.5 Extended Edition, and V4.0 Advanced Edition), the `admin.config` file provides the host and port values for the administrative server. If you used a file name other than `admin.config`, manual migration is required instead of migration through installation. You must start the administrative server of the previous version so that the installation wizard can export the configuration from the `admin.config` file.

Although you might select migration at this point in the installation process, the actual migration does not begin until after the V5 installation is complete. At that time, if the WASPreUpgrade tool fails, the installation wizard does not call the WASPostUpgrade tool to complete the migration but instead, displays the `WASPreUpgrade.log` and `WASPostUpgrade.log` log files for you to diagnose the problem. After fixing the problem, such as starting the administrative server of a previous release, you can start the migration again as described in ″Migrating and coexisting″.

10. Choose a type of installation and click **Next**.

    If you use the GUI, you can choose a full installation type, which installs all features, or a custom installation type. The custom installation type lets you select which features to install. You can improve performance by deselecting certain features, as described in **Performance considerations** below.

    - Choose **full install** to install everything you need to run Web applications on your server, including IBM HTTP Server and the embedded messaging feature. The plug-in for IBM HTTP Server is installed and configured in the full installation. Use this option if you are new to WebSphere Application Server and are unsure of what to install.

    - Choose **Custom install** to select installation features. Choose this option if you already know what to install. Choose this type of installation if you must install and configure the plug-in for a Web server other than IBM HTTP Server. The installation wizard installs all plug-ins if you select any of

them, but it only configures selected plug-ins. Also choose this type of installation if you intend to deselect the WebSphere Application Server embedded messaging server because you have already installed the WebSphere MQ V5.3 product, or another JMS provider.

If you choose a full installation, you can skip the next step.

After selecting a full installation, the wizard prompts you to select the directory for the program code. After selecting a custom installation, the wizard displays the list of features.

11. Select features to install and click **Next** to continue, when performing a custom installation.

This step is available only when you choose the custom installation type. A description of each feature appears at the bottom of the panel when you roll the cursor over the feature.

Selecting certain features causes the installation of other prerequisite features. The following table shows this feature relationship.

Features and feature dependencies

If you select this feature:		This feature is also installed:	Feature description in the InfoCenter
Application Server		Admin scripting	These features are the Application Server (trun_svr_conf) run time and its Samples.
Sub-features:	Application Server Samples		
Administration			"Welcome to system administration" (welc_configop)
Sub-features:	Admin scripting		
	Administrative console	Application Server	
		Admin scripting	
Application assembly and deployment tools		Admin scripting	• Assembling applications (uaatt_start) • Deploying and managing applications (trun_appl) • "ANT tasks" (rovr_antapi)
Sub-features:	Application assembly tool		
	Deploy tool	Application Server	
		Admin scripting	
	ANT utilities		
Embedded messaging			• Installing and configuring a JMS provider (tmj_inst0) • Message-driven beans samples (xmb_smp00)
Sub-features:	Server and client	Application Server	
	Client only		
	Messaging-driven bean Samples	Application Server samples	
		Admin scripting	
IBM HTTP Server V1.3.26.x			Preparing to install and configure a Web server
Web Server Plugins			Preparing to install and configure a Web server
Sub-features:	IBM HTTP Server		
	Apache Web Server		
	Microsoft Internet Information Services (IIS)		
	Sun ONE (iPlanet) Web Server		
	Lotus Domino Web Server		

If you select this feature:		This feature is also installed:	Feature description in the InfoCenter
Performance and analysis tools			• Tivoli Performance Viewer features (uprf_rtpvgui)
Sub-features:	Tivoli Performance Viewer		
	Dynamic cache monitor		• Displaying cache information (tprf_servletmonitor)
	Performance servlet		• Performance monitoring infrastructure servlet (cprf_servlet)
	Log analyzer		• Log analyzer (ctrb_jfla.xml)
Javadocs			IBM WebSphere Application Server, Release 5 API specification in Javadoc format

Embedded messaging feature considerations

Refer to the ″Tips for installing the embedded messaging feature″ for important information about installing the embedded messaging feature. In particular, you must create two operating system groups at this time when installing the feature on a UNIX or Linux platform.

Performance considerations

For better performance, either in a development or production environment, it is recommended that you do not install the Samples. By omitting the Samples, you can improve Application Server startup time by 60 percent and save 15 percent of disk space. Moreover, you can potentially save 30 percent of process footprint (based on a maximum heap size of 256MB). Furthermore, if your applications do not use JMS messaging, it is recommended that you do not install embedded messaging. This recommendation is especially true if your system has 256MB or less of physical memory. By not installing the embedded messaging feature, you can save about 36MB of message queuing process memory footprint. Also, you can potentially improve application startup time by another 5 percent, and save an additional 72MB of disk space (for Windows machines).

Web Server plug-in feature considerations

You can install a Web server to work with WebSphere Application Serve as described in Preparing to install and configure a Web server.

You must make only minor changes to the IBM HTTP Server or other Web server configuration file, if you upgrade a previous release of the supported Web server by installing it in the same directory as the previous release. You simply must remove the configuration file entries for the previous release as described in ″Preparing to install and configure a Web server″.

You can get started easily with Secure Sockets Layer (SSL) connections, by making only a few configuration changes as described in the ″Configuring IBM HTTP Server for SSL mutual authentication″ topic in the InfoCenter. If you run the IBM HTTP Server on the Windows operating systems, you can easily configure the fast response cache accelerator to boost performance. You can also make many other configuration changes with Apache directives. Refer to the InfoCenter for IBM HTTP Server, for a description of configuring the Web server for SSL, the fast response cache accelerator, or Apache directives.

Adding additional features at a later time or uninstalling features

You can add features at any time, by running the installation wizard again.

If you federated the node into a cell, invoke the **removeNode** command from the `bin` folder in the installation root, before adding additional features. After adding the features, use either the **addNode** command or the deployment manager administrative console, to add the node to the cell configuration again. This action synchronizes the master cell configuration with changes you make to the Application Server configuration during the installation of the new features. An installation wizard panel lets you select whether to add features to the existing installation, or perform a new installation to another directory.

When adding features, previously installed features are checked and grayed out with the term **(Installed)** at the end of the feature name.

You can run the uninstall program to remove all installed features.

12. Specify a destination directory. Click **Next** to continue.

   Specify a target directory for the base WebSphere Application Server product, and for the IBM HTTP Server and embedded messaging features, if you are installing them. On UNIX and Linux platforms, you cannot change the default installation directory for the embedded messaging feature.

   Deleting the default target location and leaving an installation directory field empty stops you from continuing the installation process. The installation wizard does not allow you to proceed when you click **Next**. Enter the required target directory to proceed to the next panel.

   When installing on a Solaris system, do not use a double-byte character set (DBCS) name for the directory, or a name with spaces in it.

   Avoid a space in the name of the destination directory when installing on Windows NT, to avoid possible problems when migrating.

   Verify that there is adequate space available in the target directory. Also verify that there is approximately 100 MB of free space in the platform `tmp` directory, or on the disk with the Windows `temp` directory. The actual space required depends on the operating system and the features you are installing. The WebSphere Application Server and WebSphere Application Server Network Deployment products each require a minimum of 35 MB free space in the `tmp` directory. If you have problems accessing the administrative console after installation, check the `installAdminConsole.log` file for a failure indication. You can clean up the `/tmp` space and reinstall the administrative console using the **wsadmin** scripting facility.

   If you must increase the /tmp allocation on a Linux or UNIX-based platform, stop the installation program, increase the allocation, and restart the installation.

   To migrate applications from an earlier version, verify that there is enough space for application objects. As a rough guideline, plan for space equal to 110 percent of the size of the application objects:

   - **For V3.5:** The size of application JAR, WAR, and servlet files.
   - **For V4.0:** The size of enterprise application EAR files

   If you install the IBM HTTP Server product in the same directory as a previous version that works with WebSphere Application Server V4.0.x, follow the instructions to allow WebSphere Application Server V4.0.x and V5 to share the new IBM HTTP Server.

   If you select the embedded messaging feature and there are missing prerequisites, the installation wizard displays the `mq_prereq.log` error log and takes you back to the installation type panel. Choose **Custom** installation and deselect the embedded messaging feature to continue. The `mq_prereq.log` file is in the system temp directory.

13. Specify target directories for configuration files for any selected Web server plug-ins. Click **Next** to continue.

   If you are installing the IBM HTTP Server, you do not have to specify a location for its plug-in configuration file. The wizard uses the installation path you specified for the Web server to derive the location. If you have previously installed the IBM HTTP Server product on the same machine as the

WebSphere Application Server, and are now installing just the plug-in, enter a configuration file location, *IHS_DIR* `/conf/httpd.conf`, where *IHS_DIR* is the directory where the IBM HTTP Server product is installed.

- Verify there is adequate space available in the target directory. On an AIX system, also verify that there is adequate space in the `/tmp` directory.
- When installing on a Solaris system, do not use a double-byte character set (DBCS) name for a directory, or a name with spaces in it.
- If you install a plug-in for an existing Web server installation, specify a path that matches the path of the Web server.

14. Specify node information and click **Next**.

Specify the node name and host name. Although the wizard inserts the machine name (of the installation platform) as the node name, you can specify any unique name. The node name is an arbitrary WebSphere Application Server-specific name that must be unique.

The host name is the network name for the physical machine on which the node is installed. It must resolve to a physical network node on the server. When there are multiple network cards in the server, for example, the host name or IP address must resolve to one of the network cards. Remote WebSphere Application Server nodes use the host name to connect to, and communicate with, this node. It is extremely important to select a host name that other machines can reach within your network.

The value you use for the host name can be the fully qualified DNS hostname, the short host name, or even a numeric IP address. A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node you name with a numeric IP address without DNS being available. The disadvantage is that the numeric IP address is fixed. You must change this setting in the configuration whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. You also cannot use the node if the numeric IP host name is disconnected from the network.

A fully qualified domain hostname is usually the best choice. Advantages are that such host names are flexible enough to support DHCP systems and can run disconnected from the network. Remote nodes usually resolve the address. A disadvantage is that a remote node depends on DNS availability to connect to either a fully qualified or a short domain host name.

15. **(Optional)** Create a Windows service, or plan to create a UNIX monitored process. Click **Next**.

Run WebSphere Application Server and IBM HTTP Server as services on Windows systems only by clicking the check boxes. Clicking a check box on this panel configures a manually started service. You can also create UNIX monitored processes after the installation is complete.

A Windows user ID with spaces cannot be validated for installing services.

When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. The user is not allowed to proceed with the installation.

To work around this problem, install with a user ID that does not contain spaces, or do not choose to install services.

Processes started by a **startServer** command are not running as monitored processes, regardless of how you have configured them.

For example, you can configure a base Application Server as a WebSphere Application Server Windows service. However, if you start the Application Server instance using the **startServer** command, Windows does not monitor or restart the Application Server because it was not started as a Windows service. The same is true on UNIX-based platforms. You must start the server process with a shell script based on the example `rc.was` file, to have the server running as a monitored process.

To perform this installation task, your local Windows user ID must belong to the Administrator group and have the advanced user rights *Act as part of the operating system* and *Log on as a service*. (If

your Windows user ID belongs to the Administrator group, the installation wizard grants it the advanced user rights.) You can also create other Windows services after the installation is complete, to start other server processes.

Although the installation wizard creates Windows services only, you can create a shell script that performs a similar function on UNIX-based platforms. Use the `rc.was` example shell script, in the `install_root/bin` directory, to create the UNIX equivalent of a Windows service. Create a new shell script for each process that the UNIX-based system is to monitor and restart. Edit the `inittab` table of the operating system, to add an entry for each shell script you have created. This causes the UNIX-based system to call each shell script whenever the system reboots.

Similar to a Windows service, each shell script monitors and restarts an individual WebSphere Application Server server process in a stand-alone environment. Comments in the header of the `rc.was` file provide instructions for identifying a WebSphere Application Server process, and show a sample `inittab` entry line for adding each copy of the script.

16. Review the summary information and click **Next** to install the product code or **Back** to change your specifications.

When the installation is complete, the wizard displays the `install_root\logs\mq_install.log` installation log if you selected the embedded messaging feature and there are errors with its installation.

17. Review the `mq_install.log` installation log if it appears. Click **Next** to continue.

The wizard displays the registration panel.

18. Click **Next** to register the product, or deselect the check box and click **Next** to register at a later time.

The installation wizard starts the First Steps tool.

19. Click **Finish** to close the installation wizard.

The installation wizard configures the product. It is not necessary to perform further configuration at this time.

Uninstalling and reinstalling: If you must uninstall the product, follow the steps in ″Uninstalling WebSphere Application Server″. After uninstalling WebSphere Application Server, reinstalling into the same directory without first deleting all directory contents, results in invalid XML configurations because of the possibility of retaining some old files. To delete all files so that you can reinstall with a clean system, perform a manual uninstall as described in ″Uninstalling WebSphere Application Server″.

Installing silently

Use this procedure to perform a silent installation. A silent installation uses the installation wizard program to install the product. However, instead of displaying a user interface, the silent installation causes the installation wizard to interact with you by reading all of your responses from a file that you must customize.

Steps for this task

1. Verify that the user ID that you are using to run the silent installation, has sufficient authority to perform the task.

The silent installation fails if the user ID does not have all authorizations required to install each option you select. Having the proper authorizations primarily affects setting up services on Windows, but also affects setting up the embedded messaging feature on Windows and UNIX-based operating platforms.

If you attempt a silent installation of services or the embedded messaging feature without the required authorizations, you might see messages in the log.txt file that are similar to these:

```
(Jan 30, 2003 12:04:32 PM), Setup.product.install,
com.ibm.ws.install.conditions.PCWebSphereCheckCondition, msg1,
higherVersionFoundPasses: false
(Jan 30, 2003 12:04:32 PM), Setup.product.install,
com.ibm.ws.install.conditions.PCWebSphereCheckCondition, msg1,
defaultEvalResult: false
(Jan 30, 2003 12:04:32 PM), Setup.product.install,
```

```
com.ibm.ws.install.conditions.PCWebSphereCheckCondition, msg1,
checkOnlyCurrentVersion: true
(Jan 30, 2003 12:04:34 PM), Setup.product.install,
com.ibm.ws.install.conditions.PCWebSphereCheckCondition, msg1,
higherVersionFoundPasses: false
(Jan 30, 2003 12:04:34 PM), Setup.product.install,
com.ibm.ws.install.conditions.PCWebSphereCheckCondition, msg1,
defaultEvalResult: false
(Jan 30, 2003 12:04:34 PM), Setup.product.install,
com.ibm.ws.install.conditions.PCWebSphereCheckCondition, msg1,
checkOnlyCurrentVersion: false
(Jan 30, 2003 12:04:41 PM), Setup.product.install,
com.ibm.ws.install.actions.LogMessageAction, msg1, INST0058E:
The install failed or did not complete due to one or more errors.
```

You must install as root on a UNIX-based operating platform. On a Windows operating platform, give the user ID these authorizations:

- Assign the user ID to the Administrator group.
- Give the user ID the advanced user right, *Act as part of the operating system*.
- Give the user ID the advanced user right, *Log on as a service*.

The installation wizard GUI grants your Windows user ID the advanced user rights, if the user ID belongs to the Administrator group. The silent installation does not. If you create a new user ID on a Windows platform to perform the silent installation, you must restart the system to activate the proper authorizations for the user ID, and to perform a successful silent installation.

2. Copy the response file as myoptionsfile and customize the copy, to preserve the original response file.

   The name of the original file is `responsefile`.

3. Issue one of these commands to use your custom response file:

   Windows
   ```
   Install.exe -options myoptionsfile
   ```

   Linux for S/390 platforms
   ```
   ./install.sh -options ./myoptionsfile
   ```

   Other Linux and UNIX-based platforms
   ```
   ./install -options ./myoptionsfile
   ```

   You can find the sample options response file in the `operating-system platform` directory on the product CD.

4. **(Optional)** Reboot your machine in response to the prompt that might appear on Windows platforms.

   If you install the embedded messaging feature, `-P mqSeriesBean.active="true"`, certain conditions, such as a locked file, might require you to reboot. You have the option of rebooting immediately, after which the installation program resumes the installation at the point it left off. You can also defer rebooting to a convenient time, such as after the overall installation is complete.

You can install silently using the response file. To determine if the silent installation is successful, examine the `log.txt` file for a line similar to this:

```
MessageAction, msg1, The WebSphere 5.0 install is complete.
```

The message is an indicator that you have successfully installed the product.

Customizing the base options response file
Before using the **Install.exe -options myoptionsfile** command, for example, to invoke a silent installation, you must customize the response file to add your selections.

Customize the options response file precisely to let the installation program read the option values it contains.

Steps for this task

1. Locate the sample options response file. The file is named *responsefile* in the `operating-system` `platform` directory on the product CD-ROM.

2. Copy the file to preserve it in its original form. For example, copy it as *myoptionsfile*.

3. Edit the copy in your flat file editor of choice, on the target operating system. Read the directions within the response file to choose appropriate values.

   To prepare the file for a silent installation on AIX, use UNIX line-end characters (0x0D0A) to terminate each line of the options response file as described in Platform-specific tips for installing and migrating.

   Here is an example of an edited entry for a feature (`mqSeriesBean`) and its sub-features (`mqSeriesServerBean` and `mqSeriesClientBean`). These entries install both the server and the client. The server option installs both subfeatures in this case as you can see from the comments in the file. These options settings do not install the Samples. If you set the client option to true, this is an error. If you have errors in the response file, the installation program logs the errors in `log.txt` and exits without installing anything.

```
# *********
# Install Embedded Messaging
# *********
-P mqSeriesBean.active="true"
# *********
# The next three features are for Embedded Messaging.  In order to install
# any of the following three subfeatures, the property to install Embedded
# Messaging denoted above must be set to "true."
# *********
# Install Embedded Messaging Server and Client
# *********
-P mqSeriesServerBean.active="true"
# *********
# Embedded Messaging Server and Client install location
# *********
# If you choose to install Embedded Messaging Server and Client above, please
# specify an install location below for Windows platforms only.
# The directory may not be configured by the user for UNIX platforms
# as it is predetermined.
# *********
-P mqSeriesServerBean.installLocation="C:\Program Files\IBM\WebSphere MQ"
# *********
# Install Embedded Messaging Client only
# *********
-P mqSeriesClientBean.active="false"
# *********
# Embedded Messaging Client Only install location
# *********
# If you choose to install Embedded Messaging Client only above, please
# specify an install location below for Windows platforms only.
# The directory may not be configured by the user for UNIX platforms
# as it is predetermined.
# *********
-P mqSeriesClientBean.installLocation="C:\Program Files\IBM\WebSphere MQ"
# *********
# Install Message-driven beans Samples
# *********
-P mqSeriesSamplesBean.active="false"
```

4. Make the first non-commented option **-silent** to have a silent install.

5. Include custom option responses that reflect parameters for your system.

   Read the directions within the response file to choose appropriate values.

6. Save the file.

You now understand how to customize the response file to prepare for running a silent installation.

Usage scenario

Edit the version of the file that ships with the WebSphere Application Server product. The following example shows a few lines from the file.

```
-W setupTypes.selectedSetupTypeId="custom"


# ********************************************
# Below is the beginning of the response file that needs to be
# filled in by the user.
# ********************************************

# ********************************************
# The below value specifies silent install. This value
# indicates that the install will be silent. If you wish not to install
# silently, just delete this value.
# ********************************************

-silent


# ********************************************
# WebSphere Application Server Install Location
#
# Please specify the destination directory for the WebSphere Application
# Server installation.   You will need to change this for UNIX
# platforms. As an example for AIX, the value may be
# "/usr/WebSphere/AppServer"
# ********************************************

-P wasBean.installLocation="C:\Program Files\WebSphere\AppServer"
```

responsefile

Silent installation is an installation method that reads all choices from the options response file, `responsefile`, which you must customize before using.

Location

The sample options response file, `responsefile`, is on the product CD in the `operating-system platform` directory.

Usage notes

- This file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as WordPad on a Windows 2003 platform.
- This file is not updated by any product components.
- The file must exist to perform a silent installation. The installation program reads this file to determine installation option values when you install silently.
- Save the file in a location you can identify when you specify the fully qualified path as part of the installation command.

First Steps tool tips

First Steps is a post-installation ease-of-use tool for directing WebSphere Application Server elements from one place. Options dynamically appear on the First Steps panel, depending on features you install. With all options present, you can use First Steps to start or stop the Application Server, verify the installation, access the InfoCenter, run the Application Assembly Tool, access the administrative console, access the Samples Gallery, or launch the product registration.

First Steps starts automatically at the end of the installation. If it is not running, start First Steps from the command line:

- On Linux and UNIX-based platforms: `install_root/bin/firststeps.sh`

- On Windows platforms: `install_root\bin\firststeps.bat`

After selecting the **Start the server** option, the message area at the bottom of the screen informs you when your server is open and ready for e-business. At the same time, the menu item changes to **Stop the server**.

If this is the first time you have started the server since installing, it is a good idea to click on **Verify installation** to make sure that all is well with your installation. If you verify before starting the server, the verification process starts the server for you.

The Web administrative console is a configuration editor that runs in a Web browser. It lets you work with Application Server configuration files encoded in XML. Changes made using the Web administrative console take effect the next time you start the Application Server. To launch the administrative console, click **Administrative console**. You can also point your browser to http://*your_host_name*:9090/admin to start the administrative console. Substitute your own host name in the address. As the administrative console opens, it prompts you for a login name. This is not a security item, but merely a tag that allows you to identify configuration changes you make during the session.

From the First Steps menu, click on **Samples gallery** to explore the sample applications. Alternatively you can point your browser directly to http://*your_host_name*:9090/WSsamples. This is case sensitive. Substitute your own host name in the address.

Using the installation verification test

After installing the product, you are ready to use the installation verification test (IVT).

The IVT program scans product log files for errors and verifies core functionality of the product installation.

Steps for this task

1. If you started the administrative server of a V3.5.x and later, or V4.0.x and later WebSphere Application Server product to let the installation wizard export its configuration and applications to V5, stop the server before running the IVT.

   Otherwise, the V5 server might not start due to port conflicts between the two servers.

2. Select **Verify Installation** from the First Steps panel and observe the results in the First Steps status window. A log file for the installation is created and stored in the `logs` directory with the name of `ivt.log`.

3. If the First Steps tool is not running, start it from the `bin` directory.
   - On Windows platforms: firststeps.bat
   - On UNIX-based platforms: ./firststeps.sh

   You can also run First Steps from the Windows Start Menu.

4. You can also start the IVT tool from the `bin` directory.
   - On Windows platforms: ivt
   - On UNIX-based platforms: ./ivt.sh

   The IVT program starts the server automatically if the server is not running. Once the server initializes, the IVT runs a series of verification tests and reports pass or fail status in a console window. It also logs results to the *install_root* `\logs\ivt.log` file.

Troubleshooting the installation

If you did not receive the *Successful installation* message, troubleshoot the installation, as described below.

Steps for this task

1. Use the First Steps tool to run the installation verification test (IVT). Check the *install_root* \logs\ivt.log file for a summary of test results. Correct any errors and retry.

   The installation wizard automatically starts the First Steps tool at the end of installation.

2. Check the installation log files for errors after installing:

   During installation, a single entry in the log.txt file points to the temporary log file, either %TEMP%\log.txt on Windows, or /tmp/log.txt on UNIX-based platforms. The installation program copies the file to the location shown at the end of the installation.

Table 9. Installation log locations when installing WebSphere Application Server

Component	Installation log path name	
	Windows system temp directory	**UNIX-based operating system: /tmp/**
Embedded messaging feature (based on WebSphere MQ) prerequisites error log	mq_prereq.log	
Component	**Installation log path name**	
	Windows: *install_root*\logs	**UNIX-based operating system: *install_root*/logs/**
WebSphere Application Server	log.txt	
IBM HTTP Server	ihs_log.txt	
Embedded messaging feature (based on WebSphere MQ) installation log	mq_install.log	
Embedded messaging feature (based on WebSphere MQ) configuration log	createMQ.*node*.server1.log	
Default Application	installDefaultApplication.log	
Samples Gallery	installSamples.log	
Administrative console	installAdminConsole.log	
Migration tools	WASPostUpgrade.log	

3. Turn on tracing if the installation logs do not contain enough information to determine the cause of the problem.

   - Report the stdout and stderr logs to the console window, by adding the -is:javaconsole parameter to the **Install.exe** or **install** command:
     - On Windows platforms:

       ```
       Install.exe -is:javaconsole
       ```

       Or capture it to a file with:

       ```
       Install.exe -is:javaconsole > drive:\captureFileName.txt
       ```
     - On Linux for S/390 platforms:

       ```
       install.sh -is:javaconsole
       ```

       Or capture it to a file with:

       ```
install.sh -is:javaconsole > captureFileName.txt 2>&1
       ```
     - On other Linux platforms and UNIX-based operating platforms:

       ```
       install -is:javaconsole
       ```

       Or capture it to a file with:

       ```
       install -is:javaconsole > captureFileName.txt 2>&1
       ```
   - Capture additional information to a log of your choice with the -is:log filename option.

- Turn on additional installation logging by passing the `-W Setup.product.install.logAllEvents="true"` parameter to the **Install.exe**, **install.sh**, or **install** command:
  - On Windows platforms:

    `Install.exe -W Setup.product.install.logAllEvents="true"`
  - On Linux for S/390 platforms:

    `install.sh -W Setup.product.install.logAllEvents="true"`
  - On other Linux platforms and UNIX-based operating platforms:

    `install -W Setup.product.install.logAllEvents="true"`

If you install on an AIX 5.1 system, with maintenance level 2, it is possible that the Web-based system manager, a standard component of AIX systems, already uses port 9090. When starting the server you get information that port 9090 is already in use. To resolve the conflicting port use, change the port assignment for the HTTP_TRANSPORT_ADMIN port in the `server.xml` file. The file path is:

```
usr/websphere/appserver/config/cells
            /cell/nodes/node/servers/server1/server.xml
```

4. Use the First Steps tool or the command line method to start the Application Server.

   To start the server from the command line:
   - On UNIX-based server platforms: *install_root*/bin/startServer.sh server1
   - On Windows platforms: *install_root*\bin\startServer server1

5. Verify whether the server starts and loads properly, by looking for a running Java process and the *Open for e-business* message in the `SystemOut.log` and `SystemErr.log` files. If there is no Java process or the message does not appear, examine the same logs for any miscellaneous errors. Correct any errors and retry.

   You can find the `SystemOut.log` and `SystemErr.log` files in the `install_root\logs\server1` (Windows) or `install_root/logs/server1` (UNIX-based) directory.

6. Refer to the plug-in configuration documentation, if you have installed plug-ins and the Web server does not come up properly.

7. Start the Snoop servlet.
   a. Point your browser to URL: `http://localhost:9090/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://localhost/snoop` to test the Web server plug-in.
   b. Start the Application Server.
   c. Start the IBM HTTP Server or the Web server you are using.

      Use a command window to change directories to the IBM HTTP Server installed image, or to the installed image of your Web server. Issue the appropriate command to start the Web server, such as these commands for IBM HTTP Server:

      To start the IBM HTTP Server from the command line:
      - On Windows platforms: **apache**
      - On UNIX-based server platforms: **./apachectl start**
   d. Verify that Snoop is running.

      Either URL should display the **Snoop Servlet - Request/Client Information** page.

8. Start the WebSphere Application Server administrative console.
   a. Start the Application Server.
   b. Point your browser to URL: `http://localhost:9090/admin`.
   c. Type any ID and click **OK** at the administrative console window.

   The server starts. The administrative console starts. You can access the administrative console through the browser. The administrative console accepts your login.

9. Resolve any IP address caching problems.

   By default, the Java 2 SDK caches the IP address for the DNS naming lookup. After resolving the host name successfully, the IP address stays in the cache. By default, the cache entry remains forever. This default IP caching mechanism can cause problems, as described below.

   Problem scenario 1

   Suppose the Application Server at host1.ibm.com has an initial IP address of 1.2.3.4. When a client at host2.ibm.com conducts a DNS lookup of host1.ibm.com, it stores the 1.2.3.4 address in the cache. Subsequent DNS name lookups return the cached value, 1.2.3.4. The cached value is not a problem until the host1.ibm.com IP address changes, to 5.6.7.8, for example. The client at host2.ibm.com does not retrieve the current IP address but always retrieves the previous address from the cache. If this scenario occurs, the client cannot reach host1.ibm.com unless you stop and restart the client process.

   Problem scenario 2

   Suppose the Application Server at host1.ibm.com has an initial IP address of 1.2.4.5. Although the IP address of the Application Server does not change, a network outage can record an exception code as the IP address in the cache, where it remains until the client is restarted on a working network. For example, if the client at host2.ibm.com disconnects from the network because of an unplugged cable, the disconnected lookup of the Application Server at host1.ibm.com fails. The failure causes the IBM Developer Kit to put the special exception code entry into the IP address cache. Subsequent DNS name lookups return the exception code, which is java.net.UnknowHostException.

   Using the IP address cache setting

   You can always stop and restart a deployment manager process to refresh its IP address cache. However, this might be expensive or inappropriate.

   The networkaddress.cache.ttl (public, JDK1.4) and sun.net.inetaddr.ttl (private, JDK1.3) parameters control IP caching. The value is an integer that specifies the number of seconds to cache IP addresses. The default value, -1, specifies to cache forever. A value of 0 specifies to never cache.

   Using a zero value is not recommended for normal operation. If you do not anticipate network outages or changes in IP addresses, use the cache forever setting. Never caching introduces the potential for DNS spoofing attacks.

   For more information

   The Java 2 SDK, Standard Edition 1.4 Web site describes the private sun.net.inetaddr.ttl property, which also works in Java 2 SDK, Standard Edition 1.3. The *Networking* section of the Java 2 SDK, Standard Edition 1.4 Web site describes a change in the behavior of the java.net.URLConnection class.

You can troubleshoot the installation.

Migrating and coexisting

Determine whether you have an existing version of WebSphere Application Server installed on the machine where you plan to install your V5 product.

If you have a previous version, you must plan whether to copy the configuration and applications of the previous version to the new version, which is *migration*. Migration does not uninstall the previous version. The earlier release is still functional. If you run it at the same time as the V5 installation, the two versions are said to be *coexisting*. You can choose to provide non-default port assignments for V5 to support coexistence, by selecting the **coexistence** option during installation.

WebSphere Application Server contains migration tools that provide all migration functionality. The installation wizard can call the migration tools, or you can call them manually at a later time. The migration tools migrate applications and configuration information to the new version, as described in detail in the *Migrating* (welc_migrating) topic in the InfoCenter and in Configuration mapping during migration.

You can also find information about migrating the configuration and applications from a previous version of WebSphere Application Server to V5 in the IBM Redbook, Migrating to WebSphere V5.0: An End-to-End Migration Guide, SG24-6910-00.

This section contains the following topics:
- Overview of migration and coexistence
- Defining coexisting port definitions
- Developing a strategy for migration and coexistence
- Configuration mapping during migration
- Migrating administrative configurations manually
    - Migrating from an unsupported operating system
    - WASPreUpgrade command
    - WASPostUpgrade command
- Configuring WebSphere Application Server after migration
- Coexistence support
- Setting up V3.5.x and V5 coexistence
- Setting up V4.0.x and V5 coexistence
- Setting up V5 coexistence
- Port number settings in WebSphere Application Server versions

Overview of migration and coexistence

The migration tools perform a fairly routine migration from V4 to V5. For example, Java 2 Platform, Enterprise Edition (J2EE) 1.2 enterprise archive (EAR) files in V4 work in V5 of WebSphere Application Server, which also supports J2EE 1.3. Similarly, it is not necessary to redeploy enterprise Java bean (EJB) 1.1 Java archive (JAR) files when moving them from V4 to V5, which also supports EJB 2.0 JAR files.

The migration from V3.5 to V5 involves significant changes in application structures, development, and deployment. The migration tools assist in this transition by migrating system configurations and creating J2EE artifacts, including mapping previous security settings to J2EE security roles. These security mappings let you access migrated assets during the transition. The migration tools create initial J2EE enterprise applications based on V3.5.x configurations. However, because of the significant changes in the application structures, carefully test and fine tune migrated applications using development and deployment tools.

There are four combinations of migration and coexistence that you can select from the installation wizard or during silent installation:
- Migrate only
- Coexist only
- Migrate and coexist
- Neither migrate nor coexist

If you neither migrate nor coexist with an earlier version of WebSphere Application Server, you are choosing to ignore the previous installation. You can run only one version at a time because of conflicting default port assignments, although it is possible that both versions might run at the same time without conflict if you use non-default ports in the earlier version. To resolve conflicting port assignments, the coexistence panel lets you manually assign ports for V5 to verify that it can run with an earlier version.

Defining coexisting port definitions

You can specify port assignments for coexistence on the installation wizard coexistence panel, by editing configuration files manually, by wsadmin scripting, or by using the **Servers > Application Servers > server1 > End Points** administrative console page.

WebSphere Application Server products use a different set of default port numbers for coexistence than for the initial installation.

Port names, default value, and description	Value
• Installation wizard name: **HTTP Transport Port** • Name in base WebSphere Application Server server.xml configuration file: HTTPTransport_1 EndPoint_1 • virtualhosts.xml: HostAlias_1 • Silent installation response file option name: coexistencePanelBean.httpTransportPort • Base WebSphere Application Server, default value: 9080 • InfoCenter description: Port used for request queues between a WebSphere Application Server plug-in for Web servers and a Web container in which the Web modules of an application reside (crun_transport)	9085

Port names, default value, and description	Value
• Installation wizard name: **HTTPS Transport Port** • Name in base WebSphere Application Server server.xml configuration file: HTTPTransport_2 EndPoint_2 • virtualhosts.xml: HostAlias_3 • Silent installation response file option name: coexistencePanelBean.httpsTransportPort • Base WebSphere Application Server, default value: 9443 • InfoCenter description: Port used for request queues between a WebSphere Application Server plug-in for Web servers and a Web container in which the Web modules of an application reside (crun_transport)	9444

Port names, default value, and description	Value
• Installation wizard name: **HTTP Admin Console Port** • Name in base WebSphere Application Server server.xml configuration file: HTTPTransport_3 EndPoint_3 • virtualhosts.xml: HostAlias_4 • Silent installation response file option name: coexistencePanelBean.adminConsolePort • Silent installation response file option name: PME_CoexistenceInformationPanel.adminConsolePort • Default value: 9090 • InfoCenter description: Port used for request queues between a WebSphere Application Server plug-in for Web servers and a Web container in which the Web modules of an application (including the administration console application) reside (crun_transport)	9091

Port names, default value, and description	Value
• Installation wizard name: **HTTPS Admin Console Secure Port** • Name in base WebSphere Application Server server.xml configuration file: HTTPTransport_4 EndPoint_4 • virtualhosts.xml: HostAlias_5 • Silent installation response file option name: coexistencePanelBean.secureAdminConsolePort • Default value: 9043 • InfoCenter description: Port used for request queues between a WebSphere Application Server plug-in for Web servers and a Web container in which the Web modules of an application (including the administration console application) reside (crun_transport)	9044

Port names, default value, and description	Value
• Installation wizard name: **Bootstrap port** • Name in base WebSphere Application Server serverindex.xml configuration file: BOOTSTRAP_ADDRESS EndPoint_1 • Silent installation response file option name: coexistencePanelBean.bootstrapPort • Base WebSphere Application Server, default value: 2809 • InfoCenter description: The bootstrap port setting together with the host name forms the initial context for JNDI references. (tnam_develop_naming)	2810

• You can use the connector type and port number parameters of certain commands to identify the bootstrap port number you specify for the RMI connector address in a coexistence scenario. For example, to use the **wsadmin** command in a coexistence environment, you can specify either of the following commands:
 	
```wsadmin.bat/sh -conntype RMI -port portNumber```
```wsadmin.bat/sh -conntype SOAP -port portNumber```
 	
Alternatively, you can update the wsadmin.properties file located in the install_root/properties directory to contain the new port values. If you do this, you need not specify the port parameter when using the wsadmin tool.
• The SOAP connection type uses the SOAP connector address port number described next.	

Port names, default value, and description	Value
• Installation wizard name: **SOAP Connector Address** • Name in base WebSphere Application Server serverindex.xml configuration file: SOAP_CONNECTOR-ADDRESS EndPoint_2 • Silent installation response file option name: coexistencePanelBean.soapConnectorAddress • Default value: 8880 • InfoCenter description: Port used for the Simple Object Access Protocol (SOAP) connector (uxml_rconnector)	8881

• You can use the connector type and port number parameters of certain commands to identify the port number you specify for the SOAP connector address in a coexistence scenario. For example, to use the **wsadmin** command in a coexistence environment, you can specify either of the following commands:
 	
```wsadmin.bat/sh -conntype RMI -port portNumber```
```wsadmin.bat/sh -conntype SOAP -port portNumber```
 	
Alternatively, you can update the wsadmin.properties file located in the install_root/properties directory to contain the new port values. If you do this, you need not specify the port parameter when using the wsadmin tool.
• The remote method invocation (RMI) connection type uses the bootstrap port number described previously.	

Port names, default value, and description	Value
• Installation wizard name: **DRS client address** • Name in serverindex.xml configuration file: DRS_CLIENT_ADDRESS EndPoint_3 • Silent installation response file option name: coexistencePanelBean.drsClientAddress • Default value: 7873 • InfoCenter description: Port used to configure the Data Replication Service (DRS), which is a JMS-based message broker system for dynamic caching (rxml_portnumber)	7874

Port names, default value, and description	Value
• Installation wizard name: **JMS server queued address** • Name in base WebSphere Application Server serverindex.xml configuration file: JMSSERVER_QUEUED_ADDRESS EndPoint_4 • Silent installation response file option name: coexistencePanelBean.jmsServerQueuedAddress • Default value: 5558 • InfoCenter description: Port used to configure the WebSphere JMS provider topic connection factory settings (umj_ptcfw)	5568

Port names, default value, and description	Value
• Installation wizard name: **JMS server security port** • Name in base WebSphere Application Server server.xml configuration file: JMSServer_1 Address_3 • Silent installation response file option name: coexistencePanelBean.jmsServerSecurityPort • Default value: 5557 • InfoCenter description: Port used in JMS security processing (cm_secty)	5567

Port names, default value, and description	Value
• Installation wizard name: **JMS Server Direct Address** • Name in base WebSphere Application Server serverindex.xml configuration file: JMSSERVER_DIRECT_ADDRESS EndPoint_5 • Silent installation response file option name: coexistencePanelBean.jmsServerDirectAddress • Default value: 5559 • InfoCenter description: Port used to configure the WebSphere JMS provider topic connection factory settings (umj_ptcfw)	5569

Port names, default value, and description	Value
• Installation wizard name: **SAS SSL ServerAuth Address** • Name in serverindex.xml configuration file: SAS_SSL_SERVERAUTH_LISTENER_ADDRESS EndPoint_6 • Silent installation response file option name: coexistencePanelBean.sasSSLServerAuthAddr • Base WebSphere Application Server, default value: 0; With security, assign your own non-conflicting value: *nnnn* • InfoCenter description: Port used to listen for requests to use the server (tsec_inboundtransport)	0
Assign a coexistence port other than 0, which dynamically takes a free port. You must have a fixed port that clients can use to find the SAS SSL ServerAuth Address. There is no mechanism for clients to use to find a dynamically assigned port.	

Port names, default value, and description	Value
• Installation wizard name: **CSIV2 ServerAuth Listener Address** • Name in base WebSphere Application Server serverindex.xml configuration file: CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS EndPoint_7 • Silent installation response file option name: coexistencePanelBean.csivServerAuthListenerAddr • Base WebSphere Application Server, default value: 0; With security, assign your own non-conflicting value: *nnnn* • Network Deployment, default value: 9403 but is not enabled until you enable security • InfoCenter description: Port used to listen for requests to use the server (tsec_inboundtransport)	0
Assign a coexistence port other than 0, which dynamically takes a free port. You must have a fixed port that clients can use to find the CSIV2 ServerAuth Listener Address. There is no mechanism for clients to use to find a dynamically assigned port.	

Port names, default value, and description	Value
• Installation wizard name: **CSIV2 MultiAuth Listener Address** • Name in base WebSphere Application Server serverindex.xml configuration file: CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS EndPoint_8 • Silent installation response file option name: coexistencePanelBean.csivMultiAuthListenerAddr • Base WebSphere Application Server, default value: 0; With security, assign your own non-conflicting value: *nnnn* • Network Deployment, default value: 9402 but is not enabled until you enable security • InfoCenter description: Port used to listen for requests to use the server (tsec_inboundtransport)	0
Assign a coexistence port other than 0, which dynamically takes a free port. You must have a fixed port that clients can use to find the CSIV2 MultiAuth Listener Address. There is no mechanism for clients to use to find a dynamically assigned port.	

Port names, default value, and description	Value
• Installation wizard name: **ORB Listener Port** • Name in base WebSphere Application Server serverindex.xml configuration file: Not applicable • Silent installation response file option name: Not applicable - See note. • Base WebSphere Application Server default value: Not applicable • InfoCenter description: Port used to listen for requests to use the server (tsec_inboundtransport)	9101

Port names, default value, and description	Value
• Base WebSphere Application Server installation wizard name: **IBM HTTP Server Port** • Base WebSphere Application Server virtualhosts.xml: HostAlias_2 • plugin-cfg.xml configuration file name:Virtual Host • Default value: 80 • Silent installation response file option name: coexistencePanelBean.ihsPort • Description: Port used to communicate with the IBM HTTP Server	81

Port names, default value, and description	Value
• Base WebSphere Application Server installation wizard name: **IBM HTTP Server Admin Port** • Network Deployment installation wizard name: Not applicable • plugin-cfg.xml configuration file name: Virtual Host • Silent installation response file option name: coexistencePanelBean.ihsAdminPort • Default value: 8008 • Description: Port used to communicate with the IBM HTTP Administration Server	8009

If you choose to both migrate and coexist, you are selecting two tasks with roughly opposite ends. The goal of migration is to reconstruct your earlier version in a nearly identical V5 environment, including applications, configuration settings, URI descriptors, and Web server context roots. The goal of coexistence is to create an environment that is not in conflict with an earlier version, so far as port settings are concerned. If you select both migration and coexistence, the installation wizard migrates the earlier version configuration and applications. Then the wizard changes port settings to support coexistence with the earlier version, which means that both nodes can start and run at the same time. Migration occurs first. Coexistence processing occurs next and alters the following configuration files to the default settings shown previously:

- The virtualhosts.xml file:
  - HTTP Transport Port
  - IBM HTTP Server Port
  - HTTPS Transport Port
  - HTTP Admin Console Port
  - HTTPS Admin Console Secure Port
- The `serverindex.xml` file:
  - Bootstrap port
  - SOAP Connector Address
  - DRS client address
  - JMS server queued address
  - JMS Server Direct Address
  - SAS SSL ServerAuth Address
  - CSIV2 ServerAuth Listener Address
  - CSIV2 MultiAuth Listener Address
- The `server1\server.xml` file
  - HTTP Transport Port
  - HTTPS Transport Port
  - HTTP Admin Console Port
  - HTTPS Admin Console Secure Port
  - JMS server security port

You must then start server1 and use it to change any ports that are in conflict, so that there is no conflict with the earlier version. Then you can run both versions at the same time.

There are two other problems that might occur when you choose both migration and coexistence:

- One problem is conflicting context roots when attempting to share the same Web server.

  Follow the procedure in Migrating plug-ins, one machine at a time to learn how to configure a Web server for sharing between WebSphere Application Server versions. The topic links to procedures that are specific to Web server products, such as the (to support multiple WebSphere Application Server versions) topic.

- The other problem occurs if you install two or more WebSphere Application Server V5 products on the same machine.

  You must resolve port conflicts as described in Setting up V5 coexistence.

Developing a strategy for migration and coexistence

Coexistence and migration are optional. Use the following procedure to develop an application and configuration migration strategy.

Steps for this task

1. Prepare to migrate or update product prerequisites and corequisites to supported versions.

    Refer to the Supported hardware and software site at http://www.ibm.com/ software/webservers/ appserv/ doc/ latest/ prereq.html site for current requirements.

2. Prepare to use the migration function of the installation wizard, to migrate silently, or to migrate manually.

    - You have the option of letting the installation wizard use the migration tools to migrate an existing WebSphere Application Server earlier version to V5 automatically.

      The installation wizard detects a previously installed version, and displays a migration and coexistence panel. You can select either option, both options, or no option.

      a. Start the administrative server of WebSphere Application Server Standard Edition (V3.5.x) or WebSphere Application Server Advanced Edition (V3.5.x or V4.0.x).

          Starting the administrative server allows the migration tools to use XMLConfig to export the configuration data repository. It is not necessary that you start the administrative server for WebSphere Application Server Advanced Single Server Edition, V4. The migration tools use the XMI configuration files directly.

      b. Select the migration option and click **Next** when using the installation wizard. When using the silent installation method, select the appropriate option.

          Enter a fully qualified path for the backup directory, where the migration tools store and read the saved configuration and other files. During migration, the wizard backs up the old version administrative configuration and user data files, but does not uninstall the old version. The wizard automatically configures the new installation with the previous configuration data. It also copies some application data to the newer version.

          The installation wizard uses the WASPreUpgrade migration tool to export the current administrative configuration. After this phase completes, the wizard runs the second phase, which uses the WASPostUpgrade migration tool to convert the backed-up administrative configuration into the new V5 configuration. You can stop the administrative server of the earlier version after the migration is complete. You must stop the earlier version server before starting the V5 server.

          The installation program prompts you for the following information during migration:

          - Backup directory.
          - Currently installed WebSphere Application Server directory. This directory must differ than the one where you install V5.
          - Name of the administrative node of the previous version. The default is the computer name, but both the previous node name and the V5 node name must be the same.

            To determine the administrative node name of the previous version, you can do either of these actions:

            - Start the administrative console of the earlier release. View the **Nodes** folder to determine the node name for the system you want to migrate.
            - Perform an XMLConfig export from the administrative console. Review the output xml file to look for <node action=″update″ name=″*nodename*″> to determine the node name.

      c. Plan to complete the migration after the installation and migration are complete.

Follow these steps to complete the migration:

1) Stop the administrative server from the earlier version.

2) Load existing applications into your development environment and fix any known problems.

3) Start the servers in the new installation.

4) Deploy the changed applications in a test environment.

5) Test all applications thoroughly.

6) Follow normal test procedures as you move the test environment into production.

The installation wizard backs up the earlier version but does not uninstall it, exports the current configuration, and migrates the configuration to the new installation. If there are errors during the WASPreUpgrade phase, such as the administrative server from the earlier version is not running, the installation wizard does not run the WASPostUpgrade phase. The wizard displays the WASPreUpgrade.log and WASPostUpgrade.log log files if there are errors.

- You can also install the base WebSphere Application Server product silently. Specify non-conflicting port assignments in the silent options response file to silently configure for coexistence during the silent installation. You can also migrate an earlier version configuration silently. You must supply values for silent migration options when you customize the options response file:

  a. Direct the installation program to migrate a previous installation by specifying the -W previousVersionDetectedBean.previousVersionDetected=″true″ parameter.

  b. Use one of the following values to identify the specific version to migrate:
     – For WebSphere Application Server Advanced Edition, V3.x and V4.0.x, use one of the following parameters:
        - -W previousVersionPanelBean.selectedVersionEdition=″AE″
        - -W previousVersionPanelBean.selectedVersionEdition=″advanced″
     – For WebSphere Application Server Advanced Single Server Edition, V4.0.x, use the -W previousVersionPanelBean.selectedVersionEdition=″AEs″ parameter.
     – For WebSphere Application Server Standard Edition, V3.x, use the -W previousVersionPanelBean.selectedVersionEdition=″standard″ parameter.

  c. Specify the location where the previous version is installed in the -W previousVersionPanelBean.selectedVersionInstallLocation=″ /opt/WebSphere/AppServer″ parameter.

  d. Specify the path to the configuration file:
     – For WebSphere Application Server Advanced Edition, V3.x and V4.0.x, use the -W previousVersionPanelBean.selectedVersionConfigFile=″ /opt/WebSphere/AppServer/config/admin.config″ parameter.
     – For WebSphere Application Server Advanced Single Server Edition, V4.0.x, use the -W previousVersionPanelBean.selectedVersionConfigFile=″ /opt/WebSphere/AppServer/config/server-cfg.xml″ parameter.
     – For WebSphere Application Server Standard Edition, V3.x, use the -W previousVersionPanelBean.selectedVersionConfigFile=″ /opt/WebSphere/AppServer/config/server-cfg.xml″ parameter.

  e. Specify the version number of the previous version, such as ″4.0″, ″4.0.1″, or ″3.5″, in the -W previousVersionPanelBean.previousVersionSelected=″4.0″ parameter.

  f. Indicate that you are selecting the version you have identified with the -W previousVersionPanelBean.migrationSelected=″true″ parameter.

  g. Specify the backup directory where WASPreUpgrade is to store information from the previous version with the -W migrationInformationPanelBean.migrationBackupDir=″ /tmp/migrationbackup″ parameter.

  h. Specify the directory for storing migration logs with the -W migrationInformationPanelBean.migrationLogfileDir=″ /tmp/migrationlogs″ parameter.

- Migrate manually if you prefer a more incremental approach.

The *Migrating* (welc_migrating) topic in the InfoCenter describes migrating from V4.x (very little to consider) or V3.5.x (several things to consider). It describes differences between V3.5.x and V5 brought about by the full compliance of V5 with Java 2 Platform, Enterprise Edition (J2EE) specifications.

3. Migrate Web server plug-ins.
4. Set up multiple versions of WebSphere Application Server to coexist.
   a. Run V3.5.x and V5 together.
   b. Run V4.0.x and V5 together.
   c. Install V5 more than once on the same machine.

There must be no run-time conflicts for multiple instances and versions of WebSphere Application Server to run at the same time on the same machine. Potential conflicts can occur in these areas:
- Prerequisites and corequisites
- Operating system registration
- Environmental conflicts
- File system
- Port assignments

Migration migrates all your resources and applications, but does not migrate entities in your classes directory. As the WebSphere Application Server installation wizard migrates applications to V5, it also migrates any applications that use IBM WebSphere Application Server Enterprise Edition, V4.0.x. The base migration removes (from the server configuration) those parts of your configuration that are dependent upon the following IBM WebSphere Application Server Enterprise Edition, V4.0.x services:

- **Business rule beans**
- **Internationalization**
- **Work area services**

Migration backs up all files in the following directories.

For V3.5.x
- `bin`
- `classes`
- `deployableEJBs` (Advanced Edition only)
- `deployedEJBs` (Advanced Edition only)
- `hosts`
- `properties`
- `servlets`

For V4.0.x
- `bin`
- `classes`
- `config` (V4.0.x Advanced Single Server Edition only)
- `installableApps`
- `installedApps`
- `installedConnectors` (V4.x Advanced Edition only)
- `properties`

You can coexist with, or migrate the applications and configuration from a previous version of WebSphere Application Server.

Migration from V4.x to V5 does not require extensive tuning. Migration from V3.5.x to V5 does require you to examine the migrating applications.

See Migrating administrative configurations manually for a description of fine tuning a migration. Part of the procedure for manual migration includes a description of what to look for after using the migration tools.

Configuration mapping during migration

This topic describes the basic migration scenario, which always involves a single machine. One typical example of this is a development environment on a stand-alone machine. Open the InfoCenter for the Network Deployment product to learn how the migration tools map models, clones, server groups, clusters, and Lightweight Third Party Authentication (LTPA) security settings.

The V5 migration tools map objects and attributes to the V5 environment when you restore a configuration from a previous version.

Stdin, stdout, stderr, passivation and working directories

The location for these directories is typically within the installation directory of a previous version. The default location for `stdin`, `stdout`, and `stderr` is the `logs` directory of the V5 installation root. The migration tools attempt to migrate existing passivation and working directories. Otherwise, appropriate V5 defaults are used.

Note: In a coexistence scenario, using common directories between versions can create problems.

Property files from V3.5.x and 4.0.x

Migration does not process property files from V3.5.x and 4.0.x. You must manually convert settings in these files to the V5 equivalent configuration.

Command line parameters

The migration tools convert appropriate command line parameters to JVM settings in the server process definition. Most settings are mapped directly. The mapping is not direct from a V3.5 and 4.0 server-owned process to a V5 process that owns multiple servers. For example, it is pointless to migrate memory heap size settings because sizes that work well in a server-owned process do not work well in a process-owned server, and vice-versa.

Bootstrap port

Migration maps a default bootstrap NameServer port setting, 900, from V3.5.x and 4.0.x Advanced Edition to the V5 NameServer default, 2809. The migration tools map a non-default value directly into the V5 environment.

For Advanced Single Server Edition migration, the bootstrap NameServer port is mapped to the NameServer of the Application Server defined in the server configuration file.

Default Server

The name of the default server in V5 is **server1**. All objects previously owned by the **Default Server** of the prior version, are owned by the **server1** of V5 after migration.

JDBC drivers and datasources

Version 5 significantly redefines JDBC and datasource objects. The migration tools map V3.5.x and V4.0.x datasources to V5 datasources, using predecessor settings as input variables. The datasource that is used is the WebSphere Application Server 4.x data source that uses the old ConnectionManager architecture.

Name bindings

There is a new naming structure in Version 5. All references, such as enterprise Java bean (EJB) references that were valid in previous versions no longer work in Version 5. However, you can use the administrative console to add a name binding that maps an old name into the new Version 5 naming structure. For example, the name of the Version 3.5.x or 4.0.x enterprise bean reference can be both the name of the binding and the JNDI name in the Version 5 name space.

For an example, see Migrating to WebSphere V5: An End-to-End Migration Guide, which is available from the Redbooks Web site at http://www.ibm.com/redbooks.

Node name

A V3.5.x and a V4.x repository can contain more than one node name and associated children. The WASPostUpgrade tool processes only those objects and children that match the node name of the node being migrated. The tool identifies node names in configuration files it is migrating, selecting any in the configuration file that match the long network name or short network name of the machine being migrated.

PageList servlet

The configuration of the PageList servlet has changed in V5. Direct use of the servlet has been deprecated. The PageList servlet is available as part of the servlet extension configuration in the WAR file. All references are updated to the servlet configuration supported in V5.

You can also use the Application Assembly Tool (AAT) to modify the servlet configuration.

If you have used or extended the PageList servlet, you might see an error similar to this when running a migrated application that uses the servlet:

```
Error 500: No PageList information is configured for servlet
EmpInfoApp.SearchByDept
```

Use the Application Assembly Tool (AAT) to correct the error, by moving the usage or extension to your migrated EAR file:

1.
   - Start the AAT (assembly) and load the EAR file that generates the error.
   - Open up the Web modules within the EAR file.
   - Expand the Web module that generates the error.
   - Open the Web components and find the one that generates the error.
   - Expand the Servlets. You should now see **PageList Extensions**.
   - Add your extension information.
   - Save the EAR file and redeploy it.

Properties directory, classes directory, and lib/app directory

Migration does not copy files from these prior version directories into the V5 configuration. Property files are not compatible between versions. The `classes` directory might contain incompatible files. You must migrate these files on an individual basis to move them into the V5 configuration.

Samples

There is no migration of samples from previous versions. There are equivalent V5 samples that you can use.

Security

Java 2 Security is enabled by default in V5. Security enablement might cause some applications to run on V4.0 and not run on V5. There are several techniques that you can use to define different levels of Java 2 Security in V5. One is to create a `was.policy` file as part of the application, to enable all security permissions. The migration tools call the **wsadmin** command to add an existing `was.policy` file in the V5 `properties` directory to enterprise applications as they are being migrated. The migration tools perform this task while moving V4.0 applications into V5.

Global security that uses LTPA authentication in Versions 3.5.x and 4.0.x is migrated to the base WebSphere Application Server product. However, although global security was enabled in Versions 3.5.x and 4.0.x, it is disabled during migration to V5.

Version 4.x introduced properties to support tuning the JNDI search timeout value along with LDAP reuse connection. These two properties are now settings in the Security Center of the V5 administrative console. There is no migration of V4.x property values to the V5 settings.

- The jndi.LDAP.SearchControl.TimeLimit property is equivalent to the V5 Search Timeout setting, which is 300 by default in V5.
- The jndi.LDAP.URLContextImplementation property is equivalent to the V5 Reuse Connection setting, which is true by default in V5.

Use the V5 administrative console to change these settings to match your V4 property values, if necessary.

Servlet package name changes

The package that contains the DefaultErrorReporter, SimpleFileServlet, and InvokerServlet servlets has changed for V5. In Versions 3.5.x and 4.0.x, the servlets are in the com.ibm.servlet.engine.webapp class. In V5, the servlets are in the com.ibm.ws.webcontainer.servlet class.

Transport ports

The migration tools migrate all ports. The tools log port conflict warnings if a port is already defined in the configuration. You must resolve port conflicts before running the servers that are in conflict, at the same time.

The default transport type of the Servlet Engine in V3.5.x is Open Servlet Engine (OSE). Because V5 no longer supports OSE transport, the migration tools map these transports to HTTP transports, using the same port assignments.

You must manually add VirtualHost alias entries for each port.

Web modules

The V5 level of J2EE requires Web container behavior changes in regard to setting content type. If a default servlet writer does not set the content type, not only does the V5 Web container no longer default to it, the Web container returns the call as ″null″. This might cause some browsers to display resulting Web container tags incorrectly. Migration sets the autoResponseEncoding IBM extension to true for Web modules as it migrates enterprise applications. This prevents the problem.

V3.5 > V5 migration

The migration tools assist in the transition from V3.5.x to V5, by migrating system configurations and creating J2EE artifacts, including J2EE security roles mapping. The migration tools create initial J2EE enterprise applications based on V3.5.x configurations. However, because of the significant change in application structures, plan to carefully test and fine tune migrated applications, using development and deployment tools, to determine exactly how the applications function in V5.

Analyze the WASPostUpgrade.log file for detailed information about migrated enterprise beans. The J2EE programming model specifies an architecture for how applications are created and deployed. Because applications in V3.5.x do not have the same architecture, the WASPostUpgrade tool recreates applications. It creates all migrated Web resources and enterprise beans in J2EE applications. It maps all enterprise applications from the V3.5.x installation into J2EE applications with the same name, deployed in the same server.

The WASPostUpgrade tool maps Web resources and enterprise beans that are not included in an enterprise application, into a default J2EE application that includes the name of the server. The tool maps Web applications to J2EE WAR files. The tool deploys enterprise beans as EJB 1.1 beans in J2EE JAR files. The tool combines resources in a J2EE EAR file and deploys it in the V5

configuration. There are some differences between the EJB 1.0 and EJB 1.1 Specifications, but in most cases, EJB 1.0 beans can run successfully as EJB 1.1 beans.

V3.5 > V5 mapping details

-
  - **datasources.xml**

    You can use a V3.5.x `datasources.xml` file to augment datasource configuration settings. V3.5.x stores the file in the `properties` directory. The migration tools migrate an existing `datasources.xml` file by merging properties in the file into the datasource and JDBC driver configuration.

  - **Enterprise applications**

    The V3.5.x enterprise-application entries are optional, they are most often used to organize sets of objects together for Security definitions. The enterprise bean and web-application portions of the enterprise-application point to their respective entries in other portions of the xml file. Each enterprise-application is processed to create a J2EE application with the same name. The enterprise bean and Web-application entries are used as pointers to the definitions of enterprise beans and Web-applications. The details of these entries are then used to build a J2EE application.

    For enterprise bean files, the JAR-file definition is used to find the JAR files to redeploy and add to the J2EE application The Web-application document-root entries are used to find the resources used within the Web-application (HTML, JSP pages, and so forth) These files are copied to the WAR file within the J2EE application. The Web-application classpath entries are used to find servlets and JAR files that are copied to the WAR file within the J2EE application.

    Enterprise applications are created during the migration from V3.5.x. These are created as J2EE 1.2 compatible enterprise applications and contain EJB 1.1-, Servlet 2.2- and JSP 1.1-level modules. This provides the most straight forward compatibility and enables interoperability with previous WebSphere Application Server versions.

  - **Enterprise beans**

    Version 3.x supports only the EJB 1.0 Components Specification level. V5 supports EJB 1.1 and 2.0 components. However, many EJB 1.0 beans can successfully deploy as EJB 1.1 beans. The migration tools redeploy enterprise beans automatically as part of the application migration phase. Check the `WASPostUpgrade.log` file for deployment details of these enterprise beans. Make any necessary changes and redeploy.

    No redeployment is required when moving EJB 1.1 JAR files from V4.

    Specify only one backend datastore vendor per JAR file. If there are enterprise beans that use different backends, package them into separate JAR files.

  - **J2EE security**

    The security authorization model in V3.5.x is based on the notion of Enterprise Application and Method Groups. The cross product of the enterprise application and the method groups is a WebSphere Application Server permission. J2EE specifies an authorization model based on roles.

    To convert from the WebSphere Application Server permission model in V3.5.x to the role based authorization model in V5, the migration tools create a one-to-one mapping from a WebSphere Application Server permission to a new role under that application. Therefore, for each enterprise application and each method group in V3.5.x, the migration tools create a role in V5, contained in the J2EE application deployment descriptor. The authorized subjects for each role are contained in an authorization table found in the J2EE application binding.

    The J2EE specification includes an authorization model based on roles. WebSphere Application Server interprets the role to mean a set of permissions to access a resource. In the case of an enterprise bean method invocation, the permission to access the method on a particular bean is specified by a method permission. This method permission is associated with one or more roles in the deployment descriptor in the bean jar file.

In the case of accessing Web resources, the permission to access a Web URI and invoke a HTTP method on that URI is specified in terms of Web resource collections and security constraints in J2EE. The deployment descriptor of the Web application WAR file contain the security constraints and Web resource collections.

- **JSP levels**

  Version 5 runs JSP 1.0 and 1.1 objects as JSP 1.2 objects, which is the only supported level.

- **Servlet Redirector**

  Version 5 does not support the Servlet Redirector from previous versions. The migration tools ignore these objects.

- **Servlet package name changes when migrating from V3.5.x to V5**

  If the V3.5.x configuration defines the SimpleFileServlet servlet, the servlet is not migrated. The migration tools set the FileServingEnabled attribute in the ibm-web-ext.xmi Web module file to `true`.

  If the V3.5 configuration defines the InvokerServlet servlet, the servlet is not migrated. The migration tools set the ServeServletsByClassnameEnabled attribute in the ibm-web-ext.xml Web module file to `true`.

  If the V3.5.x configuration defines the DefaultErrorReporter servlet, the servlet is migrated into the web.xml Web module file. Migration uses the new package to set the class name.

- **Transports**

  The default transport type of the Servlet Engine in V3.5.x is Open Servlet Engine (OSE). Because V5 no longer supports OSE transport, the migration tools map these transports to HTTP transports, using the same port assignments. You must manually add VirtualHost alias entries for each port.

V4.0 > V5 migration

This migration is much less complicated than moving from V3.5.x. The V4.0.x configuration is already at the J2EE 1.2 level. Although V5 is at the J2EE 1.3 level, J2EE 1.2 objects are supported.

-
  - **Enterprise beans**

    No redeployment is required when moving EJB 1.1 JAR files from V4.0.

    Specify only one backend datastore vendor per JAR file. If there are enterprise beans that use different backends, package them into separate JAR files.

  - **JMS Resources**

    All JMS resources from V4.0 are mapped into generic JMS resources in the V5 configuration. Reconfigure JMS resources that use IBM WebSphere MQ as IBM WebSphere MQ specific resources. MQ JMS resources have better integration with System Management. There is no need to manually define entries in the name space. You can see the backing MQ queue definitions through MQ JMS entries.

  - **JSP precompiling**

    In V4.0.x, the classes generated from JSP pages are in a package based on the directory structure of the WAR file. Any JSP at the top of the context root is in the `unnamed` package. JSP pages in subdirectories of the root are in packages named after the subdirectories. In V5, the classes generated from JSP pages are all in the package `org.apache.jsp`. Therefore, the class files are not compatible between versions.

    When migrating an enterprise application from V4.0.x to V5, recompile the JSP pages to regenerate the class files into the correct packages.

    The migration tools provide this support, by using the `-preCompileJSPs` option of the **wsadmin** tool during the installation of the application.

Use the same option to install any V4.0.x enterprise applications that you manually move to V5.

– **J2EE security**

You can apply security in two V4.x locations to enterprise applications. Information in the repository has precedence over information in the enterprise application bindings. The migration tools migrate information in the repository to the enterprise application.

– **Secure Sockets Layer (SSL) migration**

The following SSLConfig attributes that point to user-defined key files are migrated from WebSphere 4.0.x to 5.0 as follows:

4.0 Settings:

```
<key-file-name>${WAS_HOME}/keys/WASLDAPKeyring.jks</key-file-name>
<trust-file-name>${WAS_HOME}/keys/WASLDAPKeyring.jks</trust-file-name>
```

V5 settings:

```
keyFileName="${INSTALL_ROOT}/keys/WASLDAPKeyring.jks"
trustFileName="${INSTALL_ROOT}/keys/WASLDAPKeyring.jks"
```

where ${INSTALL_ROOT} is the installation root of the base WebSphere Application Server, V5 product, and also the Network Deployment product installation root.

The migration tools used by the installation wizard do not copy the key files to the corresponding directory in the base WebSphere Application Server V5 product or the Network Deployment product. After migration has completed and before enabling security, manually copy the key files to the corresponding directory under each V5 base product node and Network Deployment node in the configuration.

– **Servlet package name changes when migrating from V4.0 to V5**

If the V4.0 `web.xml` Web module file defines the SimpleFileServlet servlet, the migration tools update the class name to reflect the V5 package. The tools also set the FileServing Enabled attribute to `true`.

If the V4.0 web.xml Web module file defines the InvokerServlet servlet, the migration tools update the class name to reflect the V5 package. The tools also set the ServeServletsByClassnameEnabled attribute to `true`.

If the V4.0 web.xml Web module file defines the DefaultErrorReporter servlet, the migration tools update the class name to reflect the V5 package.

Migrating administrative configurations manually

If you use an earlier version of WebSphere Application Server, the system administrator might have fine-tuned various application and server settings for your environment. It is important to have a strategy for migrating these settings with maximum efficiency and minimal loss.

You can migrate administrative configurations with the installation wizard or manually, as this task describes. If you decide to migrate manually, do not select the migration check box on the installation wizard migration panel.

You can perform incremental manual migration by calling the migration tools multiple times, each time specifying a different configuration file. There are various reasons for having multiple configuration files. Whatever the reason, migrating one configuration file at a time lets you test applications incrementally before continuing to the next configuration file.

Before using the migration tools, consult the V5 Release Notes document to understand what fixes you must apply to earlier versions. Applying fixes to an earlier version might also apply fixes to files that have a role in the migration. Apply any fixes to verify the most effective migration of configurations and applications possible.

Manual migration provides a more incremental migration approach than the complete migration that the installation wizard provides. IBM provides a set of migration tools for migrating administrative configurations to the base WebSphere Application Server product from either edition of V3.5.x, or from V4.x. The overall migration process is to back up the current configuration and necessary files, install the V5 product, and restore the configuration.

5.0.1 Alternate procedure for migrating from the WebSphere Application Server - Express product

5.0.1 If you are migrating from the WebSphere Application Server - Express product to the base WebSphere Application Server product, do not complete the steps below. Follow the procedure for migrating from the WebSphere Application Server - Express product instead.

Steps for this task

1. Save the current configuration using the WASPreUpgrade tool.

   The WASPreUpgrade tool provides status to the screen and to log files in the backup directory. ASCII log file names start with the text WASPreUpgrade and include a date timestamp.

   The WASPreUpgrade tool saves all files from the following directories in the existing V3.5.x or V4.x configuration to the backup directory:

   For V3.5.x
   - `bin`
   - `classes`
   - `deployableEJBs` (Advanced Edition only)
   - `deployedEJBs` (Advanced Edition only)
   - `hosts`
   - `properties`
   - `servlets`

   For V4.x
   - `bin`
   - `classes`
   - `config` (V4.0.x Advanced Single Server Edition only)
   - `installableApps`
   - `installedApps`
   - `installedConnectors` (V4.x Advanced Edition only)
   - `properties`

   The WASPreUpgrade tool saves selected files from the V3.5.x and V4.x `bin` directory. It also exports the existing Application Server configuration from the repository. If you migrate V3.5.x Advanced Edition or V4.x Advanced Edition, verify the administrative server of the existing environment is running.

   If you migrate from V4.x Advanced Edition, the **WASPreUpgrade** command calls the V4.x **XMLConfig** command to export the existing Application Server configuration from the repository. If errors occur during this part of the **WASPreUpgrade** command, you might have to apply fixes to the V4.x installation to successfully complete the export step. See the IBM Support page for the latest fixes that might be applicable. When viewing this information from the InfoCenter, you can click **Support** to link to the IBM Support page.

2. Install the V5 product.

Do not select the migration option, if it appears.

If you select the coexistence option, the installation wizard updates the server1 configuration with coexistence port settings that you specify. WASPostUpgrade processing might change the coexisting port assignments so that there is a conflict. A conflict can cause server1 to fail to start if the previous Administrative Server or any of its associated Application Servers are running. After each use of WASPostUpgrade, verify V5 port settings in two files:

- Verify the `BOOTSTRAP_ADDRESS` port assignment for `server1` in the `serverindex.xml` file

  If the BOOTSTRAP_ADDRESS port of the earlier version is 900, migration maps this to 2809. If the BOOTSTRAP_ADDRESS port of the earlier version is not 900, migration maps the value to server1 in a Advanced Edition migration, or to the actual server name in an Advanced Single Server Edition migration.

- Verify the `HTTP Transport` port assignments in the `server.xml` file

  WASPostUpgrade processing adds the HTTP Transport ports from the earlier version to the V5 `server.xml` file. This means that server1 contains duplicate HTTP Transport port assignments, from both the coexistence panel and the previous version *Default Server*.

3. Migrate the previous configuration to the new installation with the WASPostUpgrade tool.

   The WASPostUpgrade tool migrates V3.5.x or V4.x configuration information created by the WASPreUpgrade tool, to the V5 installation. Because the V5 product adheres to the J2EE programming model and V3.5.x does not, significant changes are required to apply the V3.5.x configuration to a V5 installation.

   The WASPostUpgrade tool does not migrate Samples because there are new ones for J2EE in V5. Use the new Samples instead of the ones provided with the V3.5.x product.

   The WASPostUpgrade tool records detailed information specific to each enterprise bean it deploys, in the `WASPostUpgrade.log` file.

   Stop the administrative server of the earlier version before running the V5 node.

4. Configure WebSphere Application Server after migration.

   Configuring WebSphere Application Server after migration is a way of verifying the results of the migration tools. You can also use Configuration mapping during migration to verify the results of the migration. The topic has a detailed description of how the migration tools migrate objects, and what you should verify.

If you have read this topic to get an understanding of manual migration, you now have enough information to decide whether to perform manual migration, or to let the installation wizard automatically migrate your previous version of WebSphere Application Server.

Migrating from WebSphere Application Server - Express

5.0.1

This topic describes how to use the WebSphere Application Server migration tools to migrate your applications and configuration from a stand-alone WebSphere Application Server - Express product to a stand-alone base WebSphere Application Server product. Perform this migration before federating the base WebSphere Application Server node into a Network Deployment cell. If the base product is federated, you can remove it from the deployment manager cell with the **removeNode** command, located in the `bin` directory of the base product installation root.

Steps for this task

1. Apply PTFs at the same level, to both the WebSphere Application Server - Express product and the base WebSphere Application Server product.

   Use the **versionInfo** command in the `bin` directory of each installation root to verify that both products have the same level of PTF applied. The **versionInfo** command displays the exact version of each product, which must be identical.

   Do not use the **versionInfo** command while installing the product or an interim fix or fix pack.

2. Back up the current configuration of the base WebSphere Application Server product.

   Use the **backupConfig** command in the `bin` directory of the base WebSphere Application Server installation root to create a copy of the current configuration. This copy lets you restore the original configuration.

3. **5.0.2** Use the migration tools to transfer configuration settings from Express to the base product.

   This step transfers configuration information for the Express remote server resources, security, variables, and virtual hosts to the base product. Security and virtual host information is stored only at cell level. Resource and variable information is stored at server, node, and cell level. All stored information is in XML files in the `install_root/config/cells` directory of each product. For a full description of each XML file, see the ″Configuration documents″ and ″Configuration document descriptions″ topics in the InfoCenter. The **WASPreUpgrade** command line tool saves selected files from the /bin directory to a backup directory you specify as a parameter on the command. It also saves files from the following directories to the backup directory:

   - classes (This directory is not saved for iSeries platforms.)
   - config
   - installableApps
   - installedApps (or an alternate directory specified by the user)
   - properties (A subset of files in this directory is saved for iSeries platforms.)

   Later, use the WASPostUpgrade migration tool to restore the environment in the backup directory into the base WebSphere Application Server installation.

   The WASPreUpgrade migration tool migrates most of the configuration files you need in the /config directory of the Express installation root:

   - admin-authz.xml
   - namestore.xml
   - namestore-cell.xml
   - namestore-node.xml
   - naming-authz.xml
   - resources.xml
   - security.xml
   - server.xml
   - serverindex.xml
   - variables.xml
   - virtualhosts.xml

   If the migration tools warn you to manually migrate the following configuration files, follow the instructions you received from IBM for how to migrate the files:

   - namestore.xml
   - namestore-cell.xml
   - namestore-node.xml

   Perform this procedure:

   a. Run the WASPreUpgrade migration tool to export application and configuration files from Express to a backup directory.

      Migrating from Express to V5 involves fewer options than does migrating from another version. Therefore there are fewer options to specify on the WASPreUpgrade command:

```
WASPreUpgrade backupDirectoryName
              ExpressInstall_rootName
              [-traceString trace_spec [-traceFile file_name]]
```

For a full parameter list, see WASPreUpgrade command.

b. Run the WasPostUpgrade migration tool to import file from the backup directory into the base WebSphere Application Server product:

```
WASPostUpgrade backupDirectoryName
               [-cellName cell_name]
               [-nodeName node_name]
               [[-traceString trace_spec [-traceFile file_name]]
```

For a full parameter list, see WASPostUpgrade command.

4. **5.0.2** Resolve port differences.

The migration tools create a list of ports used by the combined Express/base WebSphere Application Server configuration. If the same port is used for more than one purpose, the migration tools log a message that states you must manually resolve the port conflict.

There are two groups of port differences that can remain within port assignments in the virtual hosts file, at the cell level:

- **HTTP transport ports**

  WebSphere Application Server and Express use different HTTP transport ports by default. To maintain port assignments used by Express when migrating to the base WebSphere Application Server product, add the Express port values at the server level:

  a. In the WebSphere Application Server administrative console, click the **plus sign** (″+″) next to **Servers**

  b. Click **Application Servers**.

  c. Click the server on which you plan to install your applications.

  d. Click **Web Container**.

  e. Click **HTTP transports**.

  f. Use Studio Site Developer to find the original Express remote server HTTP transport ports:

     1) Display the **Server Configuration** view.

     2) Double-click the Express remote server that you are migrating.

     3) Click **Ports**.

     4) View ports defined under **Server Settings**.

  g. Click **New** in the WebSphere Application Server administrative console to create additional entries needed to define a port.

- **Advanced ports (special end points)**

  There are a number of special purpose ports defined for WebSphere Application Server. These ports are described in Migrating and coexisting. Many advanced ports default to different values on Express than on the WebSphere Application Server. Use the WebSphere Application Server values, to continue to run both Express and the WebSphere Application Server on the same system (coexistence), or if WebSphere Application Server applications are dependent on the port values.

  Retaining WebSphere Application Server port values might require you to change applications you developed for Express. Change any references to port values you are changing. To identify Express applications that require port changes, modify the Express port configuration to use WebSphere Application Server port values. Then test the applications that you intend to migrate, to identify port references that are in error.

  a. In the WebSphere Application Server administrative console, click the **plus sign** (″+″) next to **Servers**.

  b. Click **Application Servers** to get a list of servers.

  c. Click the server on which you intend to install the applications.

d. Click **End Points**.
   e. Use Studio Site Developer to find the original Express remote server port assignments:
      1) Display the **Server Configuration** view.
      2) Double click the Express remote server that you are migrating.
      3) Click **Ports**.
      4) View **Advanced Ports** defined under **Server Settings**.
5. Verify migrated applications.

   To verify that applications are installed successfully:
   a. Click **Enterprise Applications** in the administrative console.
   b. Check your migrated applications.
   c. Click **Start**.

   Verify that your applications start successfully.

You can successfully migrate applications to the base WebSphere Application Server product.

Migrating from an unsupported operating system
You can migrate an earlier version of WebSphere Application Server V3.5.x or V4.0.x release that is running on an operating system that V5 does not support.

Steps for this task

1. Start up the WebSphere Application Server V3.5.x or V4.0.x Administrative Server.
2. Run the **WASPreUpgrade** command line migration tool.

   There are two options. You can run the command from the `migration\bin` (or `migration/bin`) directory in the `platform_root` of the V5 CD-ROM. Or, you can copy the files in the directory on the CD-ROM to a directory you create on your hard drive.

   Identify the V3.5.x or 4.0.x release, and identify a backup directory where the command stores configuration files and migrating applications from the earlier version. See WASPreUpgrade command for command syntax.

   a. Run the command from the `migration\bin` (or `migration/bin`) directory in the `platform_root` of the V5 CD-ROM.

      Identify the backup directory and the location of the configuration files.

      `CD_drive:\WASPreUpgrade backupDirectory`
      `      filepath\WebSphere\AppServer yourNodeName`

      If this works, go to Step 4. If this does not work for some reason, perform steps 2B through 2F.
   b. Make a **migration** directory on your hard drive.
   c. Copy the `WASPreUpgrade.bat` (or `WASPreUpgrade.sh`) and the `setupCmdLine.bat` (or `setupCmdLine.sh`) files from the `migration\bin\` (or `migration/bin/`) directory in the `platform_root` of the V5 CD-ROM, to the directory you created on your hard drive.
   d. Edit the `setupCmdLine.bat` (or `setupCmdLine.sh`) file in your new directory.

      Change the following variables:
      - **WAS_HOME** to point to the fully qualified path to the migration directory you created
      - **JAVA_HOME** to point to the fully qualified path to your JDK\Java directory
   e. **(Optional)** Verify that the executable bit is on for the `setupCmdLine.sh` and `WASPreUpgrade.sh` files in the `migration/bin` directory in the `UNIX-based_platform_root` of the V5 CD-ROM, if you are backing up a UNIX-based installation.
   f. Run the command from the `migration` directory you created.

      Identify the backup directory and the location of the configuration files.

   `yourMigrationDirectory\WASPreUpgrade backupDirectory filepath\WebSphere\AppServer yourNodeName`

3. Shut down the WebSphere Application Server V3.5.x or V4.0.x release by stopping all server nodes in the configuration, including the administrative server node.

4. **(Optional)** Tar or zip the backup directory and FTP it to another system.

5. Install the new operating system, keeping the same host name.

   If possible, keep the system name and passwords the same as the old system. Place any database files related to applications you are migrating in the same path as the previous system. In general, try to keep paths the same. However, do not install V5 in the same directory as the previous version. If you do change paths and names, you can edit the XML configuration files to reflect the changes. Make such changes before running the **WASPostUpgrade** command below.

6. **(Optional)** FTP the backup directory from the other system and unzip it.

7. Install WebSphere Application Server, V5. Do not select the migration option, if it appears.

8. Run the **WASPostUpgrade** command line migration tool, from the `bin` directory of the V5 installation root.

   Specify the backup directory (and any non-standard configuration file name in the directory) that the **WASPreUpgrade** command created. See WASPostUpgrade command for proper command syntax.

```
install_root\bin\WASPostUpgrade  WAS_HOME\migration
```

WASPreUpgrade command

The **WASPreUpgrade** command is a migration tool for migrating the configuration and applications of previous versions to a V5 Application Server node.

The command file is located in the `bin` subdirectory of the `install_root` directory.

The command uses the com.ibm.websphere.migration.preupgrade.WASPreUpgrade class.

Syntax

```
WASPreUpgrade backupDirectory currentWASDirectory
              [adminNodeName ]
              [-nameServiceHost host_name [-nameServicePort port_number ]]
              [-import xmiDataFile ]
              [-traceString trace_spec [-traceFile file_name ]]
```

Parameters

The first two arguments are required and positional. The third argument is required and positional only when upgrading from WebSphere Application Server, V3.5.x or Advanced Edition, V4.0.x.

Supported arguments include:

backupDirectory

Required name of the directory in which the WASPreUpgrade tool stores the saved configuration and files, and from which the WASPostUpgrade tool later reads the configuration and files. The WASPreUpgrade tool creates this directory if it does not already exist.

This parameter is equivalent to the -W migrationInformationPanelBean. migrationBackupDir=″ /tmp/migrationbackup″ parameter in the silent installation options response file.

currentWASDirectory

Required name of the installation root for the current V3.5.x or V4.x installation. This version can be either WebSphere Application Server Standard Edition, V3.5.x, WebSphere Application Server Advanced Edition, V3.5.x, any form of WebSphere Application Server, V4.0.x, or WebSphere Application Server - Express, V5.

5.0.2 This also includes WebSphere Application Server - Express, V5.

This parameter is equivalent to the -W previousVersionPanelBean. selectedVersionInstallLocation=″ /opt/WebSphere/AppServer″ parameter in the silent installation options response file.

adminNodeName

The name of the node containing the administrative server for the currently installed product. The value of this argument must match the node name given in the topology tree on the `Topology` tab of the administrative console for the currently installed product. The WASPreUpgrade tool calls the XMLConfig tool using this parameter. This parameter is only required when upgrading from WebSphere Application Server Standard Edition, V3.5.x, WebSphere Application Server Advanced Edition, V3.5.x, or WebSphere Application Server Advanced Edition, V4.0.x.

There is no equivalent parameter in the silent installation options response file.

-nameServiceHost -nameServicePort

When specified, the WASPreUpgrade tool passes these optional parameters to the XMLConfig tool. Use these parameters to override the default host name and port number used by the XMLConfig tool.

There is no equivalent parameter in the silent installation options response file.

-import

The name of the WebSphere Application Server Advanced Single Server Edition, V4.0 XML Metadata Interchange (XMI) configuration file to process. This parameter is optional because the program uses the `config\server-cfg.xml` file by default.

When migrating a configuration that uses anything other than the default `server-cfg.xml` file name, you must use the -import option along with a path to point to the non-default XMI configuration file. You also must use the -import and path option when running the WASPostUpgrade tool later, to point to the non-default XMI configuration file in the directory created by the WASPreUpgrade tool.

This parameter is equivalent to the -W previousVersionPanelBean. selectedVersionConfigFile=″ /opt/WebSphere/AppServer/config/server-cfg.xml″ parameter in the silent installation options response file.

-traceString -traceFile

Optional parameters to gather trace information for IBM Service personnel. Specify a trace_spec of ″*=all=enabled″ (with quotation marks) to gather all trace information.

There is no equivalent parameter in the silent installation options response file.

Logging

The WASPreUpgrade tool displays status to the screen while it is running. It also saves a more extensive set of logging information in the `backup` directory. You can view the `WASPreUpgrade.log` file with a text editor.

Examples

The following examples demonstrate correct syntax.

Migrating from a 3.5.x Standard Edition or Advanced Edition or from a 4.0.x Advanced Edition

The following example specifies a backup directory named backupDirectory, and identifies the installation root of the existing installation as `d:\WebSphere\AppServer`.

```
WASPreUpgrade backupDirectory d:\WebSphere\AppServer yourNodeName
```

Migrating from a V4.0.x Advanced Single Server Edition node with multiple backup directories

This example shows how to migrate incrementally, to migrate separate configuration files from a single node with a single installation of WebSphere Application Server Advanced Single Server Edition. To migrate more than one configuration file, you must run the WASPreUpgrade tool multiple times to multiple backup directories because not all of the applications are in the same `installedApps` directory. For this reason, using a single backup directory for all runs of the WASPreUpgrade tool is not recommended. Use a separate backup directory for each run. The intent of this example is to show how to migrate a single node with multiple configuration files.

1. Run the following WASPreUpgrade commands to migrate applications A, B, C, D, and E, which reside in three separate application directories. Server assumptions include:

   - The Application Server uses the default configuration file, `server-cfg.xml`, as well as `myServer1-cfg.xml` and `OldServer-cfg.xml`.

```
> WASPreUpgrade "C:\WAS4ABBACKUP"  G:\WebSphere\AppServer
> WASPreUpgrade "C:\WAS4CDBACKUP"  G:\WebSphere\AppServer
      -import  G:\WebSphere\AppServer\config\myServer1-cfg.xml
> WASPreUpgrade "C:\WAS4EBACKUP"  G:\WebSphere\AppServer
      -import G:\WebSphere\AppServer\config\OldServer-cfg.xml
```

2. Run the following WASPostUpgrade commands to restore the applications and configurations to the V5 Application Server:

```
> WASPostUpgrade  C:\WAS4ABBACKUP
> WASPostUpgrade "C:\WAS4CDBACKUP" -import "C:\WAS4CDBACKUP\myServer1-cfg.xml"
> WASPostUpgrade "C:\WAS4EBACKUP"  -import "C:\WAS4EBACKUP\OldServer-cfg.xml"
```

WASPostUpgrade command

The **WASPostUpgrade** command is a migration tool for migrating the configuration and applications of previous versions to a V5 Application Server node.

The command file is located in the `bin` subdirectory of the `install_root` directory.

The command uses the com.ibm.websphere.migration.postupgrade.WASPostUpgrade class.

The WASPostUpgrade tool installs all migrated applications into the `WAS_install_root/installedApps` directory for the V5 installation. The tool includes applications that it found in the `installedApps` directory and other directories from the earlier version.

Syntax

```
WASPostUpgrade backupDirectory [-import xmi_data_file]
                               [-cellName cell_name]
                               [-nodeName node_name]
                               [-serverName server_name]
                               [-webModuleAdditionalClasspath classpath]
                               [-documentRootLimit number]
                               [-substitute "key1=value1[;key2=value2;[...]]"]
                               [-portBlock port_starting_number]
                               [[-traceString trace_spec [-traceFile file_name]]
```

Parameters

The first argument is required. Supported arguments include:

backupDirectory

Required name of the directory in which the WASPreUpgrade tool stores the saved configuration and files, and from which the WASPostUpgrade tool reads the configuration and files. The WASPreUpgrade tool creates this directory if it does not already exist.

This parameter is equivalent to the -W migrationInformationPanelBean. migrationBackupDir=″/tmp/migrationbackup″ parameter in the silent installation options response file.

-import

The name of the WebSphere Application Server Advanced Single Server Edition, V4.0 XML

Metadata Interchange (XMI) configuration file to process. This parameter is optional because the program uses the `config\server-cfg.xml` file by default.

When migrating a configuration that uses anything other than the default `server-cfg.xml` file name, you must use the -import option along with the path to the non-default XMI configuration file in the directory created by the WASPreUpgrade tool.

5.0.2 This parameter is ignored when migrating from WebSphere Application Server - Express, V5.

This parameter is equivalent to the -W previousVersionPanelBean. selectedVersionConfigFile=″ /opt/WebSphere/AppServer/config/server-cfg.xml″ parameter in the silent installation options response file.

-cellName

Optional parameter to specify the cell name for the program to update. If not specified, the program inspects the configuration for cell names. When one cell name exists, the program uses it. Otherwise, the program returns an error.

This parameter is equivalent to the -W nodeNameBean. cellName=″nodenameNetwork″ parameter in the silent installation options response file for the Network Deployment product. There is no equivalent parameter in the silent installation options response file for the base product.

-nodeName

Optional parameter to specify the node name for the program to update. If not specified, the program inspects the configuration for node names. When one node name exists, the program uses it. Otherwise, the program returns an error.

This parameter is equivalent to the -W nodeNameBean. nodeName=″nodenameManager″ parameter in the silent installation options response file for the Network Deployment product. There is no equivalent parameter in the silent installation options response file for the base product.

-webModuleAdditionalClasspath

Optional parameter to specify the path or the path and file names of specific directories or files that you do not want copied into the Web archive (WAR) file. Instead, the program adds the paths and files to the Web Module extension (`ibm-web-ext.xmi`) `additionalClassPath` attribute. This is only applicable when migrating a V3.5.x installation.

There is no equivalent parameter in the silent installation options response file.

-documentRootLimit

Optional parameter to specify the number of files that the program copies from the document-root field of Web-application. It is only applicable to V3.5.x upgrades. If not specified, the default is 300.

There is no equivalent parameter in the silent installation options response file.

-substitute

Optional argument passed to the XMLConfig tool. Specify values for security variables to substitute (for example, `-substitute "NODE_NAME=admin_node;APP_SERVER=default_server"`).

In the input XML data file, each key appears as *key* for substitution. This argument substitutes any occurrence of `$NODE_NAME$` with *admin_node* and `$APP_SERVER$` with *default_server* in the input XML file.

If the substitution string contains semicolons, use `$semiColon$` to separate it from the ″;″ delimiter. On UNIX platforms, add an escape character to each dollar sign ($) within the substitution string (for example, `\$semiColon\$`).

This parameter is applicable for configurations saved from Advanced Edition, Versions 3.5.x and 4.0.x.

There is no equivalent parameter in the silent installation options response file.

-portBlock

> Optional parameter used only on iSeries platforms, to specify the starting value to use when creating ports.

> There is no equivalent parameter in the silent installation options response file.

-traceString -traceFile

> Optional parameters to gather trace information for IBM Service personnel. Specify a trace_spec of `"*=all=enabled"` (with quotation marks) to gather all trace information.

> There is no equivalent parameter in the silent installation options response file.

Logging

The WASPostUpgrade tool displays status to the screen while running. It also saves a more extensive set of logging information in the `logs` directory. You can view the `WASPostUpgrade.log` file with a text editor.

Examples

The following examples demonstrate correct syntax.

Migrating from a 3.5.x Standard Edition or Advanced Edition or from a 4.0.x Advanced Edition

The following example identifies the backup directory, named backupDirectory, that contains exported configuration files processed by the WASPreUpgrade tool.

```
WASPostUpgrade backupDirectory
```

This example shows how to migrate incrementally, to migrate separate configuration files from a single node with a single installation of WebSphere Application Server Advanced Single Server Edition. To migrate more than one configuration file, you must run the WASPreUpgrade tool multiple times to multiple backup directories because not all of the applications are in the same `installedApps` directory. For this reason, using a single backup directory for all runs of the WASPreUpgrade tool is not recommended. Use a separate backup directory for each run. The intent of this example is to show how to migrate a single node with multiple configuration files.

1. Run the following WASPreUpgrade commands to migrate applications A, B, C, D, and E, which reside in three separate application directories. Server assumptions include:

    *   The Application Server uses the default configuration file, `server-cfg.xml`, as well as `myServer1-cfg.xml` and `OldServer-cfg.xml`.

```
> WASPreUpgrade "C:\WAS4ABBACKUP"  G:\WebSphere\AppServer
> WASPreUpgrade "C:\WAS4CDBACKUP"  G:\WebSphere\AppServer
      -import  G:\WebSphere\AppServer\config\myServer1-cfg.xml
> WASPreUpgrade "C:\WAS4EBACKUP"  G:\WebSphere\AppServer
      -import G:\WebSphere\AppServer\config\OldServer-cfg.xml
```

2. Run the following WASPostUpgrade commands to restore the applications and configurations to the V5 Application Server:

```
> WASPostUpgrade  C:\WAS4ABBACKUP
> WASPostUpgrade "C:\WAS4CDBACKUP" -import "C:\WAS4CDBACKUP\myServer1-cfg.xml"
> WASPostUpgrade "C:\WAS4EBACKUP"  -import "C:\WAS4EBACKUP\OldServer-cfg.xml"
```

Configuring WebSphere Application Server after migration

The installation wizard automatically configures IBM WebSphere Application Server, V5 and all other bundled products. There is no need for additional configuration if you do not migrate from an earlier version.

If you use the installation wizard to migrate a previous installation of WebSphere Application Server, V3.5.x, there are some items to review before considering your environment fully configured.

Steps for this task

1. Check the `WASPostUpgrade.log` file for deployment details of the V3.5.x enterprise beans. Make any necessary changes and redeploy.

   Note: No redeployment is required when moving EJB 1.1 JAR files from V4.

   The J2EE programming model specifies an architecture for how applications are created and deployed. The WASPostUpgrade tool recreates applications because V3.5.x applications do not have the same architecture as V5 applications. The new V5 J2EE applications contain all migrated Web resources and enterprise beans. All V3.5.x enterprise applications become V5 J2EE applications with the same name, deployed in the same server.

   The WASPostUpgrade tool maps Web resources and enterprise beans that are not included in an enterprise application, into a default J2EE application that includes the name of the server. The tool maps Web applications to J2EE WAR files. The tool deploys enterprise beans as EJB 1.1 beans in J2EE JAR files. The tool combines resources in a J2EE EAR file and deploys it in the V5 configuration. There are some differences between the EJB 1.0 and EJB 1.1 Specifications, but in most cases, EJB 1.0 beans can run successfully as EJB 1.1 beans.

   Version 3.5.x supports only the EJB 1.0 components specification level. V5 supports EJB 1.1 components. However, many EJB 1.0 beans can successfully deploy as EJB 1.1 beans. The migration tools redeploy enterprise beans automatically as part of the application migration phase.

2. Examine any Lightweight Third Party Authentication (LTPA) security settings you might have used, and apply the settings in the WebSphere Application Server security settings.

   The WASPostUpgrade tool migrates applicable security settings in the V3.5.x environment to J2EE security attributes.

   Global security that uses LTPA authentication in Versions 3.5.x and 4.0.x is migrated to the base WebSphere Application Server product. However, although global security was enabled in Versions 3.5.x and 4.0.x, it is disabled during migration to V5.

3. Check the `WASPostUpgrade.log` file in the `logs` directory, for details about JSP 0.91 objects that the migration tools do not migrate.

   Version 5 does not support JSP 0.91 objects. The migration tools do not migrate JSP objects configured to run as JSP 0.91 objects. The migration tools do, however, recognize the objects in the output and log them. V5 runs JSP 1.0 and 1.1 objects as JSP 1.2 objects, which is its only supported level.

4. Identify and manually migrate non-migrated nodes in V3.5.x and V4.x repositories that have multiple nodes.

   A V3.5.x and a V4.x repository can contain more than one node name and its associated children. The WASPostUpgrade tool processes only those objects and children that match the migrating node. This determination is made by checking the names of nodes in configuration files with fully qualified and non-qualified network names of the migrating machine.

5. Examine any applications that use IBM extensions.

   In many cases, IBM provides additional features and customization options that extend the specification level even further. If your existing V3.x applications use IBM extensions from earlier product versions, you might need to perform mandatory or optional migration to use the same kinds of extensions in the V5.

   Migration from V4.x requires little conversion.

6. Update J2EE resources in client JAR files to the new resource format with the ClientUpgrade tool.

   J2EE applications might exist on the client, if the client has client JAR files with J2EE resources.

7. Migrate V3.5.x XML applications to supported XML APIs

   If your XML applications use XML for Java API, V2.0.x or earlier, you must migrate them to API V3.1 or the equivalent open-source version. Although there are inherent performance improvements in later versions of the XML for Java API, you can gain additional performance by explicitly using nonvalidating parsers in application environments where you can trust the data.

The most significant change is that the TX-compatible APIs are no longer available. The Document API retains the XML manipulation APIs that were in TXDocument, but you must rewrite the following functionality:

- Creating and loading an XML parser: Use a Java API for XML Processing (JAXP) factory class.
- Writing out the Document Object Model (DOM) tree: Use a serializer. One drawback to the DOM Level 2 implementation in this level of the XML for Java API is that the grammar (DTD or schema) is no longer a node in the DOM tree, so you cannot write it out. As a result, only external grammars are recommended. You can query the system ID of the root element and use it to retrieve the name from the statement. After the tree is written to an XML file, you can read the file as text and insert a statement.

In addition to the XML API changes, it is important to understand that J2EE V1.3 mandates the use of JAXP 1.1, DOM V2, and SAX V2. JAXP V1.2, DOM V3, and SAX V3 are not allowed in products that are compliant with the J2EE V1.3 specification. This prohibition exists because the newer versions were *experimental* at the time of the J2EE V1.3 specification. Because WebSphere Application Server is compliant with the J2EE V1.3 specification, WebSphere Application Server has support for JAXP V1.1, DOM V2 and SAX V2 only.

You must only recompile a V4.0.x XML application to migrate it to the V5 level.

8. Configure WebSphere Application Server to use a database.

   For example, you can configure WebSphere Application Server to use DB2.

9. Review your Java virtual machine settings (urun_rconfproc_jvm in the InfoCenter) to verify that you are using a heap size of at least 50 for improved startup performance.

   If you have used a smaller heap size in the past, you can use the default heap size, which is now 50.

Now you are finished with pre-test configuration. You might have to fine tune your WebSphere Application Server environment as you test it. Test all redeployed applications before moving them into production.

Configuring WebSphere Application Server for DB2 access

In WebSphere Application Server V4, environment variables required for accessing a DB2 database, such as CLASSPATH, native library path, db2instance name and so on, are included or initialized in the WebSphere Application Server `admin.config` file or in the **setupCmdLine.sh** and **startupServer.sh** command scripts. Because WebSphere Application Server, V5 does not store administration repository information in a relational database management system (RDBMS), a database is not a prerequisite. Therefore, the installation and configuration of V5 do not involve database configuration. This topic describes how to configure V5 so that its applications can access a DB2 database.

With the introduction of DB2 UDB V8.1, you can install both DB2 V7 and V8.1, or multiple DB2 V8.1 instances on the same UNIX machine. This topic also describes configuring V5 on a UNIX-based platform to use multiple DB2 clients, including a V7 client and a V8.1 client.

If your WebSphere Application Server, V5 server machine has only a V7 client or a V8.1 client installed, and all DB2 data sources defined in WebSphere Application Server access DB2 databases through this client, source the `db2profile` file in the login profile of your V5 instance owner, , invoke the script `db2_home/java12/usejdbc2` to use the JDBC2 drivers instead of the default JDBC1 drivers, and put the DB2 `lib` directory into the `java.library.path`.

For example, assume that the following values are true:

WebSphere Application Server instance owner, that is the administrative user who starts WebSphere Application Server	adm00001
DB2 client instance owner	db2inst1
DB2 client instance home	/export/home/db2inst1

Steps for this task
1. Update the `.profile` of user adm0001.

    Add the following line:

    `. /export/home/db2inst1/sqllib/db2profile`

    You can also add the line to the WebSphere Application Server, V5 `setupCmdLine.sh` script.
2. Specify the JDBC provider class path.

    There are two ways to specify the JDBC provider class path on the DB2 JDBC Provider definition panel of the WebSphere Application Server administration console:

    *   Leave the default value (`${DB2_JDBC_DRIVER_PATH}/db2java.zip`) as is. Go to the **Environment > Manage WebSphere Variables** in the WebSphere Application Server administrative console. Set the DB2_JDBC_DRIVER_PATH variable to the value `/export/home/db2inst1/sqllib/java12`. This is the preferred approach.
    *   Enter the concrete class path, such as `/export/home/db2inst1/sqllib/java12/db2java.zip`.

    Windows platforms support only one DB2 installation. The DB2 environment variables are populated in the system environment automatically. It is not necessary that WebSphere Application Server set these environment variables.
3. Set DB2 required environment variables for a particular Application Server.
    a. On the WebSphere Application Server Version 5 administrative console, click **Application Servers > *server* > Configuration > Additional Properties > Process Definition > Additional Properties > Environment Entries > New** to create a new entry.
    b. Create a DB2INSTANCE entry. Enter **DB2INSTANCE** in the Name field, and the instance name, such as **db2v7** in the Value field.
    c. Create a library path entry.

        Create the appropriate library path:
        *   **AIX:** LIBPATH entry
        *   **HP-UX:** SHLIB_PATH entry
        *   **Linux:** LD_LIBRARY_PATH entry
        *   **Solaris:** LD_LIBRARY_PATH

        Use the value of the DB2 native library path, such as `/export/home/db2v7/sqllib/java12:/export/home/db2v7/sqllib/lib` for a V7 client, or `/export/home/db2v8/sqllib/lib` for a V8.1 client.

    The DB2 `lib` directory must be on the `java.library.path`, otherwise the Application Server cannot load the library `db2jdbc.so` and cannot work with DB2. Even with the DB2 `lib` directory on the `java.library.path`, you must invoke the `/home/db2inst1/db2profile` environment into the root shell before you start the Application Server.
4. **(Optional)** Configure the WebSphere Application Server product in a way that it can use both DB2 V7 and V8.1 clients.

    In cases where both a DB2 V7 client and a V8.1 client, or multiple V8.1 clients are installed on a WebSphere Application Server V5 machine, and you intend to use two or more clients in your V5 Application Servers, you can set DB2 environment variables based on each Application Server, instead of setting them globally as shown in the previously.

    DB2 UDB V8 uses a new client and server communications mechanism that is based on distributed relational database architecture. While the new communications mechanism provides a number of advantages, it also introduces restrictions on the communications capability between DB2 V7 and V8 products.

    Because there are a number of restrictions when using a V8 client to communicate with a V7 server, this configuration is not recommended. However, a V7 client can access a V8 server without difficulty.

If you have a WebSphere Application Server application that accesses both a V7 server and a V8 server, only one V7 client is required on the Application Server instance. Use this server to access both the V7 server and the V8 server.

You can choose to use a V7 client to access a V7 server, and use a V8 client to access a V8 server. This results in two versions of DB2 clients on the same WebSphere Application Server machine.

When you use a V7 client to access a V8 server, you must explicitly bind V7 packages to the V8 server. To do this in the V7 client environment, make a connection to the V8 server and bind both `db2cli.lst` and `db2ubind.lst` files. For example:

```
cd /home/db2inst1/sqllib/bnd
db2 connect to v8server
db2 bind @db2cli.lst
db2 bind @db2ubind.lst
db2 terminate
```

One WebSphere Application Server node can support multiple Application Server instances. Each Application Server is essentially a single Java run-time environment-one Java Virtual Machine (JVM). Each JVM can have its own set of environment variables that differ from other Application Server instances.

You can set DB2 environment variables per Application Server instance. This means that each Application Server instance communicate with a single DB2 client instance. The client instance can have multiple databases catalogued.

Such a configuration means that you cannot have one Application Server instance communicating with both a V7 client and a V8.1 client. However, you can create another Application Server instance to communicate with a different DB2 client.

5. Match the appropriate data source to the application component.

   When mapping a data source to an application component, do not mismatch data sources from different DB2 client instances.

   For example, if you set the server1 Application Server instance to run in the DB2 V7 client instance, server1 application components should use only data sources defined under the same V7 client JDBC driver.

   WebSphere Application Server V5 supports defining a JDBC provider on different scopes: cell, node (the default) and server. If you have different DB2 client instances, consider defining them on the server level instead of on the node level. Server level definitions avoid possible mismatches between the data sources and different DB2 JDBC providers.

Now you are finished with pre-test configuration. You might have to fine tune your WebSphere Application Server environment as you test it. Test all redeployed applications before moving them into production.

Coexistence support

Coexistence as it applies to WebSphere Application Server products, is the ability of multiple installations of WebSphere Application Server to run on the same machine at the same time. Multiple installations include multiple versions and multiple instances of one version. Coexistence also implies various combinations of Web server interaction.

The V5 WebSphere Application Server products can coexist with supported, previous versions as described below. The installation wizard looks for these existing installations to determine if it should prompt you for coexistence information:

- IBM WebSphere Application Server Standard Edition and Advanced Edition, V3.5.5 and up
- IBM WebSphere Application Server Advanced Server Single Edition and Advanced Edition, V4.0.2 and later
- IBM WebSphere Application Server Enterprise Edition, V4.0.2 and later

Multiversion coexistence scenarios appear in the following table. Open the InfoCenter for the Network Deployment product to see coexistence scenarios for that product.

Multiversion coexistence scenarios

Installed product	IBM WebSphere Application Server, V5
IBM WebSphere Application Server Standard Edition, V3.5.5 and later	Supported
IBM WebSphere Application Server Advanced Edition, V3.5.5 and later	Supported
IBM WebSphere Application Server Advanced Single Server Edition, V4.0.2 and later	Supported
IBM WebSphere Application Server Advanced Edition, V4.0.2 and later	Supported
IBM WebSphere Application Server Enterprise Edition, V4.0.2 and later	Supported

In addition to multiversion coexistence, WebSphere Application Server also lets you install multiple times on one machine (multiple installation instances), or install once and have multiple configurations (multiple configuration instances).

Multiple V5 instances on one machine include:

- Multiple Application Server instances from multiple installations of the base WebSphere Application Server product
- Multiple Application Server configuration instances from a single installation of the base product

Multiple WebSphere Application Server instance scenarios

Installed product	IBM WebSphere Application Server, V5
IBM WebSphere Application Server, V5	Supported, but with limitations

Setting up V3.5.x and V5 coexistence

You must migrate prerequisite and corequisite programs to the levels required by WebSphere Application Server, V5. You must also identify ports in use in V3.5.x before you begin the V5 installation, to avoid possible conflicts during coexistence. The first two steps in this task describe these activities.

You can install WebSphere Application Server V3.5.5 (and later) and V5 on the same node. When the V5 installation wizard detects the V3.5.x installation, it displays the migration and coexistence panel, where you can select either option, or neither option. If you select coexistence, the installation wizard displays the port number panel, to verify you install V5 without port conflicts.

Silent installation also supports configuring for coexistence silently. You can specify non-conflicting port assignments in the options response file.

By default, there are port conflicts between V3.5.x and V5 that you must resolve. Also, if you migrate more than two V3.5.5 nodes to V5, there are port conflicts that you must resolve, as described in Setting up V5 coexistence.

Steps for this task

1. Migrate prerequisite and corequisite programs to the levels required by WebSphere Application Server, V5.

Refer to the WebSphere Application Server supported hardware, software, and APIs Web site at http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html for current requirements.

2. Resolve port conflicts.

   Refer to ″Port number settings in WebSphere Application Server versions″ to see a list of default port numbers, and where they are defined.

   Inspect the configuration of the V3.5.x product, either WebSphere Application Server Advanced Edition, or WebSphere Application Server Standard Edition.

   If necessary, use the following command to examine existing node and port settings when the administrative server is running.

   ```
   xmlConfig -export config.xml -nodeName theNodeName
   ```

   Review the output xml file to look for <node action=″update″ name=″nodename″> to determine the node name. You can also find port number assignments in the output.xml file. The installation wizard displays a default set of coexistence port numbers as suggested V5 port numbers. Change the values to ports that are not in use. The installation wizard uses whatever values you approve.

   Change conflicting HTTP transport ports manually, if necessary.

3. Associate a Web server with each WebSphere Application Server:
   - Use a separate Web server for each WebSphere Application Server.
     a. Create a Web server instance using the Web server documentation.
     b. Select the appropriate Web server plug-in feature during WebSphere Application Server installation, and identify the Web server configuration file location. (For example, identify the location of the `httpd.conf` file for IBM HTTP Server.)
   - Use the same Web server for both WebSphere Application Server versions.

     Upgrade the Web server to the common level supported by both versions of the Application Server.

     See one of the following topics in the online InfoCenter for more information about preparing the same Web server to support both Application Server versions:
     – Migrating IBM HTTP Server to support multiple WebSphere Application Server versions
     – Migrating the Sun ONE (iPlanet) Web server to support multiple WebSphere Application Server versions
     – Migrating Lotus Domino to support multiple WebSphere Application Server versions
     – Migrating Microsoft Internet Service Manager to support multiple WebSphere Application Server versions

4. **(Optional)** Fix any problems with environmental variables on Windows platforms.

   For example, installing WebSphere Application Server, V5 updates the system variable PATH, potentially affecting tools with the same name across installations. To run tools with conflicting names, alter the PATH environment variable in a command window and place the directory for the former installation before the directory for the latter installation. For example, PATH=E:\WebSphere\AppServer\35\bin;%PATH%. Then, invoke the tools from the `bin` directory.

Setting up V4.0.x and V5 coexistence

You must migrate prerequisite and corequisite programs to the levels required by WebSphere Application Server, V5. You must also identify ports in use in V4.0.x before you begin the V5 installation, to avoid possible conflicts during coexistence. The first two steps in this task describe these activities.

You can install WebSphere Application Server V4.0.x and V5 on the same node. When the V5 installation wizard detects the V4.0.x installation, it displays the migration and coexistence panel, where you can select either option, both options, or neither option. If you select coexistence, the installation wizard displays the coexistence panel to verify you install V5 without port conflicts.

Silent installation also supports configuring for coexistence silently. You can specify non-conflicting port assignments in the options response file.

By default, there are port conflicts between V4.0.x and V5 that you must resolve. Also, if you migrate more than two V4.0.x nodes to V5, there are port conflicts that you must resolve, as described in Setting up V5 coexistence.

Steps for this task

1. Migrate prerequisite and corequisite programs to the levels required by WebSphere Application Server, V5.

   Refer to the WebSphere Application Server supported hardware, software, and APIs Web site for current requirements.

2. Resolve port conflicts.

   Refer to Port number settings in WebSphere Application Server versions to see a list of default port numbers, and where they are defined.

   Inspect the configuration of the previous version:

   - **For WebSphere Application Server Advanced Single Server Edition, V4.0.x:** Inspect the `server-cfg.xml` file to get port values for the configuration.

   - **For WebSphere Application Server Advanced Edition, V4.0.x:** Inspect the admin.config file to get port values for the configuration. When the administrative server is running, use this command:

     ```
     xmlConfig -export config.xml -nodeName theNodeName
     ```

     Review the `config.xml` file to look for <node action="update" name="nodename"> to find the appropriate node and port number assignments in the file. The installation wizard displays a default set of coexistence port numbers as suggested V5 port numbers. Change the values to ports that are not in use. The installation wizard uses whatever values you approve.

     Change conflicting HTTP transport ports manually, if necessary.

3. Associate a Web server with each WebSphere Application Server.

   - Use a separate Web server for each WebSphere Application Server.

     a. Create a Web server instance using the Web server documentation.

     b. Select the appropriate Web Server plug-in feature during WebSphere Application Server installation, and identify the Web server configuration file location. (For example, identify the location of the `httpd.conf` file for IBM HTTP Server.)

   - Use the same Web server for both WebSphere Application Server versions.

     To use the same Web server for both Application Server versions, you must first upgrade the Web server to the common level supported by both versions of the Application Server.

     Follow this procedure to use the same Web server for both WebSphere Application Server versions.

     a. Select the appropriate Web Server plug-in feature during WebSphere Application Server installation, and identify the Web server configuration file location. (For example, identify the location of the `httpd.conf` file for the IBM HTTP Server that is associated with V4.0.x.)

     b. Edit the Web server configuration file to remove entries for V4.0.x.

        The WebSphere Application Server, V5 plug-in acts as the routing agent to route requests to both versions.

     c. Generate the plug-in configuration files for both versions of the Application Server.

     d. Replace the original `plugin-cfg.xml` file of the V5 installation with the master file.

4. **(Optional)** Fix any problems with environmental variables on Windows platforms.

   For example, installing WebSphere Application Server, V5 updates the system variable PATH, potentially affecting tools with the same name across installations. To run tools with conflicting names, alter the PATH environment variable in a command window and place the directory for the former

installation before the directory for the latter installation. For example, `PATH=E:\WebSphere\AppServer\40\bin;%PATH%`. Then, invoke the tools from the `bin` directory.

Setting up V5 coexistence

After installing the base WebSphere Application Server product, you can install it again on the same machine.

Select the new installation option from the installation wizard panel, to install a new instance instead of adding features to the last installation, and to install into a separate installation directory. Select the coexistence option to provide non-conflicting ports.

Each installation of the base product is a stand-alone Application Server (server1) with its own set of unique configuration files.

Be aware of multiple instance limitations:
- If you install more than one instance, the most recent installation is the only one in the operating system registry.
- If you install more than two instances, the third and subsequent installations require that you change all port numbers on the coexistence panel, to avoid potential conflicts.
- To uninstall a registered product instance, always use the operating system remove program function, such as the Add/Remove Program utility on Windows 2003. To uninstall an unregistered instance, use the **Uninstall.exe** or **uninstall** command (or the **uninstall.sh** command on Linux for S/390 platforms) in the `_uninstall` directory of the `install_root` that matches the instance you intend to remove.

Reasons to use multiple installation instances include:
- You can achieve complete isolation between each WebSphere Application Server instance. You can uninstall one instance independently of the others.
- You can install the base WebSphere Application Server more than once on the same machine.

Reasons to not use multiple installation instances include:
- The machine might have a hard disk space constraint.
- You cannot use the operating system registry to locate any but the last installed instance of a WebSphere Application Server product.

  When you install any product a second time, the last installation is the one that appears in the registry.
- Uninstalling the last instance removes any record of the product in the registry.

  Suppose you have installed three instances of the base WebSphere Application Server product. Further suppose that you use the operating system remove program function to uninstall the registered third copy of the base product. There is no longer a registry record to indicate the existence of the other two installation instances. Other applications cannot use a query of the operating system registry to detect the presence of either base WebSphere Application Server product instance.

Creating multiple V5 configuration instances describes installing the base WebSphere Application Server product once and creating multiple configuration instances.

Use the following procedure to install multiple installation instances.

Steps for this task
1. Use the installation wizard for multiple installations.

   If you intend to share a single Web server among installations, install a Web server plug-in feature during only one installation, as described in the next step.
2. Share a Web server among multiple installation instances.
   a. Select a Web server plug-in feature only during one installation.

b. Generate the plug-in configuration files for every installation instance.

c. Edit the `plugin-cfg.xml` configuration files to merge them into one master configuration.

d. Replace the original `plugin-cfg.xml` file with the master file, on the Application Server where you selected the Web server plug-in feature.

You can access samples from only one of the installation instances.

3. **(Optional)** Create additional servers in a multiple instance or coexistence environment.

4. Change port assignments in configuration files if you have a node that you cannot start because of port conflicts.

Port number settings in WebSphere Application Server versions

This topic provides reference information about identifying port numbers in versions of WebSphere Application Server, as a means of determining port conflicts that might occur when you intend for an earlier version to coexist or interoperate with V5.

V3.5.x port numbers

Resolve port conflicts by inspecting the configuration of the V3.5.x product, either WebSphere Application Server Advanced Edition or WebSphere Application Server Standard Edition. When the administrative server is running, use this command to examine current node and port settings:

`xmlConfig -export config.xml -nodeName` *theNodeName*

Look for the OSE Remote port assignments.

Table 10. WebSphere Application Server V3.5.x default port definitions in the admin.config file

Port name	Standard Edition	Advanced Edition
	Value	Value
lsdport (Location Service Daemon)	9000	9000
bootstrapPort	900	900
LSDSSL (With security on)	9001	9001

V4.0.x port numbers

For WebSphere Application Server Advanced Single Server Edition, V4.0.x: Inspect the `server-cfg.xml` file to find the Web container HTTP transports port values for the configuration.

For WebSphere Application Server Advanced Edition, V4.0.x: When the administrative server is running, use this command to extract the configuration from the database:

`xmlConfig -export config.xml -nodeName` *theNodeName*

Look for the Web container HTTP transports port assignments.

Table 11. WebSphere Application Server V4.0.x port definitions

Port name	Value	Advanced Edition	Enterprise Edition	Advanced Single Server Edition
		File	File	File
bootstrapPort	900	admin.config	admin.config	server-cfg.xml
lsdPort	9000			
LSDSSLPort	9001			
HTTP transport port	9080	database	database	
HTTPS transport port	9443			
Admin Console HTTP transport port	9090			
ObjectLevelTrace	2102			
diagThreadPort	7000			

V5 port numbers

You can use the administrative console to configure the V5 Application Server, to resolve conflicting ports.

Table 12. WebSphere Application Server V5 default port definitions

Port name	WebSphere Application Server Value	File
HTTP_TRANSPORT	9080	• server.xml • virtualhosts.xml
HTTPS Transport Port (HTTPS_TRANSPORT)	9443	
HTTP Admin Console Port (HTTP_TRANSPORT_ ADMIN)	9090	
HTTPS Admin Console Secure Port (HTTPS_TRANSPORT_ ADMIN)	9043	
Internal JMS Server (JMSSERVER_SECURITY_ PORT)	5557	server.xml
JMSSERVER_QUEUED_ ADDRESS	5558	serverindex.xml
JMSSERVER_DIRECT_ ADDRESS	5559	
BOOTSTRAP_ADDRESS	2809	
SOAP_CONNECTOR_ ADDRESS	8880	
DRS_CLIENT_ ADDRESS	7873	
SAS_SSL_SERVERAUTH_ LISTENER_ ADDRESS	0	
CSIV2_SSL_SERVERAUTH_ LISTENER_ ADDRESS	0	
CSIV2_SSL_MUTUALAUTH_ LISTENER_ ADDRESS	0	
ORB_LISTENER_ADDRESS	0*	
IBM HTTP Server Port	80	• virtualhosts.xml • plugin-cfg.xml
IBM HTTP Server Admin Port	8008	plugin-cfg.xml

* While the ORB_LISTENER_ADDRESS port is normally a dynamic port, it must be a static port when it is used in conjunction with a firewall. If you are using a firewall and running stand-alone EJB clients, change the ORB_LISTENER_ADDRESS port in the com.ibm.CORBA.Listener.Port property before configuring your firewall.

Migrating Web server configurations

This topic describes what to do to migrate a Web server from supporting IBM WebSphere Application Server, V3.5.x , to support IBM WebSphere Application Server, V5.

If you install the Web server and Application Server on different machines, you must install the binary plug-in module and the `plugin-cfg.xml` configuration file on the same machine as the Web server, as described in step 1.

If you migrate from a V3.5.x plug-in, and your Web server uses open servlet engine (OSE) transport, you must switch to HTTP transport when migrating to V5. Step 2 describes two ways to migrate from OSE transport to HTTP transport:

* Migrate all WebSphere Application Server machines from V3.5.x to V5 at one time, and migrate the Web server plug-in to support V5 only.
* Migrate some WebSphere Application Server machines from V3.5.x to V5, and migrate the Web server plug-in to support both versions of WebSphere Application Server at the same time.

Steps for this task

1. Install the IBM HTTP Server, V1.3.26.x and its plug-in, or a plug-in for another supported Web server from the product CD-ROM:

   * Install the HTTP Server and its plug-in on the same machine as the WebSphere Application Server, by selecting **IBM HTTP Server** and its plug-in, or select the plug-in for another supported Web server, on the feature selection page of the installation wizard.

   * Install the HTTP Server and its plug-in on a different machine from the WebSphere Application Server by following this procedure:

      a. Insert the product CD-ROM labeled, **WebSphere Application Server, IBM HTTP Server**, into the machine.

      b. Close the LaunchPad if it starts automatically.

      c. Change directory to the `IHS` directory on the product CD-ROM.

      d. Click the installation script for your platform, to install the IBM HTTP Server:

         – UNIX-based operating platforms: **InstallIHS.sh**
         – Windows: **InstallIHS.bat**

      This script installs the plug-in you need and makes the necessary configuration changes for the supported Web server.

   IBM HTTP Server, V1.3.26.x can coexist with earlier versions, or can upgrade V1.3.19 and later, to V1.3.26.x. Upgrading relieves you from having to uninstall and reinstall the HTTP server. Install V1.3.26.x into the same directory structure as V1.3.19 and later, to upgrade the earlier version. If you install the HTTP Server into a different directory, V1.3.26.x coexists with the previous version. By default, the administration server and the Web Server use the same ports as the previous version, which causes a conflict. However, you can change the port assignments on the coexistence panel of the WebSphere Application Server installation wizard.

   * Change the port number assignments for the new installation if you install into a separate directory. You can change port numbers on the coexistence panel. You can back track through the installation wizard and change the port settings if you have not already done so. Or, you can change the port settings after installation, in the `httpd.conf` file in the HTTP Server directory.

   * Update the IBM HTTP Server `httpd.conf` configuration entries to remove entries for earlier WebSphere Application Server versions if you install into the same directory as an earlier version.

      Both V4 and V5 of the WebSphere Application Server use the same HTTP transport plug-in binary module. If the Web server configuration file contains WebSphere Application Server V4.0.x plug-in

information, you must manually remove it. Otherwise, when the HTTP Server attempts to start the second V5 plug-in binary module, there is an error. The error indicates that the module is already loaded.

The configuration file might contain duplicate entries for accessing WebSphere Application Server Samples. Remove any aliases for previous versions and retain the V5 entries:

```
V3.5.x installations
Alias /IBMWebAS/  "C:\WebSphere\AppServer35\web\"

V4.0.x installations:
Alias /IBMWebAS/  "C:\WebSphere\AppServer40\web\"
Alias /WSsamples  "C:\WebSphere\AppServer40\WSsamples\"

V5 installation:
Alias /WSsamples "c:\Program Files\WebSphere\AppServer\WSsamples"
Alias /IBMWebAS/ "c:\Program Files\WebSphere\AppServer\web\"
```

2. Migrate plug-ins to work with IBM WebSphere Application Server, V5.

   - Migrate a plug-in that supports an earlier version of WebSphere Application Server to support only V5.

   - Migrate a plug-in that supports an earlier version of WebSphere Application Server to support both the earlier version and V5.

     This migration path includes instructions for enabling Web servers to work with both V3.5.x and V5 at the same time, or V4.x and V5 at the same time. This lets you migrate applications from the V3.5.x or V4.x Application Server to the V5 Application Server, without changing the Web server.

You now understand what you must do to install or migrate a Web server and plug-in to work with WebSphere Application Server, including installing the Web server on the same machine with the Application Server, or on different machines.

Migrating plug-ins on all nodes at once

If you migrate all earlier versions of the WebSphere Application Server to V5 at the same time, you can migrate all plug-ins at the same time and update the Web server to use the new binary plug-in and `plugin-cfg.xml` file. Migrating plug-ins on all nodes simultaneously is easier than migrating nodes one at a time.

Steps for this task

1. Verify that the Web server installation is at the supported level.

   If you are unsure of the level of the supported Web server product, see the WebSphere Application Server supported hardware, software, and APIs Web site for current requirements.

2. Install the correct binary plug-in module and an initial `plugin-cfg.xml` file on the Web server node.

   - If the Web server and Application Server are installed on the same node and you select the appropriate Web server plug-in during Application Server installation, the installation wizard configures the Web server configuration file for you, by adding the current entries.

   - Otherwise, insert the WebSphere Application Server CD-ROM into the Web server machine.

     – To update IBM HTTP Server and its plug-in, close the LaunchPad, if it appears. Browse the disk to select the **InstallIHS.bat** or **InstallIHS.sh** command from the `IHS` directory in the `platform_root` of the CD. These programs update the Web server and the binary plug-in.

     – To install the correct plug-in for another supported Web server, launch the installation program and select the **Custom installation** type. Select only the necessary Web server plug-in feature for installation. This selection installs the binary plug-in module and the initial `plugin-cfg.xml` file, and makes necessary configuration changes for the Web server. The installation wizard does not delete the previous entries for the load module (binary file) and the plug-in configuration line. You must do this migration manually as described in Preparing to install and configure a Web server.

3. Migrate the nodes from V3.5.x to V5, or from V4.x to V5.

Select the migration option during the installation procedure. Or, manually migrate a V3.5.x or 4.x installation to a previously installed V5 installation.

4. Regenerate the plug-in configuration file (trun_app_regen in the InfoCenter).

   • If the Web server and Application Server are running on the same node, further changes to the plug-in configuration file are probably not necessary.

   • Otherwise, edit the plug-in (trun_plugin in the InfoCenter) to make sure that the hostname attribute for the transports is set to the host name or IP address on which the Application Server is running.

5. Move the plugin-cfg.xml file (trun_plugin_location in the InfoCenter) to the correct location on the Web server.

Migrating plug-ins, one machine at a time

It is possible to run the V3.5.x Open Servlet Engine (OSE) plug-in and the V5 HTTP plug-in, in the same Web server installation. This ability lets a cluster of V3.5.x machines and a cluster of V5 machines coexist as you migrate machines one at a time from V3.5.x to V5. It is also possible to run the V4.x HTTP plug-in and the V5 plug-in in the same Web server installation.

The most direct way to migrate a supported Web server is to let the installation wizard automatically update the Web server configuration file, as you use the wizard to install the plug-in on the Web server machine. The installation wizard adds the current entries to the Web server configuration file but does not delete the previous entries for the earlier version. You must do this migration manually, as described in Preparing to install and configure a Web server.

Instructions for manual, incremental migration vary by Web server product and by platform as described below.

Steps for this task

1. Edit the system path environment variable on Windows platforms, to add the directory that contains the plug-in binary files that you copied from the Application Server. Otherwise, the Web server plug-ins do not load.

2. Reboot the computer after you update the system path, if necessary.

3. Verify that each machine in the routing rules for the plug-in has a unique URI.

   For example, you cannot have /servlet/* defined in both the V3.5.x OSE properties file and the V5 plug-in-cfg.xml configuration file. When there is duplication, operating behavior is erratic.

4. Migrate the Web server configuration and binary plug-in to the version supported by V5 of WebSphere Application Server.

   See one of the following topics in the online InfoCenter for more information about migrating Web server plug-ins for both V3.5.x and V4.x:

   • Migrating IBM HTTP Server to support multiple WebSphere Application Server versions (tins_websIHS in the InfoCenter)

   • Migrating the Sun ONE (iPlanet) Web server to support multiple WebSphere Application Server versions (tins_websIPl in the InfoCenter)

   • Migrating Lotus Domino to support multiple WebSphere Application Server versions (tins_websDom in the InfoCenter)

   • Migrating Microsoft Internet Service Manager to support multiple WebSphere Application Server versions (tins_websIIS in the InfoCenter)

Repeat this procedure for each Web server machine that you must migrate.

Automatically restarting server processes

There are several server processes that the operating system can automatically restart when the server processes stop abnormally. To set up this function on a UNIX-based operating system, you must have root authority to edit the inittab. On a Windows operating system, you must belong to the Administrator group and have the advanced user rights *Act as part of the operating system* and *Log on a service* to define Windows services. The installation wizard grants you the advanced user rights, if your user ID is part of the Administrator group. It displays a message that states that your user ID will have the rights the next time you log on.

There are various environments where you might use this function of automatically restarting servers. You can restart the **server1** server process in a stand-alone WebSphere Application Server environment, for example. Here is a list of processes you might consider restarting:

- The **server1** process on a stand-alone Application Server
- The **IBM HTTP Server** process
- The **IBM HTTP Administration** process
- The **WebSphere Embedded Messaging Publish And SubscribeWAS_*node_name*_jmsserver** process, if the Application Server node has the embedded messaging feature
- The **WebSphere Embedded Messaging Publish And SubscribeWAS_*node_name*_server1** process, if the Application Server node has the embedded messaging feature

You can create Windows services during installation, using the installation wizard. The wizard lets you create services for these servers:

- The **server1** process in a stand-alone base product environment, defined as a manually started (versus automatic) service
- The **IBM HTTP Server** process and the **IBM HTTP Administration** process, defined as automatically started services when you choose to install the IBM HTTP Server feature during the base product installation

You must manually create a shell script that automatically starts any of the processes previously mentioned, on a UNIX-based operating system. Each Windows service or UNIX shell script controls a single process, such as a stand-alone WebSphere Application Server instance. Multiple stand-alone Application Server processes require multiple Windows services or UNIX scripts, which you can define.

If you do not install the WebSphere Application Server base product as a Windows service during installation, you can use the **WASService.exe** command line tool, in the `install_root/bin` directory, to do so at a later time. You can use the tool to add any WebSphere Application Server process as a Windows service. The operating system can then monitor each server process, and restart the server if it stops.

Steps for this task

1. **(Optional) Use the installation wizard** to set up a Windows service to automatically monitor and restart processes related to the WebSphere Application Server product.
    - Perform the following procedure from the installation wizard for the base WebSphere Application Server product, to select services that the installation wizard can set up:
        a. Click **Run WebSphere Application Server as a service**.

        If you select this option, the installation wizard creates the following service during the installation:

        ```
        IBM WebSphere Application Server V5 - server1
        ```

        The IBM WebSphere Application Server V5 - server1 service controls the **server1** process in a stand-alone environment.

After you complete and verify the installation, use the Windows NT Services Control panel, or the Windows 2003 Services panel to change the **IBM WebSphere Application Server V5 - server1** service to an automatic startup type.

1) Right click **IBM WebSphere Application Server V5 - server1** and click **Properties**.

2) Click **Automatic** from the **Startup type** list box and click **OK**.

b. Click **Run IBM HTTP Server as a service**.

Select this option on the machine where you are installing the IBM HTTP Server.

If you select this option, the installation wizard creates the following services during the installation:

– **IBM HTTP Server 1.3.26.x**

– **IBM HTTP Administration 1.3.26.x**

The installation wizard defines the startup type of these services as **automatic**. It is not necessary that you do so.

c. Enter your user ID and password and click **Next**.

In a coexistence environment, you can change the default service names to make them unique. In a same version coexistence scenario for IBM HTTP Server 1.3.26.x or 2.0 on a Windows platform, you cannot use the default service names created by the installer because they are common.

To work around this problem:

a. Install the first copy of IBM HTTP Server, either by itself or with WebSphere Application Server and select to install the services.

b. Customize the service names for the first install by running the following commands from the first install location:

```
apache -k install -n "IHS 1.3.26.x(1)"
    apache -k install -f conf\admin.conf -n "IHS 1.3.26.x Administration (1)"
```

c. Edit the AdminAlias directive in the `installLocation 1\conf\admin.conf` file to point to the new service name, such as **IHS 1.3.26**.*x***(1)**.

d. Remove the default service names installed by the first install by running

```
apache -k uninstall -n "IBM HTTP Server 1.3.26.x"
    apache -k uninstall -n "IBM HTTP Administration 1.3.26.x"
```

e. Install the second copy of IBM HTTP Server, either by itself or with WebSphere Application Server. The default service names will now correspond to the second install.

Customized service names must be unique on your system.

2. **(Optional) After installing**, use the `WASService.exe` utility in the `bin` directory of the installation root to manually define a Windows service for another installation or configuration instance of the base WebSphere Application Server product.

You can also use the **net start** and **net stop** commands to control the IBM HTTP Server services on a Windows system. For more information about these commands, see the Windows help file. Access these commands from the Start menu, clicking **Start > Programs > IBM HTTP Server**.

You can also use the **Start the Server** and **Stop the Server** commands to control the IBM WebSphere Application Server on a Windows system. Access these commands from the Start menu, clicking **Start > Programs > IBM WebSphere > Application Server V5.0**.

Processes started by a **startServer** command are not running as monitored processes, regardless of whether you have configured them to be.

For example, you can configure a base Application Server as a WebSphere Application Server Windows service. However, if you start the Application Server instance using the **startServer** command, the

Windows system does not monitor or restart the Application Server because it was not started as a Windows service. The same is true on UNIX-based platforms. You must start the server process with a shell script based on the example rc.was script, to have the server running as a monitored process.

WASService command

The **WASService** command line tool lets you add any WebSphere Application Server server process as a Windows service. WebSphere Application Server server processes that you could add as Windows services include:

- Any base Application Server, such as the default **server1** process
- The **jmsserver** process that exists on a base Application Server node when you have installed the embedded messaging server feature and the node is part of a deployment manager cell
- The **nodeagent** process on a base Application Server node that is part of a deployment manager cell
- The deployment manager process, **dmgr**, on the Network Deployment node

The WASService.exe command file is located in the bin subdirectory of the installation root directory.

Starting an existing service

```
WASService.exe [-start] service_name
```

Creating a service

```
WASService.exe -add service_name
                -serverName Server_name
                    [-wasHome  install_root]
                    [-configRoot configuration_repository_directory]
                    [-startArgs additional_start_arguments]
                    [-stopArgs additional_stop_arguments]
                    [-userid execution id -password password ]
                    [-logFile service log file]
```

Deleting a service

```
WASService.exe -remove service_name
```

Stopping a running service

```
WASService.exe -stop service_name
```

Retrieving service status

```
WASService.exe -status service_name
```

Parameters

-add *service_name*
>   Creates a service named *service_name*.

-configRoot *configuration_repository_directory*
>   Optional parameter that identifies the configuration directory of the installation root directory of a WebSphere Application Server product. Versions prior to V5 let you specify a configuration directory. Deprecated function for V5 and later.

-logFile *service_log_file*
>   Optional parameter that identifies a log file that the **WASService** command uses to record its activity.

-remove *service_name*
>   Deletes the specified service.

-server *Server_name*
Identifies the server that the service controls.

-start *service_name*
Starts the existing service. The -start parameter is the default parameter; you can omit the -start parameter and still start the service.

-startArgs *additional_start_arguments*
Optional parameter that identifies additional parameters.

-status *service_name*
Returns the current status of the service, which includes whether the service is running or stopped.

-stop *service_name*
Stops the specified service.

-stopArgs *additional_stop_arguments*
Optional parameter that identifies additional parameters.

-userid *execution_ID -password password*
Optional parameters that identify a privileged userid and password.

-wasHome *install_root*
Optional parameter that identifies the installation root directory of the WebSphere Application Server product.

Creating an Application Server service

This example creates a service called `IBMWAS5Service - server2` that starts an Application Server process:

```
WASService -add server2
          -servername server2
          -wasHome "D:\Program Files\WebSphere\AppServer"
          -logfile "D:\Program Files\WebSphere\AppServer\logs\dmgr\startServer.log"
          -restart true
```

Creating multiple V5 configuration instances

Multiple configuration instances, as it applies to WebSphere Application Server products, is the ability to install the base WebSphere Application Server product once, and to use the **wsinstance** command to create multiple ***instances of it***, all running on the same machine at the same time. There is no support for creating configuration instances on remote machines.

In contrast to multiple configuration instances is *coexistence*, which refers to multiple *installations* of WebSphere Application Server, running on the same machine at the same time. Both coexistence and multiple configuration instances of the base WebSphere Application Server product imply various combinations of Web server interaction.

You can find a description of coexistence in Coexistence support.

This section contains the following topics:
- Procedure for creating configuration instances
- The wsinstance command

Procedure for creating configuration instances

You can install the base WebSphere Application Server one time and use the wsinstance tool in the `install_root\bin\wsinstance` folder to create multiple configuration instances. Each WebSphere Application Server configuration instance is a stand-alone instance with a unique name, and its own set of configuration files and user data folders. Configuration folders include `config`, `etc` and `properties`. User

data folders include `installedApps`, `installableApps`, `temp`, `logs`, `tranlog` and `wstemp`. Each instance also has the administrative console application to manage the configuration instance. Each instance shares all run-time scripts, libraries, the Software Development Kit, and other files with the initial Application Server.

You can configure and operate each configuration instance independently of other instances.

These limitations apply to multiple configuration instances:

- You cannot use the **Uninstall.exe** or **uninstall** command (or the **uninstall.sh** command on Linux for S/390 platforms) to remove a configuration instance. You must remove configuration instances with the -delete option of the **wsinstance** command, before uninstalling the installation instance.

- You cannot migrate the configuration and applications from an earlier release to a V5 configuration instance.

Reasons to use configuration instances include:

- You install the product only once.

- You can have stand-alone environments when installing the full product multiple times is not an acceptable solution because of space or procedural constraints.

- You can have a centralized development environment that is easy to maintain, where multiple developers and testers share one machine.

- You can create identical environments, where everyone is developing, testing, or performing in an identical workspace.

Use the following procedure to create and configure multiple V5 instances.

Steps for this task

1. Install the base WebSphere Application Server product.

2. Use the **wsinstance** command in the `WAS_HOME\bin\wsinstance` folder, to create a configuration instance of the server you installed in step 1.

   Refer to the description of the **wsinstance** command to learn more about the command, and to see examples of use.

   You must specify a unique directory path and a unique node name for each configuration instance.

   You must also specify unique port numbers for each configuration instance. For example, on a Windows 2003 platform, specify ports beginning at `20002`, for node shasti, in configuration instance_root `G:\shasti\WebSphere`, on the planetjava machine, by issuing this command:

   ```
   wsinstance.bat -name shasti
        -path G:\shasti\WebSphere
        -host planetjava
        -startingPort 20002
        -create
   ```

   This command creates a separate set of configuration and other data files.

3. Run the **setupCmdLine.bat** (or **setupCmdLine.sh**) script in the `bin` directory of the `instance_root` folder to set the WebSphere Application Server environment to the configuration instance.

   Or, you can set WAS_USER_SCRIPT to `instance_root\bin\setupCmdLine.bat`, which has the same effect as running the **setupCmdLine** command.

   On Linux and UNIX-based platforms, *source* the script to the parent shell to inherit the exported variables by running this command:

   ```
   # . ./setupCmdLine.sh
   ```

   Or, you can set WAS_USER_SCRIPT to `. instance_root/bin/setupCmdLine.sh`, which has the same effect as running the **setupCmdLine** command. For example:

   ```
   # export WAS_USER_SCRIPT=. /opt/inst1/bin/setupCmdLine.sh
   ```

After completing this step, you can start the Application Server configuration instance with the **startManager** or **startServer** command.

If you are using the embedded messaging feature and you are logging on as a non-root user on a UNIX-based or Linux platform, you must add the user to the `mqm` and `mqbrkrs` operating system groups as described in Installing WebSphere embedded messaging as the JMS provider.

4. **(Optional)** Associate a Web server with a configuration instance.

    If you create multiple configuration instances of the base WebSphere Application Server product, you can configure each instance to either share a Web server with other configuration instances, or to use a separate Web server.

    - **Share the Web server**
        a. Generate a `plugin-cfg.xml` configuration file for each Application Server configuration instance that is to share a Web server.
        b. Manually merge all `plugin-cfg.xml` configuration files into one file of the same name, `plugin-cfg.xml`.

            Put the file in the default folder for the first configuration instance, such as:

```
instance_root/WebSphere/AppServer/config/cells/plugin-cfg.xml
```

            If you use the same server name in multiple instances, (for example, suppose there are two server1 entries), the merged file has duplicate entries that you must address. All requests for either server1 instance are routed to the last one in the merged list.

            To correct the problem, change duplicate server name entries to make them unique in the PrimaryServers stanza:

```
<PrimaryServers>
 <Server Name="server1"/>
 <Server Name="server1a"/>
</PrimaryServers>
```

            Change each duplicate name in the merged file to match its unique name in the PrimaryServers stanza.
        c. Edit the Web server configuration file on the Web server as described in Manually configuring supported Web servers to point to the `plugin-cfg.xml` file.

        Repeat this process to add or remove a configuration instance from the configuration that shares the Web server, or when you regenerate a plug-in configuration file for any configuration instance in the group.

    - **Use a separate Web server**

        This configuration is simple. Each Application Server configuration instance uses a different Web server.
        a. Use the Web server documentation to create a new Web server instance for each configuration instance.

            For example, to install multiple images of the IBM HTTP Server, V1.3.26.*x*:

            1) Use the **InstallIHS** command script in the IHS directory on the product CD to install IBM HTTP Server, V1.3.26 for a second or later instance.
            2) Install the multiple installation instance to a new directory.
            3) Set a unique port for each multiple instance by editing the "port" directive in the `conf/httpd.conf` file in the IBM HTTP Server installation root.
            4) Apply any service to each multiple instance of the IBM HTTP Server using the silent interface of the WebSphere Application Server update installer:

```
updateSilent -ihsOnly -install -ihsInstallDir dir
```

5) Copy the current binary plug-in from the `install_root\bin` directory of the WebSphere Application Server product, to the HTTP Server machine and update the LoadModule `ibm_app_server_http_module` directive in the `httpd.conf` file accordingly.

6) Copy the `plugin-cfg.xml` file from the `install_root\config\cells` directory of the WebSphere Application Server product, to the HTTP Server machine and update the `WebSpherePluginConfig` directive in the `httpd.conf` file accordingly.

b. Modify the Web server configuration file, such as the `httpd.conf` file for IBM HTTP Server, to point to the configuration instance.

1) Specify the location of the WebSphere Application Server plug-in binary program. For example, when you install a plug-in feature, the installation wizard installs the plug-in binary program in the `install_root/bin` folder.

2) Specify the location of the `plugin-cfg.xml` file of the configuration instance. This file is located in the `config/cells` or `config\cells` folder of the configuration instance root location.

5. **(Optional)** Create additional servers in a multiple instance or coexistence environment.

6. Change port assignments in configuration files if you have a node that you cannot start because of port conflicts.

The wsinstance command

The **wsinstance.bat** (or **wsinstance.sh** for UNIX-based platforms) is a command line tool based on the base product ANT Utilities feature, which is required. The wsinstance.bat command line tool creates multiple configuration instances of one initial installation of the base WebSphere Application Server product.

Each WebSphere Application Server configuration instance is a stand-alone instance with a unique name, and its own set of configuration files and user data folders. Configuration folders include `config`, `etc` and `properties`. User data folders include `installedApps`, `installableApps`, `temp`, `logs`, `tranlog` and `wstemp`. It also has the administrative console application to manage the configuration instance. It shares all run-time scripts, libraries, the Software Development Kit, and other files with the initial Application Server.

The **wsinstance.bat** command file is located in the `wsinstance` subdirectory of the `bin` directory in the `install_root` of the product.

Syntax

Windows platforms:
```
wsinstance.bat  -name instanceName
                -path instanceLocation
                -host hostName
              [-startingPort startingPort]
               -create|-delete
               -debug
```

UNIX-based operating platforms:
```
wsinstance.sh  -name instanceName
               -path instanceLocation
               -host hostName
             [-startingPort startingPort]
              -create|-delete
              -debug
```

Parameters

Supported arguments include:

-name Specifies the instance name of the new configuration instance. Verify this value is unique. The

wsinstance command uses this name to construct the node name, which is *instanceName_hostname* for the new configuration instance of an Application Server.

-path   Specifies the file path of the instance. All required folders for the instance are in this directory, which is unique to the configuration instance.

-host   Specifies the hostname, which is the name of the host on which you are creating the configuration instance. This should match the host name you specified during installation of the initial product.

-create|-delete

Specifies whether to create or delete the configuration instance.

-startingPort

Optional parameter. Specifies the starting port number for generating all ports for the configuration instance. If not specified, the **wsinstance** command uses default ports, or custom-defined ports from a file that you can create.

Example of startingPort parameter use

The **wsinstance** command generates an `instanceName_portdef.props` file in the `wsinstance` subdirectory of the `bin` directory in the `install_root` folder. The **wsinstance** command assigns port numbers in the file to the configuration instance as it creates the instance.

If you do not use the -startingPort parameter the first time you create a configuration instance, the **wsinstance** command adds one (1) to the default port numbers for the base WebSphere Application Server product. If you create two configuration instances without using the -startingPort parameter, both instances have the same, conflicting port numbers.

You can create the `instanceName_portdef.props` file manually with predefined ports. You do not have to specify the -startingPort parameter again. The **wsinstance** command reads an existing `instanceName_portdef.props` file, to use the port numbers specified in the file. This command lets you manually create the file and specify the port numbers, before creating the configuration instance.

Use the template file, `portdef.props`, to create a new `instanceName_portdef.props` file, before creating the new configuration instance.

The following example `shasti_portdef.props` file is created with this command:

```
wsinstance.bat -name shasti
    -path G:\shasti\WebSphere
    -host planetnt
    -startingPort 20002
    -create

HTTPS_TRANSPORT_ADMIN=20002
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=20004
HTTP_TRANSPORT_ADMIN=20003
HTTP_TRANSPORT=20005
HTTPS_TRANSPORT=20006
INTERNAL_JMS_SERVER=20007
BOOTSTRAP_ADDRESS=20008
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=20009
DRS_CLIENT_ADDRESS=20011
SOAP_CONNECTOR_ADDRESS=20010
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=20012
JMSSERVER_QUEUED_ADDRESS=20013
JMSSERVER_DIRECT_ADDRESS=20014
```

Examples for Windows systems

An example of creating a configuration instance for user `shasti` follows:

```
wsinstance.bat -name shasti
     -path G:\shasti\WebSphere
     -host planetnt
     -create
```

On a base WebSphere Application Server product installation, the command creates an Application Server configuration instance, named `shasti`, with a node name of `shasti_planetnt`.

Examples for UNIX-based platforms

Example of creating a configuration instance for user `shasti`:

```
wsinstance.sh -name shasti
     -path /usr/shasti/WebSphere
     -host planetaix
     -create
```

On a base WebSphere Application Server product installation, the command creates an Application Server configuration instance, named `shasti`, with a node name of `shasti_planetaix`. If user shasti is to use the embedded messaging feature, you must add the user (shasti) to the mqm and mqbrkrs operating system groups, as described in ″Installing WebSphere embedded messaging as the JMS provider″. This is true for any non-root user on a UNIX-based or Linux platform.

Example of a creating a configuration instance for user `shasti` in a multiuser environment:

1. Create the configuration instance:

```
>wsinstance.sh -name shasti
     -path ~shasti/WebSphere
     -host myhost
     -startingPort 12000
     -create
```

2. Change the owner of the folder:

```
>chown shasti ~shasti/WebSphere
```

3. Add a call to script . `~shasti/WebSphere/bin/setupCmdLine.sh` in the profile of user `shasti` to set the environment when he logs in. User `shasti` can go directly into the `WebSphere_install_root/bin` location and start the server.

4. Give these folder permissions to user `shasti`:

```
install_root/bin, java, --- rx(read, execute)
install_root/properties, deploytool, config, lib, classes, null, samples, Web    ----rx(read,execute)
```

Example of deleting a configuration instance

The following command deletes the configuration instance named shasti:

```
wsinstance.sh -name shasti -host planetaix -delete
```

Creating servers in coexistence or multiple instance environments

WebSphere Application Server lets you create multiple servers based on an existing template, or using an existing server as a template. You can generate unique ports for the new server during its creation. Always select the unique port option when creating servers in a coexistence environment of multiple versions, installations or configuration instances, due to the likelihood of conflicting port assignments. Verify port assignments for the newly created server and change them if necessary.

Steps for this task

1. Create a server (MyServer, for example) using either the administrative console (trun_svr_create in the InfoCenter) or wsadmin scripts.
2. Log on to the administrative console.
3. Go to **Servers > Application Servers > MyServer > End Points** (urun_rsvr in the InfoCenter).
4. Go into each end point and change the port numbers.
5. Change the HTTP Transport ports.
   a. Go to **Servers > Application Servers > MyServer > Web Container > HTTP Transports** (urun_rhttptransport_prop in the InfoCenter).
   b. Change the HTTP transport port numbers.
   c. Make a record of the new port numbers.
6. Add new port numbers in the Virtual Hosts configuration (trun_plugin_vhost in the InfoCenter).
7. Change the JMS server ports, if you create servers from an Application Server template that has the embedded messaging feature.
   a. Go to **Servers > Application Servers > MyServer > Server Components > JMS Servers > Security Port Endpoint**.
   b. Change the port numbers.

Changing HTTP transport ports

This topic describes how to change HTTP transports manually by editing configuration files. Use this procedure when a conflicting HTTP transport setting is preventing an Application Server instance from starting.

Edit the configuration files to resolve conflicting port assignments, as described below.

Steps for this task

1. Look for symptoms of port number conflicts.

   Troubleshooting topics describe symptoms and ways to identify and fix possible port number conflicts. The following topics are among those described. Each topic has the article name in parentheses that you can use to search the InfoCenter.
   - Administration and administrative console troubleshooting tips (rtrb_admincomp)
   - Client program does not work (rtrb_clientprobs)
   - Debugging WebSphere Application Server applications (ttrb_debugwsa)
   - Errors after enabling SSL, or SSL-related error messages (rtrb_sslprobs)
   - Problems starting or using the **wsadmin** command (rtrb_wsadminprobs)
   - Troubleshooting migration problems (rtrb_migratnprobs)
   - Web Container troubleshooting tips (rtrb_webcntrcomp)
   - Web module or Application Server dies or hangs (rtrb_appdies)
   - Web resource (JSP file, servlet, HTML file, image) does not display (rtrb_pagedisplayprob)
2. **(Optional)** Edit an Application Server configuration.
   a. Open the `server.xml` configuration file for the `server1` Application Server.

      The file path for the server1 configuration, with a node name of **mynode**, is:

          install_root/config/cells/mynode/
                nodes/mynode/servers/server1
   b. Look for `transports xmi:type="applicationserver.webcontainer:HTTPTransport"`.

      The administrative console application uses transport ports 9090 and 9043. The sample applications use transport ports 9080 and 9443. Change the port numbers and save the file. Make a record of the new port numbers.
   c. Open the `virtualhosts.xml` file in the `install_root/config/cells/mynode` folder.

This file contains alias entries for transport ports defined in the `server.xml` file.

d. Look for `aliases xmi:id` to change port number assignments for any ports you changed.

Installing interim fixes and fix packs

The update installer application installs and uninstalls interim fixes and fix packs (also known as *fixpacks*, *FixPaks* and *program temporary fixes*, or PTFs) on WebSphere Application Server products. There are two interfaces to the installer application, a wizard with a graphical interface, and a command-line, silent interface.

This topic describes the proper procedure for installing an interim fix or a fix pack on a stand-alone IBM WebSphere Application Server, V5 environment, using the update installer application.

To apply an interim fix or fix pack, first set up and configure the environment by downloading the interim fix and the update installer, or downloading the fix pack (which includes the update installer), creating update repositories, and setting the JAVA_HOME environment variable. Then use the update installer to install the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also uninstall fixes and fix packs.

If the base WebSphere Application Server node is within a cell, open this topic in the *Getting Started* PDF for the Network Deployment product. If the base Application Server node is extended by the Enterprise product, open this topic in the *Getting Started* PDF for the Enterprise product. This topic describes applying an interim fix or fix pack to a stand-alone base node.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact fix and version level of the product. Do not use the **versionInfo** command while installing or uninstalling an interim fix or fix pack.

You can also use the silent update installer application to:
- View interim fix information
- View fix pack information

Before installing or uninstalling fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all Application Servers, and all servers, such as the IBMHttpServer process, that belong to serviceable features. Features with servers include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for updating a stand-alone base node by installing an interim fix or fix pack. The Removing interim fixes and fix packs topic describes how to remove an interim fix or fix pack from a stand-alone base node.

Installing a fix pack uninstalls all interim fixes. Uninstalled fixes at a later level than the fix pack are not included in the fix pack. Reinstall such fixes to bring your system back to the previous fix level.

Space requirements vary depending on what you are installing. The size of each download is available on the Support site. For a fix pack, have approximately 400 MB of free space in the `/tmp` directory and another 400 MB in the file system that hosts the WebSphere Application Server image (typically `/usr`) on a UNIX-based platform, or approximately 800 MB of free space on the disk drive where you are installing on a Windows platform.

Verify that the free space is available before beginning the installation. After unpacking the ZIP file, you can delete the ZIP file to free space if necessary. After it is installed, the Fix Pack 2 code increases the IBM WebSphere Application Server installation and run-time footprints by a small amount.

Space is required for the `/update` directory of the product installation root. The space required is about the same as the size of the fix pack, typically somewhere between 50 MB to 250 MB.

Space is also required for backup files in the `/properties/version/backup` directory of the product installation root. The space required is about the same as the size of the fix pack, somewhere between 50 MB to 250 MB, varying by product and platform.

Interim fixes require much less space to install.

Steps for this task

1. Stop each server process on the base WebSphere Application Server node with the **stopServer** command.

   Stop all WebSphere Application Server-related Java processes. On a Windows platform, use the task manager to stop Java processes. On a UNIX-based platform, use the **kill** command to stop Java processes.

2. Create an `install_root/update` directory, if it does not already exist.

3. Download the update installer application ZIP file to the `install_root/update` directory if you are installing an interim fix.

   Download the current version of the file even though you might have an update installer from a previous interim fix installation. The Support page links to the current installer.

4. Extract the contents of the zip file to the `update` directory.

5. Create the `update/fixes` repository if you are installing an interim fix. Unpacking the fix pack creates the `fixpacks` repository directory, if it does not already exist.

6. Download the interim fix or download and unpack the fix pack.

   Download an interim fix from the Support page to the `install_root/update/fixes` directory. Download a fix pack ZIP file to the `install_root/update` directory. Unpack the fix pack to automatically create the `fixpacks` directory.

   On Windows platforms, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

7. **Optional:** Set up the Java environment for the update installer.

   If the update installer can set the Java environment, this step is not necessary. Otherwise, this is a required step.

   The location of the `update`, `fixes` repository, and `fixpacks` repository directories is arbitrary. Create the directories anywhere. However, the `install_root/update`, `install_root/update/fixes`, and `install_root/update/fixpacks` locations are recommended.

   If you use a non-standard installation root, it is possible that the update installer cannot set the JAVA_HOME environment variable. If you receive a message that the update installer cannot set JAVA_HOME, set the environment variable yourself, or issue the appropriate command script yourself, from the `/bin` directory of the installation root:

   a. Open a command line window.

   b. Change directories to the `bin` directory of the installation root.

   c. Issue the appropriate command to set JAVA_HOME:

      - `. install_root/bin/setupCmdLine.sh` (source the command on UNIX platforms)
      - `source install_root/bin/setupCmdLine.sh` (source the command on Linux platforms)
      - `install_root\bin\setupCmdLine.bat` (Windows platforms only)
      - `. install_root/bin/setupClient.sh` (source the command for the Application Server client)

- source *install_root*/bin/setupClient.sh (Linux platforms only)
- *install_root*\bin\setupClient.bat (Windows platforms only)

8. Use one of the two interfaces to the update installer to apply the interim fix or fix pack to the base node:
   - Refer to**UpdateWizard** command for usage information.
   - Refer to the **UpdateSilent** command description for the proper syntax for installing the interim fix or fix pack:
     - Installing interim fixes
     - Installing fix packs

   For example, to install the `was50_fp2_win` fix pack, use this **updateSilent** command:

   ```
   C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -skipIHS
        -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
        -install
        -fixpackID was50_fp2_win
   ```

   The command is shown here on more than one line, for clarity.

9. Restart the node agent for the base node with the **startNode** command if the node is part of a cell.

   ```
   startnode
   ```

10. Restart each server on the node with the **startServer** command.

    ```
    startserver server1
    ```

    If the node is part of a cell:

    ```
    startserver jmsserver
    ```

11. Verify that the node is online and functioning correctly.

    There are several ways to verify the successful application of an interim fix or fix pack:
    - Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
    - Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
    - Is there an `[interim fixID].efix` or `[fix packID].ptf` file in the `install_root/properties/version/version` directory, or an `[interim fixID].efixApplied`, `[interim fixID].efixDriver`, `[fix packID].ptfApplied`, or `[fix packID].ptfDriver` file in the `install_root/properties/version/history` directory?
    - Do the product version and history reports show the fix or fix pack to be installed or removed?
    - Does collecting interim fix information and update state show that the interim fix is installed or removed?
    - Does collecting fix pack information and update state show that the fix pack is installed or removed?

    Run the installation verification tool on the node as described in the InfoCenter to verify that the node is operational.

UpdateSilent command

The **updateSilent** command is the silent, command-line interface to the IBM WebSphere Application Server update installer application. You can also use a wizard interface to the update installer application, the updateWizard command. The update installer installs and uninstalls fixes and fix packs for WebSphere Application Server products. This topic describes the silent interface to the update installer command, and its command-line parameters.

Stop all Java processes on the machine that use the IBM SDK that WebSphere Application Server provides Before installing or uninstalling fixes and fix packs on a machine, stop all Java processes on the machine that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all Application Server processes, the nodeagent process, the deployment manager process, and all server processes, such as the jmsserver process, that belong to serviceable features. Features with server processes include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process runs, IBM does not guarantee that the product can continue to run successfully, or without error.

Remove the WebSphere MQ tray icon if present On a Windows platform, remove the WebSphere MQ tray icon if it is present. The WebSphere MQ tray icon in the lower right corner indicates that a WebSphere MQ process (amqmtbrn.exe) is running. Right click the tray icon and click Hide to remove it.

Do not launch multiple copies of the update installer at one time The update installer cannot be launched concurrently with itself. Performing more than one update at the same time can lead to a failed or faulty installation.

Installation roots

The symbol *install_root* means the root directory for WebSphere Application Server. By default, this varies per product and operating system:

- Base WebSphere Application Server product:
  - AIX platforms: `/usr/WebSphere/AppServer`
  - Other UNIX and Linux platforms: `/opt/WebSphere/AppServer`
  - Windows platforms: *drive*`\Program Files\WebSphere\AppServer`
- Network Deployment product:
  - AIX platforms: `/usr/WebSphere/DeploymentManager`
  - Other UNIX and Linux platforms: `/opt/WebSphere/DeploymentManager`
  - Windows platforms: *drive*`\Program Files\WebSphere\ DeploymentManager`
- Enterprise product that extends the base product:
  - AIX platforms: `/usr/WebSphere/AppServer`
  - Other UNIX and Linux platforms: `/opt/WebSphere/AppServer`
  - Windows platforms: *drive*`\Program Files\WebSphere\AppServer`
- Enterprise product that extends the Network Deployment product
  - AIX platforms: `/usr/WebSphere/DeploymentManager`
  - Other UNIX and Linux platforms: `/opt/WebSphere/DeploymentManager`
  - Windows platforms: *drive*`\Program Files\WebSphere\DeploymentManager`

Space requirements

Space requirements vary depending on what you are installing. The size of each download is available on the Support site. After unpacking the ZIP file you download, delete the ZIP file to free space. For a fix pack, have approximately 400 MB of free space in the `/tmp` directory and another 400 MB in the file system that hosts the WebSphere Application Server image (typically `/usr`) on a UNIX-based platform, or approximately 800 MB of free space on the disk drive where you are installing on a Windows platform.

Verify that the free space is available before beginning the installation. After unpacking the ZIP file, you can delete the ZIP file to free space if necessary. After it is installed, the Fix Pack 2 code increases the IBM WebSphere Application Server installation and run-time footprints by a small amount.

Space is also required for backup files in the `install_root/properties/version/backup` directory. When installing a fix pack the space required is typically about the same as the size of the fix pack, that is, between 50 MB and 250 MB, depending on the particular fix pack.

Fixes require much less space to install.

The update installer checks for required space before it installs the fix pack. However, there is a bug in the space checker when installing Fix Pack 2.

When installing Fix Pack 2, the update installer checks for required disk space in the `/tmp` directory and in the `backup` directory. If there is not enough space in the `/tmp` directory or there is not enough space in the `backup` directory, the update installer issues a warning message.

A temporary problem exists where the update installer issues the warning for both directories when it should issue a warning message for the `/tmp` directory only. For example, if there is less than 248 MB in the `/tmp` directory and more than 248 MB in the installation root, messages similar to these appear:

```
"Insufficient disc space found in /tmp.
 This fix pack installation requires 248 megabytes."

"Insufficient disc space found in /WebSphere/AppServer/properties/version/backup.

 This fix pack installation requires 248 megabytes."
```

To fix the problem, allocate enough space in the `/tmp` directory. Neither message will appear.

Command name

updateSilent.sh and updateSilent.bat, command-line interface to the `installer.jar` file.

Related command

updateWizard.sh and updateWizard.bat, graphical interface to the `installer.jar` file.

Prerequisite environment setting

The JAVA_HOME environment setting.

If you use a non-standard installation root, it is possible that the **updateWizard** command cannot set the JAVA_HOME environment variable. If you receive a message that the wizard is unable to set JAVA_HOME, set the environment variable yourself, or issue the appropriate command script yourself, from the `/bin` directory of the installation root:

1. Open a command line window.
2. Change directories to the `bin` directory of the install_root.
3. Run the appropriate command:
   - . `install_root`/bin/setupCmdLine.sh (source the command on UNIX platforms)
   - source `install_root`/bin/setupCmdLine.sh (source the command on Linux platforms)
   - `install_root`\bin\setupCmdLine.bat (Windows platforms only)
   - . `install_root`/bin/setupClient.sh (source the command for the Application Server client)
   - source `install_root`/bin/setupClient.sh (Linux platforms only)
   - `install_root`\bin\setupClient.bat (Windows platforms only)

Download from

Download as `updateInstaller.zip` from the WebSphere Application Server Support page, or as part of each fix pack ZIP file package. Fix packs are named according to the Application Server product, the fix pack, and the operating system platform:

Table 13. Fix pack names for Fix Pack 1. (The spaces in each file name after each underscore are for print formatting purposes.)

Product	Operating system platform	Fix Pack 2 ZIP file	Fix Pack 2 ID	Default repository in installation root directory
Base WebSphere Application Server	AIX	was50_ fp2_ aix.zip	was50_ fp2_ aix	../update/ fixpacks
	HP-UX	was50_ fp2_ hpux.zip	was50_ fp2_ hpux	
	Linux	was50_ fp2_ linux.zip	was50_ fp2_ linux	
	Linux for S/390	was50_ fp2_ Linux for S/390.zip	was50_ fp2_ Linux for S/390	
	Solaris	was50_ fp2_ solaris.zip	was50_ fp2_ solaris	
	Windows	was50_ fp2_ win.zip	was50_ fp2_ win	..\update\ fixpacks
Network Deployment	AIX	was50_ nd_ fp2_ aix.zip	was50_ nd_ fp2_ aix	../update/ fixpacks Move the fix pack to a unique directory, such as ../update/ fixpacks/ nd, to improve performance when there is a base fix pack in the default directory.
	HP-UX	was50_ nd_ fp2_ hpux.zip	was50_ nd_ fp2_ hpux	
	Linux	was50_ nd_ fp2_ linux.zip	was50_ nd_ fp2_ linux	
	Linux for S/390	was50_ nd_ fp2_ Linux for S/390.zip	was50_ nd_ fp2_ Linux for S/390	
	Solaris	was50_ nd_ fp2_ solaris.zip	was50_ nd_ fp2_ solaris	
	Windows	was50_ nd_ fp2_ win.zip	was50_ nd_ fp2_ win	..\update\ fixpacks

Table 13. Fix pack names for Fix Pack 1 (continued). (The spaces in each file name after each underscore are for print formatting purposes.)

Product	Operating system platform	Fix Pack 2 ZIP file	Fix Pack 2 ID	Default repository in installation root directory
Enterprise	AIX	was50_ pme_ fp2_ aix.zip	was50_ pme_ fp2_ aix (to extend the base product)	`../update/ fixpacks` Move each fix pack to a unique directory, such as `../update/ fixpacks/ ent` and `../update/ fixpacks/ ent/nd`, to improve performance if there is another fix pack in the default directory.
			was50_ pme_ nd_ fp2_ aix (to extend the Network Deployment product)	
	HP-UX	was50_ pme_ fp2_ hpux.zip	was50_ pme_ fp2_ hpux (to extend the base product)	
			was50_ pme_ nd_ fp2_ hpux (to extend the base product)	
	Linux	was50_ pme_ fp2_ linux.zip	was50_ pme_ fp2_ linux (base)	
			was50_ pme_ nd_ fp2_ linux (Network Deployment)	
	Linux for S/390	was50_ pme_ fp2_ Linux for S/390.zip	was50_ pme_ fp2_ Linux for S/390 (base)	
			was50_ pme_ nd_ fp2_ Linux for S/390 (Network Deployment)	
	Solaris	was50_ pme_ fp2_ solaris.zip	was50_ pme_ fp2_ solaris (base)	
			was50_ pme_ nd_ fp2_ solaris (Network Deployment)	
	Windows	was50_ pme_ fp2_ win.zip	was50_ pme_ fp2_ win (base)	`..\update\ fixpacks`
			was50_ pme_ nd_ fp2_ win (Network Deployment)	
Express	AIX	was50_ express_ fp2_ aix.zip	was50_ express_ fp2_ aix	`../update/ fixpacks` Move the fix pack to a unique directory, such as `../update/ fixpacks/ express`, to improve performance if there is another fix pack in the default directory.
	HP-UX	was50_ express_ fp2_ hpux.zip	was50_ express_ fp2_ hpux	
	Linux	was50_ express_ fp2_ linux.zip	was50_ express_ fp2_ linux	
	Linux for S/390	was50_ express_ fp2_ Linux for S/390.zip	was50_ express_ fp2_ Linux for S/390	
	Solaris	was50_ express_ fp2_ solaris.zip	was50_ express_ fp2_solaris	
	Windows	was50_ express_ fp2_ win.zip	was50_ express_ fp2_ win	`..\update\ fixpacks`

Table 13. Fix pack names for Fix Pack 1 (continued). (The spaces in each file name after each underscore are for print formatting purposes.)

Product	Operating system platform	Fix Pack 2 ZIP file	Fix Pack 2 ID	Default repository in installation root directory
WebSphere Application Server client	AIX	was50_ client_ fp2_ aix.zip	was50_ client_ fp2_ aix	../update/ fixpacks
	HP-UX	was50_ client_ fp2_ hpux.zip	was50_ client_ fp2_ hpux	Move the fix pack to a unique directory, such as ../update/ fixpacks/ client, to improve performance if there is another fix pack in the default directory.
	Linux	was50_ client_ fp2_ linux.zip	was50_ client_ fp2_ linux	
	Linux for S/390	was50_ client_ fp2_ Linux for S/390.zip	was50_ client_ fp2_ Linux for S/390	
	Solaris	was50_ client_ fp2_ solaris.zip	was50_ client_ fp2_ solaris	
	Windows	was50_ client_ fp2_ win.zip	was50_ client_ fp2_win	..\update\ fixpacks

Download to

The default location for unpacking the update installer or fix pack zip file is the `install_root/update` directory. Unpacking a fix pack creates the `../update/fixpacks` directory. Create another directory, `../update/fixes`, for a repository of fixes you download. If you use these default subdirectories, accept default interim fix and fix pack file locations when using the **updateWizard** interface. Otherwise, browse to locate the interim fixes or fix packs you are installing or uninstalling.

Create an `update` directory if it does not exist, download the update installer application ZIP file from the Support site to the `update` directory, and unzip the file.

On Windows platforms, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

Location of extfile.jar

WebSphere Application Server product `install_root/update/lib` (or `install_root\update\lib` for Windows platforms)

Location of installer.jar, readme_ptf.html, updateSilent.sh/bat, and updateWizard.sh/bat

WebSphere Application Server product `install_root/update` (or `install_root\update` for Windows platforms)

Location of fix Java archive (JAR) files

`../update/fixes` (or `..\update\fixes`)

Location of fix pack JAR files

`../update/fixpacks` (or `..\update\fixpacks`)

This directory is the location after unpacking the fix pack in the `install_root/update` directory.

Files in updateInstaller.zip

Always use the **updateSilent** (or **updateWizard**) command file from the `updateInstaller.zip` or fix pack you download, to use the most recent version. Files in the `updateInstaller.zip` package (or the fix pack ZIP package) include:

- `extfile.jar`
- `installer.jar`
- `readme_updateinstaller.txt`
- `readme_updateinstaller.html`

- `readme_updateinstaller.pdf`
- `updateSilent.sh` (or `updateSilent.bat`)
- `updateWizard.sh` (or `updateWizard.bat`)

In addition to the files listed, the fix pack ZIP file also has the fix pack JAR file, such as the `was50_fp2_win.jar` file. Each JAR file includes a fix pack.

Location of log and backup files

The update installer records processing results in log files in the `install_root/properties/version/log` directory. Backup files created during the installation of interim fixes and fix packs are in the `install_root/properties/version/backup` directory. The files are required to uninstall an interim fix or fix pack.

The directory that contains update installer logs is changed, beginning with the version of the update installer that is bundled with Fix Pack 2. Log files are in the `install_root/logs/update` directory.

Syntax examples

The silent update application actually provides two functions. Depending upon the parameters you choose, the command:

- Installs and uninstalls interim fixes and fix packs
- Provides information about the update state of interim fixes and fix packs you apply

The following examples describe the syntax of various uses of the update installer. In each example, optional parameters are enclosed by brackets ([]). Values that you supply appear in *italicized font*. Choices are denoted by the pipe symbol (|).

Help:
```
updateSilent  -help | -? | -usage

updateSilent  /help | /? | -usage (Windows platforms)
```

Use a properties file to supply values:
```
updateSilent myProps.properties
```

Fix processing:
```
updateSilent -installDir "fully qualified product install_root"
          -fix
          -fixDir "fully qualified fix repository root,
                usually install_root/update/fixes"
          -install | -uninstall | uninstallAll
          -fixes space-delimited list of fixes
          -fixJars space-delimited list of fix JAR files
        [-fixDetails]
        [-prereqOverride]
```

View applied fixes:
```
updateSilent -fix
          -installDir "fully qualified product install_root"
```

View available fixes:
```
updateSilent -fix
          -installDir "fully qualified product install_root"
          -fixDir "fully qualified fix repository root,
                  usually install_root/update/fixes"
```

Fix pack processing:

```
updateSilent -installDir "fully qualified product install_root"
            -fixpack
            -fixpackDir "fully qualified FixPack repository root,
                    usually install_root/update/fixpacks"
            -install | -uninstall
            -fixPackID fix pack ID
            [-skipIHS | [-ihsOnly] -ihsInstallDir fully qualified IBM HTTP Server root]
            [-skipMQ | -mqInstallDir embedded messaging feature root]
            [-includeOptional space-delimited list of components]
            [-fixpackDetails]
```

All other valid arguments are ignored, such as the `prereqOverride` argument, which is for fix processing only.

You need not supply the `-mqInstallDir` argument for AIX, Linux, and UNIX-based platforms. The install location is fixed on those operating platforms. Use the argument on Windows platforms. The default location on Windows platforms is the `C:\Program Files\IBM\WebSphere MQ` directory.

View applied fix packs:

```
updateSilent -fixpack
            -installDir "fully qualified product install_root"
```

View available fix packs:

```
updateSilent -fixpack
            -installDir "fully qualified product install_root"
            -fixpackDir "fully qualified fix pack repository root,
                    usually install_root/update/fixpacks"
```

Parameters

Use the following parameters for the **updateSilent** command:

-?   Shows command usage.

/?   Shows command usage on Windows platforms only.

-fix   Interim fix only: Identifies the update as an interim fix update.

-fixDetails
    Interim fix only: Displays interim fix detail information.

-fixDir   Interim fix only: Specifies the fully qualified directory where you download interim fixes. This directory is usually the `install_root`/update/fixes directory.

-fixes   Interim fix only: Specifies a list of space-delimited interim fixes to install or uninstall.

-fixJars
    Interim fix only: Specifies a list of space-delimited interim fix JAR files to install or uninstall. Each JAR file has one or more interim fixes.

-fixpack
    Fix pack only: Identifies the update as a fix pack update.

-fixpackDetails
    Fix pack only: Displays fix pack detail information.

-fixpackDir
    Fix pack only: Specifies the fully qualified directory where you download and unpack fix packs. By default, this directory is the `install_root`/update/fixpacks directory.

-fixpackID

Fix pack only: Specifies the ID of a fix pack to install or uninstall. The value you specify does not include the `.jar` extension. The value is not the fully qualified package file name, but is the name of the individual fix pack within the JAR file.

The current Application Server strategy for fix pack JAR files is to use one JAR file per fix pack. The fix pack ID is the name of the JAR file before the `.jar` extension. For example:

- **fix pack ID:** `was50_fp2_linux`
- **fix pack JAR file name:** `was50_fp2_linux.jar`
- **fix pack ZIP file name:** `was50_fp2_linux.zip`

-help   Shows command usage.

/help   Shows command usage on Windows platforms only.

-ihsInstallDir

Fix pack only: Specifies the fully qualified path of the IBM HTTP Server product, and applies any service for the IBM HTTP Server product that might exist in the fix pack.

-ihsOnly

Fix pack only: Skips the installation of all service but that for the IBM HTTP Server product, and applies any service for the IBM HTTP Server product that might exist in the fix pack. Requires the `-ihsInstallDir` parameter.

-includeOptional

Fix pack only: Specifies a space-delimited list of features. The installer applies any service for the components, if present in the fix pack. Otherwise, the installer does not apply the service.

-install

Installs the update type.

-installDir

Specifies the fully qualified installation root of the WebSphere Application Server product.

-mqInstallDir

Fix pack only: Specifies the fully qualified installation root of the embedded messaging feature, which is based on WebSphere MQ technology.

-prereqOverride

Interim fix only: Overrides any installation and uninstallation prerequisite checking. The update installer does not log missing prerequisites.

<propertyFile>.properties

Specifies an externally supplied parameters file.

You can supply parameters in an external `.properties` file, rather than directly on the command line. There are some differences in the formats of parameters:

- Parameters are [name]=[value] pairs.
- Lists of parameter values are comma-delimited instead of space-delimited.
- There are two slashes before directory names.

Use the `.properties` file template included as part of the update installer download.

For example, a `sample.properties` file for installing two fixes might look like this:

```
#Sample.properties
   #Sample parameters file to install fixes with details and prerequisite override
   fix=true
   install=true
   installDir=C:\\WebSphere\\AppServer
```

```
fixDir=C:\\WebSphere\\AppServer\\update\\fixes
fixes=Fix1,Fix2
fixDetails=true
prereqOverride=true
```

A `sample.properties` file for installing a fix pack might look like this:

```
#Sample.properties
#Sample parameters file to install a fix pack with details
install=true
installDir=C:\\WebSphere\\AppServer
fixpackDir=C:\\WebSphere\\AppServer\\update\\fixpacks
fixpackID=was50_fp2_win
fixpackDetails=true
ihsInstallDir=C:\\IBMHttpServer
```

-skipIHS

> Fix pack only: Skips any optional service for the IBM HTTP Server product that might exist in the fix pack.

> If you installed the IBM HTTP Server product as a feature, use the update installer to update it with service in an interim fix or fix pack. Otherwise, download an updated IBM HTTP Server product and install it into the same directory as your existing version, to update the existing installation. You can also uninstall the current version and install the downloaded version to avoid any issues with migration.

> Update your configuration if you reinstall. The process is described in the *Manually configuring supported Web servers* (tins_manualWebserver) topic in the InfoCenter for the base Application Server.

-skipMQ

> Fix pack only: Skips any optional service for the embedded messaging feature (which is based on the IBM WebSphere MQ product) that might exist in the fix pack.

> Always apply any outstanding corrective service to the stand-alone IBM WebSphere MQ product if you have it, before using the WebSphere Application Server update installer to update the embedded messaging feature with service in an interim fix or fix pack. Skip the installation of service to the embedded messaging feature if you install corrective service to the stand-alone IBM WebSphere MQ product.

-uninstall

> Specifies to uninstall the identified fix or fix pack.

-uninstallAll

> Interim fix only: Specifies to uninstall all applied interim fixes. This parameter does not uninstall fix packs.

> You must uninstall all interim fixes and fix packs before uninstalling the base WebSphere Application Server product, the Network Deployment product, or the Enterprise product.

-usage

> Shows command usage.

Examples

The following examples assume that:

- The installation root is the `C:\Program Files\WebSphere\AppServer` directory.
- The location of the IBM HTTP Server feature is the `C:\Program Files\IBMHttpServer` directory.
- The fix repository is the `C:\Program Files\WebSphere\AppServer\update\fixes` directory.
- The fix pack repository is the `C:\Program Files\WebSphere\AppServer\update\fixpacks` directory.

Examples in this section include:

- Getting help for the command

- Using a parameter properties file
- Installing interim fixes
- Removing interim fixes
- Viewing information about interim fixes
- Installing fix packs
- Removing fix packs
- Viewing information about fix packs

Most of the examples are split into more than one line, for clarity.

Getting help for the command: To get help for the **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -help
```

Using a parameter properties file: To use the `myProps.properties` file to supply parameter values for the **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent myProps.properties
```

Installing interim fixes: To install a collection of interim fixes:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -install
        -fixes Fix1 Fix2
```

To install a collection of interim fixes, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -install
        -fixes Fix1 Fix2
        -fixDetails
```

To install a collection of fixes, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -install
        -fixes Fix1 Fix2
        -prereqOverride
```

To install interim fixes from a Java archive (JAR) file:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -install
        -fixJar Fix1
```

To install interim fixes from a Java archive (JAR) file, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -install
        -fixJar Fix1
        -fixDetails
```

To install interim fixes from a Java archive (JAR) file, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -install
        -fixJar Fix1
        -fixDetails
```

Removing interim fixes: To uninstall a collection of interim fixes:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -uninstall
        -fixes Fix1 Fix2
```

To uninstall a collection of interim fixes, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -uninstall
        -fixes Fix1 Fix2
        -fixDetails
```

To uninstall a collection of interim fixes, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -uninstall
        -fixes Fix1 Fix2
        -prereqOverride
```

To uninstall interim fixes in a Java archive (JAR) file:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -uninstall
        -fixJar Fix1
```

To uninstall interim fixes in a Java archive (JAR) file, and display interim fix details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -uninstall
        -fixJar Fix1
        -fixDetails
```

To uninstall interim fixes in a Java archive (JAR) file, and override prerequisite checking:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
        -uninstall
        -fixJar Fix1
        -fixDetails
```

Viewing information about interim fixes: To view a list of installed interim fixes:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
```

To view a list of interim fixes available in the repository:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fix
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixDir "C:\Program Files\WebSphere\AppServer\update\fixes"
```

Installing fix packs: To install a fix pack:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -ihsInstallDir "C:\Program Files\IBMHttpServer"
        -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
        -install
        -fixpackID was50_fp2_win
```

To install a fix pack, and display fix pack details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -ihsInstallDir "C:\IBMHttpServer"
        -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
        -install
        -fixpackID was50_fp2_win
        -fixpackDetails
```

To perform a partial installation of a fix pack, by choosing to skip the installation of optional service to the WebSphere Application Server embedded messaging feature, which is based on WebSphere MQ:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
        -ihsInstallDir "C:\Program Files\IBMHttpServer"
        -install
        -fixpackID was50_fp2_win
        -skipMQ
```

The fix pack status shows partial installation.

To perform a partial installation of a fix pack, by choosing to skip the installation of optional service to the IBM HTTP Server feature:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
        -mqInstallDir "C:\WebSphere MQ"
        -install
        -fixpackID was50_fp2_win
        -skipIHS
```

The fix pack status shows partial installation.

To perform a partial installation of a fix pack, by choosing to skip the installation of optional service to both the embedded messaging feature and the IBM HTTP Server feature:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
        -install
        -fixpackID was50_fp2_win
        -skipIHS
        -skipMQ
```

The fix pack status shows partial installation.

Removing fix packs: To uninstall a fix pack:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -uninstall
        -fixpackID was50_fp2_win
```

To uninstall a fix pack, and display fix pack details:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -uninstall
        -fixpackID was50_fp2_win
        -fixpackDetails
```

Viewing information about fix packs: To view a list of installed fix packs:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
```

To view a list of fix packs available in the repository for the base WebSphere Application Server product:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
        -installDir "C:\Program Files\WebSphere\AppServer"
        -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
```

UpdateWizard command

The **updateWizard** command is the wizard interface to the IBM WebSphere Application Server update installer application. You can also use a silent, command-line interface to the update installer application, the **updateSilent** command. The update installer installs and uninstalls interim fixes and fix packs for WebSphere Application Server products.

This topic describes the wizard interface to the update installer and gives information about its panels and fields.

Before installing or removing interim fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all Application Server processes, the nodeagent process, the deployment manager process, and all server processes, such as the jmsserver process, that belong to serviceable features. Features with server processes include the IBM HTTP Server and the embedded messaging feature.

Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

On a Windows platform, remove the WebSphere MQ tray icon if it present. The WebSphere MQ tray icon in the lower right corner indicates that a WebSphere MQ process (**amqmtbrn.exe**) is running. Right click the tray icon and click **Hide** to remove it.

Command name
>   `updateWizard.sh` and `updateWizard.bat`, graphical interface to the update installer (`installer.jar`) file

Related command
>   `updateSilent.sh` and `updateSilent.bat`, command-line interface to the update installer (`installer.jar`) file

Starting the wizard

The **updateWizard** command (*install_root*/update/updateWizard.sh and *install_root*\update\updateWizard.bat) launches the wizard interface to the update installer application.

For example:
```
install_root/update/updateWizard.sh
install_root\update\updateWizard.bat (Windows only)
```

Parameters

Apply zero, one, or more of these optional parameters in any order, by issuing the **updateWizard** command from the command line.

-dpInstall

Lets you install an interim fix when there are prerequisite check errors.

The default behavior of the update installer application prevents further action if prerequisites are missing.

-dpUninstall

Lets you uninstall an interim fix when there are prerequisite check errors.

-fixOnly

Allows interim fix installation and uninstallation only.

To install and uninstall interim fixes, the updateWizard does not require a local copy of the IBM Developer Kit or the Java Runtime Environment (JRE) in a client environment. This option bypasses making a local copy of the IBM Developer Kit or the JRE.

The default action for the update installer enables both interim fix and fix pack installation and uninstallation. The wizard installs a temporary version of one of the IBM products that WebSphere Application Server uses to support the Java 2 SDK on your operating system platform, such as the IBM Developer Kit for AIX, Java Technology Edition.

The **updateWizard** command copies the IBM SDK from the JAVA_HOME directory to the directory where you are running the **updateWizard** (usually *install_root*/update). If the IBM SDK is already in the directory (for example, if you used the **updateWizard** command before), it is not necessary for the **updateWizard** to make a new copy.

The copy of the IBM SDK remains in the directory until you remove it. The IBM SDK requires approximately 43 MB of free space.

If you install on a client platform, where there is a Java runtime environment instead of the IBM SDK, the **updateWizard** copies the JRE to the *install_root*/update directory. The JRE requires about 18 MB of free space.

-usage

Displays a list of parameters and how to use them in the correct command syntax.

Examples of use

Displaying usage information: This command displays information about command syntax.

```
updateWizard -usage
```

Bypassing the local copy of the IBM Developer Kit: The following command bypasses making a local copy of the IBM Developer Kit. Installing and removing fixes does not require the local copy.

```
updateWizard -fixOnly
```

Disabling prerequisite checking: Disabling prerequisite checking is recommended only as directed by IBM Support personnel. Disabling prerequisite checking can leave the installation in a non-valid state, unless done with caution and guidance.

To disable prerequisite checking when installing interim fixes:

```
updateWizard -fixOnly -dpInstall
```

To disable prerequisite checking when removing interim fixes:

```
updateWizard -fixOnly -dpInstall
```

To disable prerequisite locking when installing and when removing interim fixes:

```
updateWizard -fixOnly -dpInstall -dpUninstall
```

Panel descriptions

Panels in the wizard let you select installable fixes and fix packs, view installed fixes and fix packs, and view prerequisite fixes:

- **General**
  - Welcome panel
  - Product selection panel
  - Menu panel
- **Interim fix installation and uninstallation**
  - Fix repository specifier panel
  - Installable fix selection panel
  - Uninstallable fix selection panel
  - Prerequisite check panel
  - Pre-installation and pre-uninstallation summary panels
  - Installation and uninstallation
  - Post installation and post uninstallation summary panels
- **Fix pack installation and uninstallation**
  - Fix pack repository specifier panel
  - Fix pack selection panel
  - Fix pack features selection panel
  - Pre-installation and pre-uninstallation summary panels
  - Installation and uninstallation
  - Post installation and post uninstallation summary panel

Welcome panel: Use this panel to view a welcome message that contains a brief summary of the update wizard interface, or to link to the Support Web site. (This link is not available on some UNIX-based platforms.) You can also view relevant legal notices.

Product Selection panel: Use this panel to select an installed WebSphere Application Server product. If the wizard cannot detect an installed product, specify the product location in the **Installation directory** field. To make corrections or enter another product location, click **Specify product location**.

On some platforms, there is a limitation in the InstallShield for MultiPlatforms (ISMP) program that the update installer program uses. The ISMP program does not recognize previous installations of WebSphere Application Server on some operating platforms. To work around the problem, click **Specify product location** and type the fully qualified installation root directory for the existing product in the **Installation directory** field.

Menu panel: Use this panel to install or uninstall interim fixes, or to install or uninstall fix packs. If you started the wizard in -fixOnly mode, fix pack options are disabled.

Interim Fix Repository Specifier panel: Use this panel to provide the interim fix repository location in a directory input field. Specify the directory location of the downloaded fix JAR files. The default location for the repository is the `install_root`/update/fixes directory.

Installable Interim Fix Selection panel: Use this panel to select one or more installable fixes for installing. Installed fixes do not appear in the list. Only uninstalled fixes or partially-installed fixes appear. The list includes fix ID name, build date, and the current applied state (uninstalled or partially-installed). Click **Details** for detailed information about selected fixes. The window that appears contains build version information, a long description, and a list of installation prerequisites.

About installation status

An interim fix or fix pack is a collection of updates to one or more product components. Depending on installed product components and on update installer selections you make, the update installer applies either a full or partial set of interim fix or fix pack updates to product components.

Installed status implies that the interim fix or fix pack has no more updates to product components that you can install.

Partially installed status implies that you have updated one or more product components, but the interim fix or fix pack has at least one more update to apply to an installed product component. (There might be other updates to product components you never installed. These updates do not count in the status determination.)

Uninstalled status implies that you have not updated a single product component.

Examples of partially installed states: Several scenarios can lead to a partial installation of an interim fix or fix pack:

- Installation fails, leaving some component updates applied and some unapplied. This is a partial installation accompanied by error messages that describe the problem.
- Skipping certain optional component updates, such as might be present in a fix pack for these features: IBM HTTP Server or embedded messaging. This is a partial installation.
- Dynamically changing interim fix or fix pack status because:
  1. You apply all changes in an interim fix or fix pack that results in an installed status.
  2. You reinstall the WebSphere Application Server product to select an additional, optional feature.
  3. The interim fix or fix pack contains an update to a product component you just installed.

  The status changes dynamically from installed to partially installed.

Uninstallable Fix Selection panel: Use this panel to select one or more installed interim fixes for removing. Available interim fixes do not appear in the list. Only installed interim fixes appear. The list includes fix ID name, build date, and the current applied state (installed). Click **Details** to obtain more detailed information about a selected interim fix. The window that appears contains build version information, a long description, and a list of installation prerequisites.

You must uninstall all interim fixes before uninstalling the base WebSphere Application Server product, the Network Deployment product, or the Enterprise product.

Prerequisite Check panel: Use this panel to view prerequisite information when a selected fix has prerequisite fixes that are not installed. You cannot click **Next** until you correct the problem. The `-dpInstall` and the `-dpUninstall` parameters can override this lock, to let you continue with the installation or uninstallation despite prerequisite failure.

Pre-installation Summary and Pre-uninstallation Summary panels: **Pre-installation Summary panel** Use this panel to display a summary of interim fixes that are selected for installation, the WebSphere Application Server product each interim fix is for, and the directory where each interim fix is located.

Pre-uninstallation Summary panel Use this panel to display a summary of interim fixes that are selected for uninstallation, and the WebSphere Application Server product to which each interim fix is currently applied.

Installation and Uninstallation panel: **Installation action** Use this panel to view the progress of installing selected fixes. Click **cancel** to revert the installation. Once cancelled, a message confirms that installed fixes are being rolled back. A similar progress panel then appears, to monitor the progress of rolling back the installation of selected fixes.

Uninstallation action Use this panel to view the progress of removing selected fixes.

Post Installation Summary and Post Uninstallation Summary panels: **Post Installation Summary panel** Use this panel to view the results of the installation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the installation process is complete. Click **Finish** to exit the panel. You can also go back and install additional interim fixes, which takes you to the **Menu** panel.

Post Uninstallation Summary panel Use this panel to view the results of the uninstallation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the uninstallation process is complete. Click **Finish** to exit the panel. You can go back and uninstall additional interim fixes, which takes you to the **Menu** panel.

Fix Pack Repository Specifier panel: Use this panel to provide the fix pack repository location in a directory input field. The location should point to the directory where you unpacked downloaded fix pack JAR files. The default location for the repository is the `install_root/update/fixpacks` directory.

Fix Pack Selection panel: Use this panel to select from a list of installable fix packs. The panel displays fix packs by ID name, with a radio button next to each for selecting a single fix pack. Also displayed is the build date and current applied state (uninstalled or partially-installed) for each fix pack.

A fix pack on this panel can be in a partially installed state. No installed fix packs appear in the list. Click **Details** for more information about a selected fix pack. The window that appears contains build version information, a long description, installation prerequisites, and a list of included fixes.

See Installable Interim Fix Selection panel for a description of installation states.

Fix Pack Features Selection panel: Use this panel to view a list of WebSphere Application Server features with optionally installable service in the selected fix pack. Features that can appear include IBM HTTP Server and the embedded messaging feature, which is based on IBM WebSphere MQ technology.

If you do not install optional service for an installed feature, the fix pack installs successfully as a ***partially installed fix pack*** because there is service that you did not install.

If you installed the IBM HTTP Server product as a feature, use the update installer to update it with service in an interim fix or fix pack. Otherwise, download an updated IBM HTTP Server product and install it into the same directory as your existing version, to update the existing installation. You can also uninstall the current version and install the downloaded version to avoid any issues with migration.

Update your configuration if you reinstall. The process is described in the *Manually configuring supported Web servers* (tins_manualWebserver) topic in the base Application Server InfoCenter.

Always apply any outstanding corrective service to the stand-alone IBM WebSphere MQ product if you have it, before using the WebSphere Application Server update installer to update the embedded messaging feature with service in an interim fix or fix pack.

Do not install service to the embedded messaging feature if you install corrective service to the stand-alone IBM WebSphere MQ product.

Pre-installation Summary and Pre-uninstallation Summary panels: **Pre-installation Summary panel** Use this panel to display a summary of the fix pack selected for installation, the WebSphere Application Server product the fix pack is for, and the directory where the fix pack is located.

Pre-uninstallation Summary panel Use this panel to display a summary of the fix pack selected for uninstallation, the WebSphere Application Server product to which the fix pack was applied, and the directory where the fix pack is located.

Installation and Uninstallation panels: **Installation action** Use this panel to view the progress of installing the selected fix pack. Click **cancel** to revert the installation. Once cancelled, a message confirms that the fix pack is being rolled back. A similar progress panel then appears, to monitor the progress of rolling back the installation of the selected fix pack.

Uninstallation action Use this panel to view the progress of removing the selected fix pack. There is no way to cancel the uninstallation.

You must uninstall all fix packs before removing the base WebSphere Application Server product, the Network Deployment product, or the Enterprise product.

Post Installation Summary and Post Uninstallation Summary panel: **Post Installation Summary panel** Use this panel to view the results of the installation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the installation process is complete. Click **Finish** to exit the panel.

You can go back and install another fix pack, which takes you to the **Menu** panel.

Post Uninstallation Summary panel Use this panel to view the results of the uninstallation. Depending on the result, this panel can display a success message, a failure message, or a canceled message. When the success message appears, the uninstallation process is complete. Click **Finish** to exit the panel.

You can go back and uninstall another fix pack, which takes you to the **Menu** panel.

Removing interim fixes and fix packs

This topic describes the proper procedure for removing an interim fix or a fix pack in an IBM WebSphere Application Server, V5 environment using the update installer application.

Removing an interim fix or fix pack requires setting the JAVA_HOME environment variable for the update installer. The update installer performs the task by running the setupCmdLine or setupClient command script. If you use a non-standard installation root directory for your WebSphere Application Server product, it is possible that the update installer cannot set the JAVA_HOME environment variable.

If the update installer throws an error because it cannot set the Java environment, set the JAVA_HOME variable yourself. Then you can use the update installer to uninstall the interim fix or fix pack, using either its wizard interface, the **updateWizard** command, or its silent, command-line interface, the **updateSilent** command.

The update installer application can also install fixes and fix packs.

If the base WebSphere Application Server node is within a cell, go to Removing interim fixes and fix packs. If the base Application Server node is extended by the Enterprise product, go to Removing interim fixes and fix packs. This topic describes removing an interim fix or fix pack from a stand-alone base node.

Use the **versionInfo** command in the `install_root/bin` directory to display the exact interim fix and version level of the product. Do not use the **versionInfo** command while installing or removing an interim fix or fix pack.

You can also use the silent update installer application to:
* View interim fix information
* View fix pack information

Before installing or removing interim fixes and fix packs, stop all Java processes that use the IBM SDK that WebSphere Application Server provides to support the Java 2 SDK on your operating system

platform, such as the IBM Developer Kit for AIX, Java Technology Edition. Stop all Application Servers, and all servers, such as the IBMHttpServer process, that belong to serviceable features. Features with servers include the IBM HTTP Server and the embedded messaging feature. Stop all Java processes, if necessary. If you do install or uninstall an interim fix or fix pack while a WebSphere Application Server-related Java process is running, IBM does not guarantee that the product can continue to run successfully, or without error.

This procedure describes a scenario for updating a stand-alone base node by removing an interim fix or fix pack. The Installing fixes and fix packs topic describes how to apply an interim fix or fix pack to a stand-alone base node.

To remove the highest level interim fix or fix pack from a base node, perform the following procedure.

Steps for this task

1. Stop each server on the base node with the **stopServer** command.

   Stop all WebSphere Application Server-related Java processes. On a Windows platform, you can use the task manager to stop Java processes. On a UNIX-based platform, use the **kill** command to stop Java processes.

2. Set up and configure your WebSphere Application Server environment.

   The locations of the `update`, `fixes` repository, and `fixpacks` repository directories is arbitrary. Create the directories anywhere. However, the `install_root/update`, `install_root/update/fixes`, and `install_root/update/fixpacks` locations are recommended.

   a. If you do not have the update installer application, you can download it from the Support page at http://www-3.ibm.com/software/webservers/appserv/was/support/.

   b. Create an `install_root/update` directory, if it does not already exist.

   c. Extract the contents of the update installer zip file to the update directory. On Windows platforms, the pkunzip utility might not decompress the download image correctly. Use another utility (such as WinZip) to unzip the image.

   d. **(Optional)** Set the Java environment for the update installer.

      The locations of the `update`, `fixes` repository, and `fixpacks` repository directories is arbitrary. Create the directories anywhere. However, the `install_root/update`, `install_root/update/fixes`, and `install_root/update/fixpacks` locations are recommended.

      If you use a non-standard installation root, it is possible that the **updateWizard** (or **updateSilent**) command cannot set the JAVA_HOME environment variable. If you receive a message that the update installer cannot set JAVA_HOME, set the environment variable yourself, or issue the appropriate command script yourself, from the `bin` directory of the installation root:

      1) Open a command line window.

      2) Change directories to the `bin` directory of the installation root.

      3) Run the appropriate command:
         - `. install_root/bin/setupCmdLine.sh` (source the command on UNIX platforms)
         - `source install_root/bin/setupCmdLine.sh` (source the command on Linux platforms)
         - `install_root\bin\setupCmdLine.bat` (Windows platforms only)
         - `. install_root/bin/setupClient.sh` (source the command for the Application Server client)
         - `source install_root/bin/setupClient.sh` (Linux platforms only)
         - `install_root\bin\setupClient.bat` (Windows platforms only)

3. Use the appropriate command to remove the interim fix or fix pack from the base node:
   - Refer to updateWizard command topic for usage information.
   - Refer to the **updateSilent** command description for the proper syntax for removing the interim fix or fix pack:
     - Removing interim fixes

– Removing fix packs

For example, to uninstall the `was50_fp2_win` fix pack, use this **updateSilent** command:

```
C:\Program Files\WebSphere\AppServer\update> updateSilent -fixpack
      -installDir "C:\Program Files\WebSphere\AppServer"
      -skipIHS
      -fixpackDir "C:\Program Files\WebSphere\AppServer\update\fixpacks"
      -uninstall
      -fixpackID was50_fp2_win
```

The command is shown on more than one line, for clarity.

4. Uninstall the base WebSphere Application Server product manually if you cannot successfully uninstall the interim fix or fix pack.

The following platform-specific uninstallation procedures each describe a manual uninstallation process that guides you through removing every trace of the products and features you might have installed, including directories that might have changed data. Before you begin one of the procedures, back up the config and properties install-root directories. Backing up the directories lets you refer to the changed data when you reinstall.

- Manually uninstalling on AIX platforms
- Manually uninstalling on HP-UX platforms
- Manually uninstalling on Linux platforms
- Manually uninstalling on Solaris platforms
- Manually uninstalling on Windows platforms

5. Restart each server on the node with the **startServer** command.

6. Verify that the node is online and functioning correctly.

There are several ways to verify the successful removal of an interim fix or fix pack:

- Does the interim fix appear in the wizard panel that lists applied interim fixes, or does the or fix pack appear in the panel that lists applied fix packs? If so, the interim fix or fix pack is installed.
- Does the interim fix or fix pack appear in the wizard panel that lists installable interim fixes, or the panel that lists installable fix packs? If so, the interim fix or fix pack is uninstalled.
- Is there an `[interim fixID].efix` or `[fix packID].ptf` file in the `install_root/properties/version/version` directory, or an `[interim fixID].efixApplied`, `[interim fixID].efixDriver`, `[fix packID].ptfApplied`, or `[fix packID].ptfDriver` file in the `install_root/properties/version/history` directory?
- Do the product version and history reports show the fix or fix pack to be installed or removed?
- Does collecting interim fix information and update state show that the interim fix is installed or removed?
- Does collecting fix pack information and update state show that the fix pack is installed or removed?

Run the installation verification tool on the node to verify that the node is operational.

Product version and history information

The WebSphere Application Server product contains structural differences from previous versions. The `/properties/version` directory in the installation root contains important data about the product and its installed components, such as the build version and build date. This information is included in [product].product and [component].component files.

The `/properties/version/history` directory in the installation root contains a collection of records for installed fixes and fix packs. This information is included in [interim fixID].efixApplied, [interim fixID].efixDriver, [fix packID].ptfApplied, and [fix packID].ptfDriver files.

A driver file has useful information about the entire contents of an interim fix or fix pack. The applied file has relevant information about the fixes or fix packs that are currently applied.

Event.history files contain a detailed log about updates you have applied, either successfully or unsuccessfully. Time-stamped, detailed logs record each update process in the `/properties/version/log` directory of the installation root. Beginning with Fix Pack 2, the time-stamped, detailed logs are in the `install_root/logs/update` directory.

This topic describes the XML data files that store product information for V5 WebSphere Application Server products. By default, the document type declarations (DTDs) for these files are in the `properties/version/dtd` folder of the installation root, or the server root directory. See the Storage locations section for more information.

This topic includes:
- A list of product information files and file locations
- Report scripts for displaying version and history information
- A description of logs and component backups
- A list of storage locations
- A description of how the update service makes operational use of the product information
- A data dictionary that describes data in the files

Product information files

There are two kinds of product information files:
- XML files in the /properties/version directory that store version information
- XML files in the /properties/version/history directory

XML files in the /properties/version directory that store version information
The following file indicates that a WebSphere Application Server V5 product is installed:

platform.websphere
> One file whose existence indicates that a WebSphere Application Server product is installed. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE websphere PUBLIC "websphereId" "websphere.dtd">
<websphere name="IBM WebSphere Application Server" version="5.0"/>
```

The following XML files in the `/properties/version` directory represent installed items and installation events:

***<product-id>*.product**
> One file whose existence indicates the particular WebSphere Application Server product that is installed. Data in the file indicates the version, build date, and build level. For example, the file might be the `ND.product` file, which indicates that the installed product is WebSphere Application Server Network Deployment. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE product PUBLIC "productId" "product.dtd">
<product name="IBM WebSphere Application Server for Network Deployment">
  <id>ND</id>
  <version>5.0.0</version>
  <build-info date="10/5/02" level="s0239.28"/>
</product>
```

***<component-name>*.component**
> Any number of component files that each indicate the presence of an installed component, which is part of the product. Data in the file indicates the component build date, build version, component name, and product version. For example, the file might be the `activity.component` file, which

indicates that the *activity* component is installed. The activity component is part of the Network Deployment product. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE component PUBLIC "componentId" "component.dtd">
<component build-date="10/5/02" build-version="s0239.28" name="activity" spec-version="5.0"/>
```

***<extension.id>*.extension**

Any number of extension files that each indicate the presence of an extension that you install as a user extension, as part of a service engagement, or as installed by a third party product. The *<extension.id>*.extension files are not created, logged, or removed by WebSphere Application Server products.

***<fix-id>*.efix**

Any number of fix files that each indicate the presence of an installed interim fix.

***<ptf-id>*.ptf**

Any number of files, that each indicate the presence of an installed fix pack.

XML files in the /properties/version/history directory

This file stores version history information:

event.history

One file that lists update events that have occurred. An update event is an operation that installs or uninstalls an interim fix or a fix pack. The file is sorted by the date and time of the events that are listed.

The following XML files in the `/properties/version/history` directory describe fixes and fix packs that are currently installed. These XML files are related to installation items by the primary ID information, which is shown here by *<angle brackets>* and italicized text.

***<fix-id>*.efixDriver**

Fix-driver defining information

***<fix-id>*.efixApplied**

Fix installation details

***<ptf-id>*.ptfDriver**

fix pack-driver defining information

***<ptf-id>*.ptfApplied**

fix pack installation details

Reports

You can view product information by examining files in the `install_root/properties/version` directory, including the `install_root/properties/version/history` directory.

WebSphere Application Server provides the ability to generate two types of reports about the data in the files, *Version* reports and *History* reports. The following report-generation scripts are available in the installation root `bin` directory.

Product version reports

The following report generation scripts extract data from XML data files in the `properties/version` folder.

versionInfo.bat

Lets you use parameters to create a version report on Windows platforms.

Do not use the *versionInfo* command while installing or removing the product, or while installing or removing an interim fix or fix pack.

Parameters:

-format

> TEXT | HTML

> Selects the format of the report. The default is TEXT.

-file *<fileName>*

> Specifies the output file name. The report goes to standard output (stdout) by default.

-components
> Adds a list of installed components to the report.

-componentDetail
> Adds details about installed components to the report.

-efixes
> Adds a list of applied interim fixes to the report.

-efixDetail
> Adds details about applied interim fixes to the report.

-ptfs Adds a list of applied fix packs to the report.

-ptfDetail
> Adds details about applied fix packs to the report.

versionInfo.sh
> Lets you use parameters to create a version report on UNIX-based platforms. Parameters are the same as for the Windows version.

> Do not use the ***versionInfo*** command while installing or removing the product, or while installing or removing an interim fix or fix pack.

genVersionReport.bat
> Generates the `versionReport.html` report file in the `bin` directory on Windows platforms. The report includes the list of components, interim fixes, and fix packs.

genVersionReport.sh
> Generates the `versionReport.html` report file in the `bin` directory on UNIX-based platforms. The report includes the list of components, interim fixes, and fix packs.

Product history reports
The following report generation scripts extract data from XML data files in the
`install_root/properties/version/history` folder:

historyInfo.bat
> Lets you use parameters to create a history report of installed and uninstalled interim fixes and fix packs, on Windows platforms. Specify a component name to create a report that shows the history for that component.

Parameters:

-format

> TEXT | HTML

> Selects the format of the report. The default is TEXT.

-file *<fileName>*

> Specifies the output file name. The report goes to standard output (stdout) by default.

-updateID <ID>
> Specifies the ID of an fix or fix pack update. When specified, the product history report displays events for only the specified update. When not specified, the report displays events for all updates.

-component <componentName>
Specifies the name of a component. When specified, the product history report displays events for only the named component. When not specified, the report displays events for all components.

historyInfo.sh
Lets you use parameters to create a history report on UNIX-based platforms. Parameters are the same as for the Windows version.

genHistoryReport.bat
Generates the `historyReport.html` report file in the `install_root\bin` directory on Windows platforms. The report includes all updates and components.

genHistoryReport.sh
Generates the `historyReport.html` report file in the `bin` directory on UNIX-based platforms. The report includes all updates and components.

Logs and component backups

WebSphere Application Server products use two other directories when performing update operations, for logging and backups. By default, the two directories are relative to the product version directory, as follows:

***install_root*/properties/version/log**
Product updates log directory from V5.0.0 and V5.0.1.

WebSphere Application Server products store log files to document component, interim fix and fix pack operations and updates.

***install_root*/logs/update**
Product updates log directory beginning with Fix Pack 2, which is Version 5.0.2.

Beginning with Fix Pack 2, the update installer stores its detailed log files in the `install_root/logs/update` directory.

***install_root*/properties/version/backup**
Product updates `backup` directory

WebSphere Application Server products back up components before applying fixes and fix packs. If you uninstall an interim fix or fix pack, WebSphere Application Server products restore the backed-up component JAR file.

File naming convention

Time stamp
YYYYMMDD_HHMMSS

For example: `20020924_211832` is 24-Sep-2002, 9:18:32 pm, GMT. All time stamps are in GMT.

ID Fix ID or fix pack ID

For example: apar6789c is an fix ID; PTF_1 is a fix pack ID.

Operation
install | uninstall

Interim fix log file names
<timeStamp>_<fixId>_<operation>.log

For example: `properties/version/log/20020924_211832_apar6789c_install.log` and `properties/version/log/20020924_211912_apar6789c_uninstall.log`

At Fix Pack 2 (V5.0.2) or later, the update installer creates these logs: `install_root/logs/update/20020924_211832_apar6789c_install.log` and `install_root/logs/update/20020924_211912_apar6789c_uninstall.log`

Interim fix component log file names

      `<timeStamp>_<fixId>_<componentName>_<operation>.log`

      For example: `properties/version/log/20020924_211832_apar6789c_ras_install.log` and `properties/version/log/20020924_211912_apar6789c_ras_uninstall.log`

      At Fix Pack 2 or later, the update installer creates these logs: `install_root/logs/update/20020924_211832_apar6789c_ras_install.log` and `install_root/logs/update/20020924_211912_apar6789c_ras_uninstall.log`

Fix pack log file names

      `<timeStamp>_<ptfId>_<operation>.log`

      For example: `properties/version/log/20020924_211832_PTF_1_install.log` and `properties/version/log/20020924_211912_PTF_2_uninstall.log`

      At Fix Pack 2 or later, the update installer creates these logs: `install_root/logs/update/20030324_211832_was50_fp2_install.log` and `install_root/logs/update/20030325_211912_was50_fp2_uninstall.log`

Fix pack component log file names

      `<timeStamp>_<ptfId>_<componentName>_<operation>.log`

      For example: `properties/version/log/20020924_211832_PTF_1_ras_install.log` and `properties/version/log/20020924_211912_PTF_2_ras_uninstall.log`

      At Fix Pack 2 or later, the update installer creates these logs: `install_root/logs/update/20030324_211832_was50_fp2_ras_install.log` and `install_root/logs/update/20030325_211912_was50_fp2_ras_uninstall.log`

Backup JAR file names

      `<timeStamp>_<ptfId>_<componentName>_undo.jar` or `<timeStamp>_<fixId>_<componentName>_undo.jar`

      For example: `20020924_211832_apar6789c_ras_undo.jar`

      Do not delete a backup Java archive (JAR) file. You cannot remove a component update if the corresponding backup JAR file is not present.

Update processing might also use a temporary directory, if necessary. A Java property specifies this directory as described in the next section.

Storage locations

Product information files are located relative to the WebSphere Application Server product installation root, or the server root directory.

Default file paths and Java properties that set them are:

Version directory

      `install_root/properties/version` or `server_root/properties/version`

History directory

      `install_root/properties/version/history`

Updates log directory

      `install_root/properties/version/log`

      The version of the update installer that is bundled with Fix Pack 2 and later fix packs, stores log files in the `install_root/logs/update` directory.

Updates backup directory

      `install_root/properties/version/backup`

DTD directory
> `install_root/properties/version/dtd`

Temporary directory
> Specified by the java.io.tmpdir Java system property

Operational description

WebSphere Application Server products update the product version history information while performing events that install or uninstall fixes or fix packs. Events that might occur include:

- A WebSphere Application Server product adds an interim fix file (with an extension of `.efix`) to the `version` directory to indicate that an interim fix is currently installed.
- A WebSphere Application Server product removes an interim fix file from the `version` directory when it uninstalls the corresponding interim fix.
- A WebSphere Application Server product adds an interim fix driver file (with an extension of `.efixDriver`) to the `history` directory when an interim fix is installed. An interim fix driver file contains defining information for an interim fix.
- A WebSphere Application Server product removes an interim fix driver file when it removes the corresponding interim fix.
- A WebSphere Application Server product adds an interim fix application file (with an extension of `.efixApplied`) to the `history` directory when it installs an interim fix. An interim fix application file contains information that identifies component updates that have been applied for an interim fix. The application file also provides links to component log and backup files.
- A WebSphere Application Server product removes an interim fix application file when it removes the corresponding interim fix.
- A WebSphere Application Server product adds a fix pack, with an extension of `.ptf`, to the `version` directory to indicate than a fix pack is currently installed.
- A WebSphere Application Server product removes a fix pack file from the version directory when it uninstalls the corresponding fix pack.
- A WebSphere Application Server product adds a fix pack driver file (with an extension of `.ptfDriver`) to the `history` directory when it installs a fix pack. A fix pack driver file contains defining information for a fix pack.
- A WebSphere Application Server product adds a fix pack application file (with an extension of `.ptfApplied`) to the `history` directory when it installs a fix pack. A fix pack application file contains information that identifies component updates that have been applied for a fix pack. The application file also provides links to component log and backup files.
- A WebSphere Application Server product makes entries in the history file, `event.history`, when it installs or uninstalls an update.
- A WebSphere Application Server product stores a parent event for each fix that it installs or uninstalls.
- A WebSphere Application Server product stores a parent event for each fix pack that it installs or uninstalls.
- A WebSphere Application Server product stores child component events for each component update that it installs or uninstalls, beneath the corresponding fix or fix pack parent event.
- A WebSphere Application Server product stores one log file in the `log` directory as it installs or uninstalls one fix or fix pack.
- A WebSphere Application Server product stores one log file in the `log` directory as it installs or uninstalls an interim fix or fix pack, in response to each component update that occurs.
- A WebSphere Application Server product stores a component backup file in the backup directory for each component update that it installs.
- A WebSphere Application Server product removes a component backup file from the backup directory for each component update that it uninstalls.

Data dictionary

Type Family: **websphere product family**

File Types: websphere

File Type: websphere

Elements: name   string   required
                    version   string   required

Persistence: <versionDir>/platform.websphere

Type Detail:

The websphere file is placed to denote the presence of websphere family
products.

Element Detail:

websphere.name   The WebSphere product family name.
websphere.version   The WebSphere product family version.

Type Family: product

File Types: product
                  component
                  extension

File Type: product

Persistence: <versionDir>/<id>.product

Elements: id   string   required
                  name   string   required
                  version   string   required
                  build-info   complex   required

Type Detail:

A product file is placed to denote the presence of a specific WebSphere family product.
The product's id is embedded in the product file name.

Element Detail:

product.id   The id of the product.
product.name   The name of the product.
product.version   The version of the product.
product.build-info   An element containing build information for
                              the product.

Element Type: build-info

Elements: date   date   required
                  level   string   required

Type Detail:

A build-info instance details the build of a specific installed websphere
family product.

Element Detail:

build-info.date   The date on which the product was build.
build-info.level   The level code of the product's build.

```
File Type:       component

Persistence:     <versionDir>/<name>.component

Elements:        name                string     required
                 spec-version        string     required
                 build-version       string     required
                 build-date          date       required

File Detail:

A component file denotes the presence of a specific component.  The
component name is embedded in the component file name.

Element Detail:

component.name          The name of the component.
component.spec-version  The specification version of the component.
component.build-version The build level of the component.
component.build-date    The build date of the component.


File Type:       extension

Persistence:     <versionDir>/<id>.extension

Elements:        id                  string     required
                 name                string     required

File Detail:

An extension file denotes the presence of a specific extension.  The
extension's id is embedded in the extension file name.

The elements of an extension file are minimally specified.  The listed
elements are required.  Additional elements may be present as determined
by the actual installed extension.

Element Detail:

extension.id            The id of the extension.
extension.name          The name of the extension.


Type Family:     update

File Types:      efix
                 ptf
                 efix-applied
                 ptf-applied

File Type:       efix

Persistence:     <versionDir>/<id>.efix

Elements:        id                  string     required
                 apar-number         string     optional
                 pmr-number          string     optional
                 short-description   string     required
                 long-description    string     required
                 is-temporary        boolean    required
                 build-version       string     required
                 build-date          date       required
                 component-update    complex    min=1, max=unbounded
                 platform-prereq     complex    min=0, max=unbounded
                 product-prereq      complex    min=0, max=unbounded
                 efix-prereq         complex    min=0, max=unbounded
                 custom-property     complex    min=0, max=unbounded
```

Type Detail:

An efix file denotes the presence of some portion of a specific interim fix. The id of the fix is embedded in the file name.

An efix file contains all fix data, such as description, a listing of component updates, and prerequisite information.

Almost always, when installing an interim fix, all of the potential component updates within the fix are required to be installed.

A separate application file must be examined to determine the components which have been updated for a particular interim fix.

A list of custom properties may be provided. These are provided for future use.

Element Detail:

efix.id   The id of the interim fix.

efix.short-description A short description of the interim fix.

efix.long-description   A long description of the interim fix.

efix.is-trial   A flag indicating whether or not an interim fix is considered
                        a trial interim fix. Generally, a trial fix will be
                        followed up with a more permanent interim fix.
efix.expiration-date   A date on which the fix is to be considered obsolete.

efix.build-version   The build version of the interim fix. This is distinct from
                        the build version of component updates contained within
                        the interim fix.

efix-build-date   The build date of the interim fix. This is distinct from the
                        build version of the component updates contained within
                        the interim fix.

efix.apar-info   A list of APAR's which are associated with the interim fix.

efix.component-update   A list of updates for components. For an interim fix, these are
                        usually all required, and are all patch updates. At least
                        one component update must be present.

efix.efix-prereq   A list of prerequisite fixes for the interim fix. Note that
                        prerequisite fixes may be negative (see below). This
                        list may be (and is often) empty.

efix.plaform-prereq   A list of platforms on which the fix may be installed.
                        This list may be empty, in which case the fix may be
                        installed on all platforms.

efix.product-prereq   A list of products on which the fix may be installed.
                        This list may be empty, in which case the fix may be
                        installed on all products.

efix.custom-proprty   A list of properties, provided for future use.

File Type: ptf

Persistence: <versionDir>/<id>.ptf

Elements: id   string   required
                 short-description   string   required
                 long-description   string   required
                 build-version   string   required

```
                          build-date              date         required
                          component-update        complex      min=1, max=unbounded
                          product-update          complex      min=0, max=unbounded
                          platform-prereq         complex      min=0, max=unbounded
                          product-prereq          complex      min=0, max=unbounded
                          included-efix           complex      min=0, max=unbounded
                          custom-property         complex      min=0, max=unbounded
```

Type Detail:

A ptf file denotes the presence of some portion of a specific fix pack. The id of
the fix pack is embedded in the fix pack file name.

A ptf file contains all fix pack data, such as description, a listing of component
updates, and prerequisite information.

Usually, when installing a fix pack, omit certain potential component updates, but
only when the corresponding component is not installed.

Examine a separate application file to determine which components a
particular fix pack has updated.

A fix pack can include updates for a number of fixes.

A list of custom properties might be provided. These are provided for future use.

Element Detail:

ptf.id   The ID of the fix pack.

ptf.short-description A short description of the fix pack.

ptf.long-description   A long description of the fix pack.

ptf.build-version   The build version of the fix pack. This is distinct from the
                       build version of component updates contained within the fix pack.

ptf-build-date   The build date of the fix pack. This is distinct from the
                       build version of the component updates contained within the
                       fix pack.

ptf.component-update   A list of updates for components. For a fix pack, these are
                       usually all required, and are all patch updates. At least
                       one component update must be present.

ptf.plaform-prereq   A list of platforms on which you can install the fix pack. This
                       list might be empty. If so, you can install the fix pack on all platforms.

ptf.product-prereq   A list of products on which you can install the fix pack. This
                       list might be empty. If so, you can install the fix pack on all products.

ptf.included-efix   A list of fixes which are included (fixed) by the fix pack.

ptf.custom-proprty   A list of properties, provided for future use.

Element Type: component-update

```
Elements:         component-name          string       required
                  update-type             enum         required [enumUpdateType]
                  is-required             boolean      required
                  is-recomended           boolean      required
                  is-optional             boolean      required
                  is-external             boolean      required
                  root-property-file      anyURL       optional
                  root-property-name      string       optional
                  root-property-value     anyURL       optional
                  is-custom               boolean      required
```

```
                    primary-content      string    required
                    component-prereq     complex   min=0, max=unbounded
                    final-version        complex   optional
                    custom-property      complex   min=0, max=unbounded
```

Type Detail:

A component update represents a potential component update which is packaged
in an update (an interim fix or a fix pack).

An component update may be required, in which case the parent update may not
be installed unless the component update can be installed. (A component update
can be installed if the corresponding component is installed.)

A component update may be a custom update, in which case the content which
was provided must be an executable file. Otherwise, the content which is
provided must be an update jar file.

A component update has a type. A final version may be required according to
the update type.

Element Detail:

component-update.component-name   The name the component which is to be updated.

component-update.update-type   The type of the component update, one of 'add',
                                  'replace', 'remove', or 'patch'. Final version
                                  information must be provided when the update type
                                  is 'add' or 'replace'.

component-update.is-required   A flag which, when true, specifies that the parent
                                  update may not be applied unless this component
                                  update is applied.

component-update.is-recomended   A flag which, when true, specifies that this
                                  component update, although optional, should be
                                  installed.

component-update.is-optional   A flag which, when true, specifies that this update
                                  may be omitted even if its corresponding component
                                  is installed.

component-update.is-external   A flag which, when true, specifies that this
                                  component update may live outside of the usual
                                  install root.

component-update.root-property-file   For a component with an external root, this
                                      properties file provides the root value.

component-update.root-property-name   For a component with an external root, this named
                                      property provides the root value.

component-update.root-property-value For a component with an external root, this value
                                      provides the default root value.

component-update.is-custom   A flag which, when true, specifies that the update
                                  is a custom update. When true, the content must
                                  be an executable program. When false, the content
                                  must be an update jar.

component-update.primary-content The name of the content which is provided for the
                                  update. This will be an entry which is packaged
                                  in the 'components' directory of the update.

component-update.component-prereq A list of component versions, one of which must
                                  be present for this update to be installed.
                                  When this list is empty, any component version
```

```
                              is allowed.

component-update.final-version    Final version information for the component.
                                  A final version is required when the update
                                  operation is 'add' or 'replace'.

component-update.custom-property  A list of properties, provided for future use.

Element Type:      apar-info

Elements:          number              string     required
                   date                date       required
                   short-description   string     required
                   long-description    string     optional

Type Detail:

An apar-info object provides information about an APAR which is associated
with an interim fix, usually indicating that the fix provides an interim fix for the APAR.

Element Detail:

apar-info.number            The number of the associated APAR.
apar-info.date              The date of the APAR.
apar-info.short-description A short description of the APAR.
apar-info.long-description  An optional long description of the APAR.

Element Type:      efix-prereq

Elements:          efix-id             string     required
                   is-negative         boolean    required
                   install-index       int        optional

Type Detail:

An interim fix prerequisite instance denotes an interim fix that must be present (or, if negative,
must be absent) for the parent fix to be installed.

efix prerequisite instances may specify a cycle, in which case the prerequisite
specification is treated as a corequisite specification.

The following chart summarizes the interpretation of prerequisite information for
two fixes:

      fix1       fix2
       -          -          The fixes may be installed without regard to each other.
      fix2        -          fix1 must be installed after fix2 is installed.
       -         fix1        fix2 must be installed after fix1 is installed.
      fix2       fix1        fix1 and fix2 must be installed together.
     !fix2        -          fix1 may not be installed after fix2 is installed.
       -        !fix1        fix2 may not be installed after fix1 is installed.
     !fix2      !fix1        fix1 and fix2 may not ever be installed together.
     !fix2       fix1        This is an erroneous specification.
      fix2      !fix1        This is an erroneous specification.

The installation index element provides ordering information for corequisite fixes
that must be installed in a particular order.

Element Detail:

fix-prereq.efix-id          The id of the prerequisite interim fix.

fix-prereq.is-negative      A flag which indicates if the prerequisite
                            fix is required or prohibited.  If false,
                            install the interim fix before installing
                            the parent interim fix.  If true, do not install
                            the interim fix before you install the parent interim fix.
```

```
fix-prereq.install-index   An optional index number used to order
                           corequisite fixes.


Element Type:       product-update

Elements:           product-id              string    required
                    product-name            string    required
                    build-version           string    required
                    build-date              date      required
                    build-level             string    required

Type Detail:

A product update specifies a replacement to a product file.

The product update information matches the information in product files.

Multiple product updates may be present, in which case each matching
product is updated.

Element Detail:

product-update.product-id     The id of the product that is updated.
product-update.product-name   The name of the product.
product-update.build-version  The build version of the product.
product-update.build-date     The build date of the product.
product-update.build-level    The build level of the product.


Element Type:       component-prereq

Elements:           component-name          string    required
                    spec-version            string    required
                    build-version           string    required
                    build-date              date      required


Element Type:       platform-prereq

Elements:           architecture            string    required
                    os-platform             string    optional
                    os-version              string    optional

Type Detail:

A platform prerequisite instance denotes a platform which must be present
for an update to be installed.  The element values are according to the
values supplied for the matching java properties.

Note that when multiple platform prerequisites are specified, these
prerequisites have an OR relationship: At least one of the platform
prerequisites must be satisfied.

Element Detail:

platform-prereq.architecture  The name of an architecture which must be
                              present.
platform-prereq.os-platform   The name of an operating system which
                              must be present.  This element is optional.
                              When absent, the architecture is checked,
                              but the os-platform and os-version are not.

platform-prereq.os-version    The version of a the operating system which
                              must be present.  This element is optional.
                              When absent, the architecture and os-platform
                              are checked, but os-version is not.  (When
                              os-platform is absent, os-version should not
                              be set.)
```

```
Element Type:      product-prereq

Elements:          product-id           string     required
                   build-version        string     optional
                   build-date           date       optional
                   build-level          string     optional
```

Type Detail:

A product prerequisite specifies that a particular product must be present for
an update to be installed.

Note that when multiple product prerequisites are specified, these
prerequisites have an OR relationship: At least one of the product
prerequisites must be satisfied.

Note that all of the elements are required. When multiple products having
the same id are supported by an update, multiple product prerequisites must
be specified.

Element Detail:

```
product-prereq.product-id     The id of the product which must be present.
product-prereq.build-version  The version of the product which must be present.
product-prereq.build-date     The build date of the product which must be present.
product-prereq.build-level    The level date of the product which must be present.
```

```
Element Type:      component-prereq

Elements:          component-name       string     required
                   spec-version         string     required
                   build-version        string     required
                   build-date           date       required
```

Type Detail:

A version prerequisite specifies that a particular component version must be
present for an update to be installed.

Note that when multiple version prerequisites are specified, these
prerequisites have an OR relationship: At least one of the version
prerequisites must be satisfied.

Element Detail:

```
version-prereq.component-name  The name of the component which must be present.

version-prereq.spec-version    The specification version of the component which
                               must be present.

version-prereq.build-version   The version of the component which must be present.

version-prereq.build-date      The build date of the component which must be
                               present.
```

```
Element Type:   included-efix

Elements:       efix-id              string     required
```

Type Detail:

An included-efix identifies an interim fix by ID and indicates that the fix
is included in the fix pack.

Element Detail:

```
included-efix.efix-id     The ID of the fix that the fix pack includes.

Element Type:      custom-property

Elements:          property-name          string     required
                   property-type          string     optional
                   property-value         string     optional

Type Detail:

A custom property encodes a key-value pair, with an optional type
element.  Custom properties are provided for future use.

Element Detail:

custom-property.property-name   The name of the custom property.
custom-property.property-type   An optional type of the custom property.
                                The semantics of this type are defined
                                by user of the property value.
custom-property.property-value  The value of the custom property.

File Type:         efix-applied

Persistence:       <versionDir>/<id>.efixApplied

Elements:          efix-id                string     required
                   component-applied      complex    min=0, max=unbounded

Type Detail:

An efix-applied collection specifies what components have been updated for
the fix as specified by the efix id.

Element Detail:

efix-applied.efix-id            The id of the fix for which applieds are
                                recorded.
efix-applied.component-applied  The list of recorded applications.

File Type:         ptf-applied

Persistence:       <versionDir>/<id>.ptfApplied

Elements:          ptf-id                 string     required
                   component-applied      complex    min=0, max=unbounded

Type Detail:

A ptf-applied collection specified what components have been updated for
the fix pack as specified by the fix pack ID.

Element Detail:

ptf-applied.efix-id             The ID of the fix pack for which applieds are
                                recorded.

ptf-applied.component-applied  The list of recorded applications.

Element Type:      component-applied

Elements:          component-name         string     required
                   update-type            enum       required [enumUpdateType]
                   is-required            boolean    required
                   is-optional            boolean    required
                   is-external            boolean    required
                   root-property-file     anyURL     optional
```

```
                    root-property-name      string     optional
                    root-property-value     string     optional
                    is-custom               boolean    required
                    log-name                anyURL     required
                    backup-name             anyURL     required
                    time-stamp              date       required
                    initial-version         complex    optional
                    final-version           complex    optional
```

Type Detail:

An applied instance is present to indicate the application of an
update for a particular fix or fix pack to a particular component.
(The particular fix or fix pack is as specified by the applied's
parent.) An applied provides sufficient information
to undo itself.

The elements of an applied are copies of values from update
events.

Element Detail:

```
component-applied.component-name   The name of the component which was updated.
component-applied.update-type      The type of the component update.
component-applied.is-required      A flag which, when true, specifies that the parent
                                   update requires this component update.
component-applied.is-optional      A flag which, when true, specifies that the parent
                                   update does not require this component update,
                                   even if the component is installed.
component-applied.is-external       A flag which, when true, specifies that this
                                   component update was applied to a location
                                   different than the usual install_root.
component-applied.root-property-file   For an update against a component having an
                                   external root, this properties file provides
                                   the root value.
component-applied.root-property-name   For an update against a component having an
                                   external root, this named property provides
                                   the root value.
component-applied.root-property-value  For an update against a component having an
                                   external root, this is a record of the
                                   actual external root.
component-applied.is-custom        A flag which, when true, specifies that the
                                   application was a custom update.  When true, an
                                   executable program was applied.  When false, the
                                   contents of an update jar were applied.
component-applied.log-name         The name of the log file which was generated by
                                   this application.
component-applied.backup-name      The name of the backup file which was generated
                                   by this application.
component-applied.time-stamp       The time of this application (the ending time
                                   of the corresponding update event).
component-applied.initial-version  The version of the component before the
                                   application.  This version will be null if
                                   the application was an add.
component-applied.final-version    The version of the component after the application.
                                   This will be null if the update was a removal.
```

```
Element Type:     initial-version

Elements:         component-name       string     required
                  spec-version         string     required
                  build-version        string     required
                  build-date           string     required
```

Type Detail:

A initial-version instance is used to describe a component level as

the initial version of a component.

Element Detail:

```
initial-version.component-name  The name of the component.
initial-version.spec-version    The new specification version for the
                                component following the update.
initial-version.build-version   The new build version for the component.
initial-version.build-date      The new build date for the component.
```

Element Type: final-version

```
Elements:          component-name        string      required
                   spec-version          string      required
                   build-version         string      required
                   build-date            string      required
```

Type Detail:

A final-version instance is used to supply a component level for a
component which has been added or replaced.

Element Detail:

```
final-version.component-name  The name of the new component.
final-version.spec-version    The new specification version for the
                              component following the update.
final-version.build-version   The new build version for the component.
final-version.build-date      The new build date for the component.
```

Enum Type: enumUpdateType

```
Values:            0 add
                   1 replace
                   2 remove
                   3 patch
```

Type Detail:

An update type instance specifies the type of an update. An 'add' update adds
a component into an installation. A 'replace' update replaces a particular
version of a component with a different version of that component. A 'remove'
update removes a component. A 'patch' update performs a limited update to a
component, in particular, without changing the version of the component.

When adding a component, that component may not already be present.
When replacing or removing a component, that component must be present.
When patching a component, that component must be present.

When replacing or removing a component, or when patching a component, usually,
at least one version prerequisite will be specified for the component update.

Value Detail:

```
enumUpdateType.add      Specifies that an update adds a component.
enumUpdateType.replace  Specifies that an update replaces a component.
enumUpdateType.remove   Specifies that an update removes a component.
enumUpdateType.patch    Specifies that an update modifies a component, but
                        does not change its version.
```


Type Family: history

File Type: event-history

Persistence: <historyDir>/event.history

```
Elements:        update-event          complex    min=0, max=unbounded

Type Detail:

One event history is provided for a websphere product family installation.
This event history contains history of update events, corresponding with
the actual update events for that product family.

Element Detail:

event-history.update-event  The list of update events for the websphere
                            product family.  The top level events are fix
                            and fix pack events, each containing one or more
                            component events.

Element Type:    update-event

Elements:        event-type             enum       required [enumEventType]
                 parent-id              string     required
                 id                     string     required

                 update-type            enum       required [enumUpdateType]

                 is-required            boolean    required
                 is-optional            boolean    required

                 is-external            boolean    required
                 root-property-file     anyURL     optional
                 root-property-name     string     optional
                 root-property-value    string     optional

                 is-custom              boolean    required
                 primary-content        anyURI     required

                 event-action           enum       required [enumEventAction]
                 log-name               anyURI     required
                 backup-name            anyURI     required
                 start-time-stamp       dateTime   required
                 end-time-stamp         dateTime   optional

                 status                 enum       optional [enumEventResult]
                 status-message         string     optional

                 initial-version        complex    optional
                 final-version          complex    optional
                 update-event           complex    optional

Type Detail:

An update event denotes a single update action, applying to either a
fix, a fix pack, or to a component, according to the set event type.

Fix (efix) and fix pack (ptf) type events each have a collection of component events.

Currently, component events have no child events.

Element Detail:

update-event.event-type     The type of this event, either an interim fix or
                            fix pack (ptf) type event, or a component type event.

update-event.parent-id      This element is present only for component
                            events.  The ID of the parent fix or fix pack of
                            this event.

update-event.id             The ID of the fix, fix pack, or component that
                            was updated, interpreted according to the type
```

```
                            of the event.

update-event.update-type      The type of update for component events.

update-event.is-required      A flag which, when true, specifies that this
                              component update is required.

update-event.is-optional      A flag which, when true, specifies that this
                              component update is optional, even if the
                              component is installed.

update-event.is-external       A flag which, when true, specifies that this
                              update used an external root.

update-event.root-property-file  For an update of an external component, this
                              properties file contains the external root value.

update-event.root-property-name  For an update of an external component, the
                              property having this name specifies the external
                              root value.

update-event.root-property-value  For an update of an external component,
                               the root value.

update-event.is-custom        A flag that, when true, specifies that the
                              application was a custom update.  When true,
                              an executable program was applied.  When false,
                              the contents of an update jar were applied.

update-event.primary-content  The URL of the primary content for the update.

update-event.event-action     The type of action for this event.

update-event.log-name         The name of the log file which was generated
                              for this event.

update-event.backup-name      The name of the backup file which was generated
                              for this event.

update-event.start-time-stamp  The XML timestamp of the starting time of the
                              event.  This timestamp follows the XML timestamp
                              format, meaning that time zone information is
                              included.

update-event.end-time-stamp    The XML timestamp of the ending time of the
                              event.  This timestamp follows the XML timestamp
                              format, meaning that time zone information is
                              included.  When absent, the update operation
                              corresponding to the parent event failed with
                              a non-recoverable exception.

update-event.status           The result of the update.

update-event.status-message   Message text provided in addition to the basic
                              status code.  Exception text is provided through
                              the status-message when an update fails.

update-event.initial-version  This element is not used unless the update is
                              a component type update.  The initial version of
                              the component which was updated.    This element
                              is absent when the update is an add type update.

update-event.final-version     This element is not used unless the update is a
                              component type update.  The final version of the
                              component which was updated.  This element is absent
                              when the update is a remove type update.
```

```
update-event.update-event       A collection of child events.  This collection is
                                used for fix and fix pack type events.  This collection
                                is empty for component type events.


Element Type:     initial-version

Elements:        spec-version          string     required
                 build-version         string     required
                 build-date            string     required

Type Detail:

A initial-version instance is used to describe a component level as
the initial version of a component.

Element Detail:

initial-version.spec-version   The new specification version for the
                               component following the update.


initial-version.build-version  The new build version for the component.


initial-version.build-date     The new build date for the component.


Element Type:     final-version

Elements:        spec-version          string     required
                 build-version         string     required
                 build-date            string     required

Type Detail:

A final-version instance is used to supply a component level for a
component which has been added or replaced.

Element Detail:

final-version.spec-version     The new specification version for the
                               component following the update.


final-version.build-version    The new build version for the component.


final-version.build-date       The new build date for the component.


Enum Type          enumEventType

Values:          0 Fix (efix)
                 1 fix pack (ptf)
                 2 Component

Type Detail:

An event type instance specifies the type of an update event, which is either
an interim fix (efix) event, a fix pack (ptf) event or a component event.  The interpretation of
particular event elements depends on the set event type.

Value Detail:

enumEventType.efix        Specifies that an event is for an interim fix update.


enumEventType.ptf         Specifies that an event is for a fix pack update.


enumEventType.component   Specifies that an event is for a component update.


Enum Type:         enumEventAction

Values:          0 Install
```

```
                        1 Uninstall
                        2 Selective install
                        3 Selective uninstall

Type Detail:

An event action instance specified the operation performed by an update, which
can be an install or uninstall operation, and which may be a selective operation.
Component operations are always either install or uninstall type operations, only
fix and fix pack operations may be selective operations.

A selective operation is an installation which is applied to a preset list of
components.  In particular, potential component updates may be skipped, and
component updates which were already applied may be reapplied.

A selective uninstall operation is used to back out an update which was cancelled
by the user.

Value Detail:

enumEventAction.install            Specifies that an event is an install
                                   operation.

enumEventAction.uninstall          Specifies that an event is an uninstall
                                   operation.

enumEventAction.selective-install  Specifies that an event is an install
                                   operation with a preset list of components
                                   which are updated.

enumEventAction.selective-uninstall Specifies that an event is an install
                                   operation with a preset list of components
                                   which are updated.

Enum Type:       enumUpdateType

Values:          0 Add
                 1 Replace
                 2 Remove
                 3 Patch

Type Detail:

An update type instance specifies the type of a component update.  An 'add'
update adds a component into an installation.  A 'replace' update replaces
a particular version of a component with a different version of that component.
A 'remove' update removes a component.  A 'patch' update performs a limited
update to a component, in particular, without changing the version of the
component.

To add a new component, the component must not exist. To replace or remove a component,
the component must exist. To patch a component, the component must exist.

When replacing or removing a component, or when patching a component, usually,
at least one version prerequisite is specified for the component update.

Value Detail:

enumUpdateType.add     Specifies that an update adds a component.

enumUpdateType.replace Specifies that an update replaces a component.

enumUpdateType.remove  Specifies that an update removes a component.

enumUpdateType.patch   Specifies that an update modifies a component, but
                       does not change its version.
```

```
Enum Type:        enumEventResult

Values:           0 Succeeded
                  1 Failed
                  2 Cancelled
```

Type Detail:

An event result instance denotes a particular result for an update event. The result
indicates success, failure, or cancellation.

Value Detail:

`enumEventResult.succeeded` Specifies that the operation was successful.

`enumEventResult.failed`   Specifies that the operation failed.

`enumEventResult.cancelled` Specifies that the operation was cancelled.

Uninstalling WebSphere Application Server

WebSphere Application Server provides an uninstaller program.

If you installed the product on a Windows NT platform, on a drive with a FAT file system, in a path with one or more spaces in its name (such as `C:/Program Files/`, for example), you cannot uninstall the product from the **Add/Remove Software Control Panel** utility. To uninstall the product, change directory to the `install_root\_uninst` directory and run the **uninstall.exe** program.

The uninstaller program removes registry entries, uninstalls the product, and removes all related features and products, such as plug-ins. However, there are some files that the uninstall program does not remove. The uninstall program does not delete any configuration files that are changed as the result of selecting installation options, or running Samples, for example.

Reinstalling: After removing WebSphere Application Server, reinstalling into the same directory without first deleting all its contents, results in invalid XML configurations because of the possibility of retaining some old files. To delete all files so that you can reinstall with a clean system, uninstall manually as described in Step 6.

If you installed the IBM HTTP Server feature using the WebSphere Application Server installation wizard, removing WebSphere Application Server also uninstalls the HTTP Server.

You can also uninstall the product manually.

Depending on the WebSphere Application Server products you installed, uninstall all that you intend to uninstall in this order:
1. WebSphere Application Server Enterprise
2. WebSphere Application Server Network Deployment
3. WebSphere Application Server (base product)

This section contains the following topics:
* Procedure for uninstalling WebSphere Application Server and its features
* Manually uninstalling on AIX platforms
* Manually uninstalling on HP-UX platforms
* Manually uninstalling on Linux platforms
* Manually uninstalling on Solaris platforms
* Manually uninstalling on Windows platforms

Procedure for uninstalling the base WebSphere Application Server product and its features

The time required to uninstall a product depends on the number of configured servers because the uninstall program attempts to stop all running servers. As a result, the time required to uninstall is directly proportional to the number of defined servers. If a configured server is not running, the uninstall program times out, attempting to contact the server, before assuming the server is not running. Therefore, uninstalling many servers can take several minutes. It can run slightly faster if all servers are running.

Steps for this task

1. Uninstall interim fixes and fix packs.
2. Stop any embedded messaging feature services, such as WebSphere Embedded Messaging Publish And Subscribe, JMS servers, or WebSphere MQ Queue Managers, and any related Java processes.
3. Stop the IBM HTTP Server and any related Java processes.
4. Stop any WebSphere Application Server Java processes with the **stopManager** or **stopServer** commands.
5. Run the wizard for removing the program.
   - On Windows platforms, click **Settings > Control Panel > Add/Remove Programs > WebSphere Application Server**. Removing the product this way calls the uninstall program: *WAS_install_root*_uninst\Uninstall.exe

     Uninstalling the WebSphere Application Server product also selects the feature uninstall programs, to uninstall other entries that might appear on the Add/Remove Programs panel, such as the IBM HTTP Server and the IBM WebSphere MQ classes for Java and WebSphere MQ classes for Java Message Service.

     When you install the embedded messaging feature, use the WebSphere Application Server uninstaller wizard to remove the feature. Do not use the Windows Add/Remove Programs utility to uninstall the embedded messaging feature or any other WebSphere Application Server features that might appear in the list.

     You can also call the uninstaller directly from the location shown.
   - On Linux for S/390 platforms, call the uninstall.sh program: *WAS_install_root*/_uninst/uninstall.sh
   - On other Linux platforms, or on UNIX-based platforms, call the uninstall program: *WAS_install_root*/_uninst/uninstall
6. **(Optional)** Uninstall silently, by using the -silent option on the command.

   Silently uninstalling does not display the wizard. Command syntax is the same except for the -silent option. For example:
   - On Windows platforms: *install_root*_uninst\Uninstall.exe -silent
   - On Linux for S/390 platforms: *install_root*/_uninst/uninstall.sh -silent
   - On other Linux platforms, or on UNIX-based platforms: *install_root*/_uninst/uninstall -silent
7. Uninstall manually if you cannot successfully use the uninstaller.

   Manually uninstall the product if the uninstaller program is not present, or if an aborted installation did not create a complete and functional uninstaller program.

   The following platform-specific uninstall procedures each describe a manual uninstall process that guides you through removing every trace of the products and features you might have installed, including directories that might have changed data. Before you begin one of the procedures, back up the `config` and `properties` install-root directories. Backing up the directories lets you refer to the changed data later.
   - Manually uninstalling on AIX platforms
   - Manually uninstalling on HP-UX platforms
   - Manually uninstalling on Linux platforms
   - Manually uninstalling on Solaris platforms

- Manually uninstalling on Windows platforms

Manually uninstalling on AIX platforms

Always use the uninstall program, if it exists. In some cases, such as an aborted installation, the uninstaller program might not exist, or might not be complete and functional. For example, there are several valid scenarios where parts of the embedded messaging feature are not uninstalled, including:

- A message broker is still defined.
- The WebSphere Embedded Messaging Publish and Subscribe Edition (WEMPS) is not removed.
- A message queue manager is still defined.
- The Java Messaging feature and the WebSphere MQ product are not removed.

Use the following procedure to remove all remnants of WebSphere Application Server so that you can reinstall.

Depending on the WebSphere Application Server products you installed, uninstall these products in this order:

1. WebSphere Application Server Enterprise
2. WebSphere Application Server Network Deployment
3. WebSphere Application Server (base product)

In the following steps:

- The default *install_root* for WebSphere Application Server is the `/usr/WebSphere/AppServer` directory.

Steps for this task

1. Use the **kill -9** *<java_pid_1> <java_pid_2>...<java_pid_n>* command to verify no Java processes are running.

   If this is not possible because there are Java processes running that you do not intend to stop, stop all WebSphere Application Server-related processes.

2. Halt any running MQ queue managers.

   a. Type **dspmq** to show the state of any queue managers.

   b. Type **endmqm -i** for each running queue manager.

   c. Type **$ ipcs -a** to check for any IPCs.

   d. Type **$ ipcrm -[qms] [ID]** to delete the IPCs.

3. Type `kill -9 <amq_pid_1> <amq_pid_2> ... <amq_pid_n>` to verify that no MQ processes are running.

4. Run the `install_root/_uninst/uninstall` program for WebSphere Application Server, if it exists.

5. Search for other related packages.

   Either use smit to remove packages, or search for and remove packages manually:

   - Type `smit` to clean up remnants.

     a. Click **Software Installation and Maintenance**.

     b. Click **Software Maintenance and Utilities**.

     c. Click **Remove Installed Software**.

     d. Click **LIST** by Software name.

     e. Search for packages that contain the words **IBM** or **WS** to identify all WebSphere Application Server-related packages, including those that belong to **IBM HTTP Server** and the embedded messaging feature, which is based on **IBM WebSphere MQ** technology.

     f. Change the **PREVIEW ONLY** option to **NO**.

     g. Click **OK**.

- Manually search for, and remove related packages. Type these commands to search for, and remove related packages:

    a. Search for related packages:

        1) `lslpp -l | grep WS` to show packages for the base WebSphere Application Server product and the IBM HTTP Server product

        2) `lslpp -l | grep wemps` to show packages for the embedded messaging feature, which is based on IBM WebSphere MQ technology.

        3) `lslpp -l | grep mqjava` to show more packages for the embedded messaging feature, or packages for the IBM WebSphere MQ product.

        4) `lslpp -l | grep mqm` to show more packages for the embedded messaging feature or the IBM WebSphere MQ product.

            You can also execute this command as root to remove the embedded messaging feature from your system:

            ```
            installp -u wemps mqjava mqm
            ```

            Reply y[es] to all prompts.

            However, do not remove any mqm or mqjava packages if you have installed IBM WebSphere MQ separately.

            The examples below show typical package names that might appear on a system with the base WebSphere Application Server product.

            If no packages appear when using these commands, skip the next step.

    b. Type `smitty remove <packagename1> <packagename2> <packagename3> ...` to remove any WebSphere Application Server-related packages.

        Do not remove IBM WebSphere MQ packages if you have installed IBM WebSphere MQ as a separate product.

        You can remove IBM WEMPS packages.

6. Change directory to the `/usr` directory.

7. Type `rm -rf WebSphere` to delete this WebSphere Application Server-related directory, if the only subdirectory is the `AppServer` directory.

8. Change directory to the `/usr/opt` directory.

9. Type `rm -rf wemps` to delete the embedded messaging feature directory.

10. Change directory to the `/usr` directory.

11. Type `rm -rf IBMHttpServer` to delete the IBM HTTP Server directory.

12. If you do not have IBM WebSphere MQ installed as a separate product on this machine, type `rm -rf mqm` to delete the embedded messaging feature directory.

    If you installed IBM WebSphere MQ as a separate product on this host to use as the WebSphere MQ JMS provider, and do not want to continue using WebSphere MQ, you can uninstall the product as described in the WebSphere MQ information.

13. Edit the `/usr/lib/objrepos/vpd.properties` file.

    a. Remove all lines containing the ″WSB″, ″WSM″, ″WSN″, or ″WSE″ strings.

    b. Save the file and close it.

    Do not delete or rename the `vpd.properties` file because the InstallShield for MultiPlatforms (ISMP) program uses it for other products that it installs.

14. Obtain the odmclean.sh and aixclean.sh scripts from the support Web site and run them.

    a. Edit the script and replace every instance of the string `/usr/WebSphere/AppServer` with the actual install_root.

b. Run the odmclean.sh script from the command line:

```
./odmclean.sh
```

c. Run the aixclean.sh script from the command line:

```
./aixclean.sh
```

When the process completes, the product is completely uninstalled. You are now ready to reinstall.

<u>Usage scenario</u>

Example of displaying package names beginning with mqm, for the embedded messaging feature

```
==>lslpp -l | grep mqm
```

```
  mqm.base.runtime          5.3.0.1  COMMITTED  WebSphere MQ Runtime for
  mqm.base.sdk              5.3.0.1  COMMITTED  WebSphere MQ Base Kit for
  mqm.client.rte            5.3.0.1  COMMITTED  WebSphere MQ Client for AIX
  mqm.java.rte              5.3.0.1  COMMITTED  WebSphere MQ Java Client and
  mqm.msg.De_DE             5.3.0.1  COMMITTED  WebSphere MQ Messages - German
  mqm.msg.Es_ES             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.Fr_FR             5.3.0.1  COMMITTED  WebSphere MQ Messages - French
  mqm.msg.It_IT             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.Ja_JP             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.Zh_CN             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.Zh_TW             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.de_DE             5.3.0.1  COMMITTED  WebSphere MQ Messages - German
  mqm.msg.en_US             5.3.0.1  COMMITTED  WebSphere MQ Messages - U.S.
  mqm.msg.es_ES             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.fr_FR             5.3.0.1  COMMITTED  WebSphere MQ Messages - French
  mqm.msg.it_IT             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.ja_JP             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.ko_KR             5.3.0.1  COMMITTED  WebSphere MQ Messages - Korean
  mqm.msg.pt_BR             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.zh_CN             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.msg.zh_TW             5.3.0.1  COMMITTED  WebSphere MQ Messages -
  mqm.server.rte            5.3.0.1  COMMITTED  WebSphere MQ Server
```

Example of displaying package names beginning with wemps, for the embedded messaging feature

```
==>lslpp -l | grep wemps
```

```
wemps.base.runtime        2.1.0.0  COMMITTED  WebSphere Embedded Messaging
```

Example of displaying package names beginning with WS, for WebSphere Application Server-related products

```
==>lslpp -l | grep WS
```

```
  WSBAA                     5.0.0.0  COMMITTED  ISMP installed entry
  WSBAAAA                   5.0.0.0  COMMITTED  Installs tools for assembly
  WSBAC1AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBACAA                   5.0.0.0  COMMITTED  Includes adminconsole.ear, the
  WSBADAA                   5.0.0.0  COMMITTED  Installs the Administrative
  WSBAS1AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBASAA                   5.0.0.0  COMMITTED  Includes wsadmin, the
  WSBAT1AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBATAA                   5.0.0.0  COMMITTED  Includes a GUI-based tool for
  WSBAU1AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBAUAA                   5.0.0.0  COMMITTED  Includes Apache ANT, a
  WSBCO1AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBCO4AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBCO5AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBCOAA                   5.0.0.0  COMMITTED  ISMP installed entry
  WSBDT1AA                  5.0.0.0  COMMITTED  ISMP installed entry
  WSBDTAA                   5.0.0.0  COMMITTED  Includes a command-line
  WSBES1AA                  5.0.0.0  COMMITTED  ISMP installed entry
```

```
WSBES3AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBESAA                   5.0.0.0  COMMITTED  Includes samples for
WSBGK2AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBGKAA                   5.0.0.0  COMMITTED  ISMP installed entry
WSBIHAA                   1.3.26.x.0 COMMITTED Installs an Apache-powered Web
WSBIHAB                   1.3.26.x.0 COMMITTED  ISMP installed entry
WSBJA1AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBJAAA                   5.0.0.0  COMMITTED  Installs WebSphere public
WSBJD3AA                  1.3.1.0  COMMITTED  ISMP installed entry
WSBJD7AA                  1.3.1.0  COMMITTED  ISMP installed entry
WSBJD9AA                  1.3.1.0  COMMITTED  ISMP installed entry
WSBJDAA                   1.3.1.0  COMMITTED  ISMP installed entry
WSBLA1AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBLAAA                   5.0.0.0  COMMITTED  Includes a graphical utility
WSBMQ1AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBMQ2AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBMQ3AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBMQAA                   5.0.0.0  COMMITTED  Installs Java Messaging
WSBMS2AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBMS4AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBMS5AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBMSAA                   5.0.0.0  COMMITTED  Includes JMS
WSBPL1AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBPLAA                   5.0.0.0  COMMITTED  Installs plugins to configure
WSBPTAA                   5.0.0.0  COMMITTED  Installs tools for performance
WSBSM1AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBSMAA                   5.0.0.0  COMMITTED  Includes samples, including
WSBSR1AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBSR5AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBSR6AA                  5.0.0.0  COMMITTED  ISMP installed entry
WSBSRAA                   5.0.0.0  COMMITTED  Installs the application
```

Manually uninstalling on HP-UX platforms

5.0.1

Always use the uninstall program, if it exists. In some cases, such as an aborted installation, the uninstall program might not exist, or might not be complete and functional. For example, there are several valid scenarios where parts of the embedded messaging feature are not uninstalled, including:

- A message broker is still defined.
- The WebSphere embedded messaging publish and subscribe (WEMPS) function is not removed.
- A message queue manager is still defined.
- The Java Messaging feature and the WebSphere MQ product are not removed.

Use the following procedure to remove all remnants of WebSphere Application Server so that you can reinstall.

Depending on the WebSphere Application Server products you installed, uninstall these products in this order:

1. WebSphere Application Server Enterprise
2. WebSphere Application Server Network Deployment
3. WebSphere Application Server (base product)

In the following steps:

- The default *install_root* for WebSphere Application Server is the `/opt/WebSphere/AppServer` directory.
- The default directory for IBM HTTP Server is `/opt/IBMHttpServer`.
- The default directories for the embedded messaging feature, which is based on WebSphere MQ, are `/opt/wemps` and `/opt/mqm`.

Steps for this task

1. Uninstall interim fixes and fix packs.

2. Uninstall interim fixes and fix packs.

3. Use the **kill -9 *<java_pid_1> <java_pid_2>...<java_pid_n>*** command to verify that no Java processes are running.

   If there are Java processes that you do not intend to stop, stop all WebSphere Application Server-related processes.

4. Halt any running WebSphere MQ queue managers.

   a. Type `dspmq` to show the state of any queue managers.

   b. Type `endmqm -i` for each running queue manager.

   c. Type `$ ipcs -a` to check for any IPCs.

   d. Type `$ ipcrm -[qms] [ID]` to delete the IPCs.

5. Search for mqm processes by typing: `ps -eaf | grep mqm` or `ps -eaf | grep MQ*`. Type `kill -9 <amq_pid_1> <amq_pid_2> ... <amq_pid_n>` to verify that no WebSphere MQ processes are running.

6. Run the `install_root/_uninst/uninstall` program, which is the wizard for removing WebSphere Application Server, if it exists.

   Run the uninstall program silently, by using the -silent option on the command.

7. Search for other related packages.

   Use HP-UX System Administration Manager (SAM) to remove other related packages. Then search for the packages to verify their removal.

   a. Start the SAM utility and verify that your DISPLAY and TERM environment variables are set properly.

      1) Click **Software management**.

      2) Click **View installed software**.

      3) Look for these two entries in the SD list:
         - MQSERIES -> B.11.530.01 WebSphere MQ for HP-UX
         - MQSERIES -> B.11.530.03 WebSphere MQ Update (U485562) for HP-UX

      4) Close the SD list.

      5) Click **Remove local host software**.

      6) On the **SD Remove list**, remove these instances, if they exist:
         - MQSERIES -> B.11.530.01 WebSphere MQ for HP-UX
         - MQSERIES -> B.11.530.03 WebSphere MQ Update (U485562) for HP-UX

         a) Highlight the MQSERIES instances on the SD List.

         b) Click **Actions > Remove...**.

         c) On **Remove Analysis** dialog, verify that **Status** is **ready**.

         d) Click **OK**.

         e) Check the result message displayed on the dialog.

         f) If the removal fails, follow the instructions that appear.

      7) Return to the **SD Remove List**.

      8) Click **IBM HTTP Server, MQSERIES, WEMPS, WSBAA, WSNAA, gsk5bas**.

      9) Click **Actions > Mark for remove**.

      10) Click **Actions > Remove**.

      11) Click **OK** in the **Remove analysis** dialog box.

      12) Click **Logs** to display real-time removal of selected packages.

      13) Click **Done** when all packages are removed.

      14) Exit SAM.

b. Verify that you removed the packages. Type these commands to search for related packages:

1) `swlist | grep WS` to show packages for the base WebSphere Application Server product and the IBM HTTP Server product.

2) `swlist | grep WEMPS` to show packages for the WebSphere Application Server, embedded messaging feature, publish and subscribe function. You can delete these packages.

3) `swlist | grep MQ` to show packages for the embedded messaging feature, or packages for the IBM WebSphere MQ product.

Do not remove IBM WebSphere MQ packages if you have installed IBM WebSphere MQ as a separate product.

8. Change directory to the `/opt/` directory.

9. Type `rm -rf WebSphere` to delete this WebSphere Application Server-related directory. If there are no other entries, you can remove the `WebSphere` directory.

10. Type `rm -rf wemps` to delete the embedded messaging feature directory.

11. Type `rm -rf IBMHttpServer` to delete the IBM HTTP Server directory.

12. If you do not have IBM WebSphere MQ installed as a separate product on this machine, type `rm -rf mqm` to delete the embedded messaging feature directory.

If you installed IBM WebSphere MQ as a separate product on this host to use as the WebSphere MQ JMS provider, and do not intend to continue using WebSphere MQ, you can uninstall the product as described in the WebSphere MQ information.

When the process completes, the product is completely uninstalled. You are now ready to reinstall.

Manually uninstalling on Linux platforms

Always use the uninstall program (or the uninstall.sh program for Linux for S/390 platforms), if it exists. In some cases, such as an aborted installation, the uninstall (or uninstall.sh) program might not exist, or might not be complete and functional. For example, there are several valid scenarios where parts of the embedded messaging feature are not uninstalled, including:

- A message broker is still defined.
- The WebSphere Embedded Messaging Publish and Subscribe Edition (WEMPS) is not removed.
- A message queue manager is still defined.
- The Java Messaging feature and the WebSphere MQ product are not removed.

Use the following procedure to remove all remnants of WebSphere Application Server so that you can reinstall.

Depending on the WebSphere Application Server products you installed, uninstall these products in this order:

1. WebSphere Application Server Enterprise
2. WebSphere Application Server Network Deployment
3. WebSphere Application Server (base product)

In the following steps:

- The default *install_root* for WebSphere Application Server is the `/opt/WebSphere/AppServer` directory.
- The default directory for IBM HTTP Server is `/opt/IBMHttpServer`.
- The default directories for the embedded messaging feature, which is based on WebSphere MQ, are `/opt/wemps` and `/opt/mqm`.

Steps for this task

1. Uninstall interim fixes and fix packs.
2. Use the **killall -9 java** command to verify that no Java processes are running.

If there are Java processes that you do not intend to stop, stop all WebSphere Application Server-related processes.

3. Halt any running WebSphere MQ queue managers.

   a. Type `dspmq` to show the state of any queue managers.

   b. Type `endmqm -i` for each running queue manager.

   c. Type `$ ipcs -a` to check for any IPCs.

   d. Type `$ ipcrm -[qms] [ID]` to delete the IPCs.

4. Type **`kill -9 <amq_pid_1> <amq_pid_2> ... <amq_pid_n>`** to verify that no WebSphere MQ processes are running.

5. Run the `install_root/_uninst/uninstall` program for WebSphere Application Server, if it exists.

   If the program is not present, skip this step.

6. Search for related packages.

   Type these commands to search for related packages:

- **`rpm -qa | grep WS`** to show packages for the base WebSphere Application Server product and the IBM HTTP Server product
- **`rpm -qa | grep MQ`** to show packages for the embedded messaging feature, which is based on IBM WebSphere MQ technology
- **`rpm -qa | grep wemps`** to show more packages for the embedded messaging feature

   The examples below show typical package names that might appear.

   If no packages appear when using these commands, skip the next step.

7. Type **`rpm -e <packagename>`** to remove any WebSphere Application Server-related packages.

   Do not remove IBM WebSphere MQ packages if you have installed IBM WebSphere MQ as a separate product.

   You can remove IBM WEMPS packages.

   For example, execute these commands as root:

```
rpm -e MQSeriesClient-5.3.0-1
rpm -e MQSeriesMsg_Zh_CN-5.3.0-1
rpm -e MQSeriesMsg_Zh_TW-5.3.0-1
rpm -e MQSeriesMsg_de-5.3.0-1
rpm -e MQSeriesMsg_es-5.3.0-1
rpm -e MQSeriesMsg_fr-5.3.0-1
rpm -e MQSeriesMsg_it-5.3.0-1
rpm -e MQSeriesMsg_ja-5.3.0-1
rpm -e MQSeriesMsg_ko-5.3.0-1
rpm -e MQSeriesMsg_pt-5.3.0-1
rpm -e MQSeriesRuntime-5.3.0-1
rpm -e MQSeriesSDK-5.3.0-1
rpm -e MQSeriesJava-5.3.0-1
rpm -e MQSeriesServer-5.3.0-1
rpm -e MQSeriesJava-5.3.0-1
rpm -e wemps-runtime-2.1.0-0
rpm -e wemps-msg-De_DE-2.1.0-0
rpm -e wemps-msg-Es_ES-2.1.0-0
rpm -e wemps-msg-Fr_FR-2.1.0-0
rpm -e wemps-msg-It_IT-2.1.0-0
rpm -e wemps-msg-Ja_JP-2.1.0-0
rpm -e wemps-msg-Ko_KR-2.1.0-0
rpm -e wemps-msg-Pt_BR-2.1.0-0
rpm -e wemps-msg-Zh_CN-2.1.0-0
rpm -e wemps-msg-Zh_TW-2.1.0-0
```

8. Type **`rm -rf /opt/WebSphere/AppServer/`** to remove all WebSphere Application Server directories. If there are no other entries, you can remove the `WebSphere` directory.

9. Type **rm -fr /var/wemps /opt/wemps** if you are certain that there is no embedded messaging data to preserve.

10. If you do not have IBM WebSphere MQ installed as a separate product on this machine, type **rm -fr /var/mqm /opt/mqm** if you are certain that there is no embedded messaging data to preserve.

    If you installed IBM WebSphere MQ as a separate product on this host to use as the WebSphere MQ JMS provider, and do not want to continue using WebSphere MQ, you can uninstall the product as described in the WebSphere MQ information.

11. Edit the `vpd.properties` file.

    a. Locate the `vpd.properties` file in the /root directory.

    b. Remove all lines containing the ″WSB″, ″WSM″, ″WSN″, or ″WSE″ strings.

    c. Save the file and close it.

    Do not delete or rename the `vpd.properties` file because the InstallShield for MultiPlatforms (ISMP) program uses it for other products that it installs.

When the process completes, the product is completely uninstalled. You are now ready to reinstall.

Usage scenario

Example of displaying package names beginning with MQ, for the embedded messaging feature

```
==>rpm -qa | grep MQ
MQSeriesMsg_Zh_CN-5.3.0-1
MQSeriesMsg_Zh_TW-5.3.0-1
MQSeriesMsg_ko-5.3.0-1
MQSeriesClient-5.3.0-1
MQSeriesMsg_de-5.3.0-1
MQSeriesMsg_es-5.3.0-1
MQSeriesMsg_fr-5.3.0-1
WSBMQ1AA-5.0-0
WSBMQ2AA-5.0-0
WSBMQ3AA-5.0-0
MQSeriesMsg_it-5.3.0-1
MQSeriesMsg_ja-5.3.0-1
MQSeriesMsg_pt-5.3.0-1
MQSeriesSDK-5.3.0-1
MQSeriesJava-5.3.0-1
MQSeriesServer-5.3.0-1
MQSeriesRuntime-5.3.0-1
```

Example of displaying package names beginning with wemps, for the embedded messaging feature

```
==>rpm -qa | grep wemps
wemps-msg-De_DE-2.1.0-0
wemps-msg-Es_ES-2.1.0-0
wemps-msg-Fr_FR-2.1.0-0
wemps-msg-It_IT-2.1.0-0
wemps-msg-Ja_JP-2.1.0-0
wemps-msg-Ko_KR-2.1.0-0
wemps-msg-Pt_BR-2.1.0-0
wemps-msg-Zh_CN-2.1.0-0
wemps-msg-Zh_TW-2.1.0-0
wemps-runtime-2.1.0-0
```

Example of displaying package names beginning with WSB, for the base WebSphere Application Server product

```
==>rpm -qa | grep WSB
WSBSR1AA-5.0-0
WSBSR5AA-5.0-0
WSBSR6AA-5.0-0
WSBSM1AA-5.0-0
```

```
WSBAS1AA-5.0-0
WSBGK2AA-5.0-0
WSBC01AA-5.0-0
WSBAC1AA-5.0-0
WSBAT1AA-5.0-0
WSBDT1AA-5.0-0
WSBAU1AA-5.0-0
WSBMQ1AA-5.0-0
WSBMQ2AA-5.0-0
WSBMQ3AA-5.0-0
WSBMS2AA-5.0-0
WSBMS4AA-5.0-0
WSBMS7AA-5.0-0
WSBES1AA-5.0-0
WSBES3AA-5.0-0
WSBIHAB-1.3-26
WSBPL1AA-5.0-0
WSBLA1AA-5.0-0
WSBJA1AA-5.0-0
WSBC04AA-5.0-0
WSBC05AA-5.0-0
WSBJD7AA-1.3-1
WSBJD5AA-1.3-1
WSBJD9AA-1.3-1
```

Manually uninstalling on Solaris platforms

Always use the uninstall program, if it exists. In some cases, such as an aborted installation, the uninstall program might not exist, or might not be complete and functional. For example, there are several valid scenarios where parts of the embedded messaging feature are not uninstalled, including:

- A message broker is still defined.
- The WebSphere Embedded Messaging Publish and Subscribe Edition (WEMPS) is not removed.
- A message queue manager is still defined.
- The Java Messaging feature and the WebSphere MQ product are not removed.

Use the following procedure to remove all remnants of WebSphere Application Server so that you can reinstall.

Depending on the WebSphere Application Server products you installed, uninstall these products in this order:

1. WebSphere Application Server Enterprise
2. WebSphere Application Server Network Deployment
3. WebSphere Application Server (base product)

In the following steps:

- The default *WAS_install_root* for WebSphere Application Server is the `/opt/WebSphere/AppServer` directory.
- The default directory for IBM HTTP Server is `/opt/IBMHttpServer`.

Steps for this task

1. Uninstall interim fixes and fix packs.
2. Use the **kill -9 *<java_pid_1> <java_pid_2>...<java_pid_n>*** command to verify that no Java processes are running.

   If there are Java processes that you do not intend to stop, stop all WebSphere Application Server-related processes.
3. Halt any running WebSphere MQ queue managers.

   a. Type `dspmq` to show the state of any queue managers.

b. Type `endmqm -i` for each running queue manager.

c. Type `$ ipcs -a` to check for any IPCs.

d. Type `$ ipcrm -[qms] [ID]` to delete the IPCs.

4. Type **`kill -9 <amq_pid_1> <amq_pid_2> ... <amq_pid_n>`** to verify that no WebSphere MQ processes are running.

5. Run the `install_root/_uninst/uninstall` program for WebSphere Application Server, if it exists.

   Run the uninstall program silently, by using the -silent option on the command.

6. Search for related packages.

   Type these commands to search for related packages:

   - **`pkginfo | grep WS`** to show packages for the base WebSphere Application Server product and the IBM HTTP Server product.

   - **`pkginfo | grep wemps`** to show packages for the WebSphere Application Server embedded messaging feature publish and subscribe function. You can delete these packages.

   - **`pkginfo | grep mqm`** to show packages for the embedded messaging feature or the IBM WebSphere MQ product, if you have installed it.

     Do not remove IBM WebSphere MQ packages if you have installed IBM WebSphere MQ as a separate product.

   The examples below show typical package names that might appear on a system with the base WebSphere Application Server product installed.

   If no packages appear when using these commands, skip the next step.

7. Type **`pkgrm <packagename1> <packagename2> <packagename3> ...`** to remove any WebSphere Application Server-related packages.

   For example, execute this command as root to remove the embedded messaging feature from your system:

   `pkgrm wemps mqjava mqm-upd04 mqm`

   Reply y[es] to all prompts.

   Do not remove mqjava, mqm-upd04, or mqm packages if you have installed IBM WebSphere MQ as a separate product.

   You can remove wemps packages.

   Note: Remove the **mqm** package last, or you cannot remove any other embedded messaging feature packages. You will have to reinstall the embedded messaging feature to remove the packages.

   Alternatively, you can type these commands to search for and remove any WebSphere Application Server-related packages:

   a. **`ls |grep WSB|xargs -i pkgrm -n {}`** for the base WebSphere Application Server product

   b. **`ls |grep WSC|xargs -i pkgrm -n {}`** for the WebSphere Application Server Java client

   c. **`ls |grep wemps|xargs -i pkgrm -n {}`** for the WebSphere embedded messaging publish and subscribe function

   d. **`ls |grep mqjava|xargs -i pkgrm -n {}`** for IBM WebSphere MQ

   e. **`ls |grep mqm|xargs -i pkgrm -n {}`** for IBM WebSphere MQ

8. Change directory to the `/opt/` directory.

9. Type **`rm -rf WebSphere`** to delete this WebSphere Application Server-related directory, if there is only the `AppServer` subdirectory.

10. Change directories to the `/opt` directory.

11. If you do not have IBM WebSphere MQ installed as a separate product on this machine, type **rm -rf IBMHttpServer/ wemps/ mqm/** to delete the IBM HTTP Server directory, and the two embedded messaging feature directories.

If you installed IBM WebSphere MQ as a separate product on this host to use as the WebSphere MQ JMS provider, and do not want to continue using WebSphere MQ, you can uninstall the product as described in the WebSphere MQ information.

When the process completes, the product is completely uninstalled. You are now ready to reinstall.

<u>Usage scenario</u>

Example of displaying package names beginning with mqm, for the embedded messaging feature

```
pkginfo |grep mqm

application mqm          WebSphere MQ for Sun Solaris
```

Example of displaying package names beginning with wemps, for the embedded messaging feature

```
pkginfo |grep wemps

application wemps        WebSphere Embedded Messaging Publish and Subscribe Edition
```

Example of displaying package names beginning with WSB, for the base package

```
pkginfo | grep WSB

application WSBAA         WebSphere Application Server
application WSBAAAA       Application And Assembly Tools
application WSBAC1AA      adminConsoleFilesComponent
application WSBACAA       Admin Console
application WSBADAA       Admin
application WSBAS1AA      adminScriptingFilesComponent
application WSBASAA       Admin Scripting
application WSBAT1AA      applicationAssemblyToolComponent
application WSBATAA       Application Assembly Tool
application WSBAU1AA      antUtilityComponent
application WSBAUAA       ANT Utility
application WSBCO1AA      commonFiles
application WSBCO4AA      pbwServerConfigWithMQGood
application WSBCO5AA      IsmpLauncherComponent
application WSBCOAA       commonFeature
application WSBDM1AA      DCMStdComponent
application WSBDMAA       Dynamic Cache Monitor
application WSBDT1AA      deployToolComponent
application WSBDTAA       Deploy Tool
application WSBES1AA      messagingSampleFileComponentBean
application WSBES3AA      mdbServerConfigWithMQGoodUnix
application WSBESAA       mqSeriesSamples
application WSBGK2AA      gskitUnixComponent
application WSBGK3AA      gskit4SolarisComponent
application WSBGKAA       gskitFeature
application WSBIHAA       ihsFeature
application WSBIHAB       ihsComponent
application WSBJD9A       javaCommonConfigComponent
application WSBJA1AA      javadocComponent
application WSBJAAA       Javadoc
application WSBJD4AA      javaSolarisComponent
application WSBJD7AA      javaUninstallComponent
application WSBJDAA       Java
application WSBLA1A       logAnalyzerComponent
application WSBLAAA       LogAnalyzer
application WSBMQ1AA      mqSeriesSetupFileComponent
application WSBMQ2AA      mqSeriesBinComponent
application WSBMQ3AA      mqSeriesLibFilesComponent
application WSBMQAA       MQSeries
```

```
application WSBMS2AA    mqSeriesUnixPrereqBean
application WSBMS4AA    mqSeriesUnixInstall
application WSBMS6AA    mqSeriesSunConfig
application WSBMSAA     mqSeriesServer
application WSBPL1AA    component8
application WSBPL21AA   Domino
application WSBPLAA     Plugins
application WSBPS1AA    perfServletComponent
application WSBPSAA     Performance Servlet
application WSBPTAA     PerformanceAndAnalysisTools
application WSBSM1AA    samplesComponent
application WSBSMAA     Samples
application WSBSR1AA    serverStdComponent
application WSBSR5AA    serverConfigWithSamplesComponent
application WSBSR6A     serverCommonConfigComponent
application WSBSRAA     Server
application WSBTV1AA    tivoliViewerComponent
application WSBTVAA     TivoliPerformanceViewer
```

Manually uninstalling on Windows platforms

Always use the Uninstall.exe program, if it exists. In some cases, such as an aborted installation, the **Uninstall.exe** command might not exist, or might not be complete and functional. For example, there are several valid scenarios where parts of the embedded messaging feature are not uninstalled, including:

- A message broker is still defined.
- The WebSphere Embedded Messaging Publish and Subscribe Edition (WEMPS) is not removed.
- A message queue manager is still defined.
- The Java Messaging feature and the WebSphere MQ product are not removed.

Use the following procedure to remove all remnants of WebSphere Application Server so that you can reinstall.

There are three items you must delete to remove a WebSphere Application Server product or feature: files, registry entries, and the MSI record (when you have installed the embedded messaging feature). If you are uninstalling in a coexistence environment, there are registry entries only for the installation image you installed last. If you are removing any version but the last image you installed, you need delete only the file structure for the installation image you are deleting. You need not delete registry entries or the MSI record.

Depending on the WebSphere Application Server products you installed, uninstall these products in this order:

1. WebSphere Application Server Enterprise
2. WebSphere Application Server Network Deployment
3. WebSphere Application Server (base product)

In the following steps:

- The default installation root for WebSphere Application Server is the `<drive>:\Program Files\WebSphere\AppServer` directory.
- The default directory for the IBM HTTP Server feature is the `<drive>:\Program Files\IBMHttpServer` directory.
- The default directory for the embedded messaging feature is the `<drive>:\Program Files\IBM\WebSphere MQ` directory.
- The default directory for the IBM Key Management utility (for Web server support) is the `<drive>:\Program Files\IBM\gsk5` directory.

Steps for this task

1. Uninstall interim fixes and fix packs.
2. Verify that you have an Emergency Recovery Disk. Instructions for creating this disk are in the Windows help documentation.
3. Use the `regback.exe` program from the Windows Resource Kit to back up the Registry.
4. Run the `install_root\_uninst\Uninstall.exe` program for WebSphere Application Server, if it exists.
5. Restart the machine as the Uninstall.exe program suggests.
6. If you have not installed IBM WebSphere MQ as a separate product on this machine, check that the WebSphere embedded messaging has been uninstalled.
   a. Use Add/Remove Programs to check if the following programs are still listed:
      ```
      IBM WebSphere EMPS
      IBM WebSphere MQ
      ```
   b. If either of these programs is still listed, complete the following steps:
      1) Delete any Broker or Queue Manager defined for WebSphere Application Server.
         - To delete any Broker defined for WebSphere Application Server, run the **wempsdeletebroker.exe** program, if it still exists:
      ```
      <drive>:\IBM\WebSphere MQ\WEMPS\bin\wempsdeletebroker.exe brokername -q
      ```

         The -q option deletes the Broker's WebSphere MQ queue manager. For example:
      ```
      <drive>:\IBM\WebSphere MQ\WEMPS\bin\wempsdeletebroker.exe WAS_nodename_server1 -q
      ```
         - To delete any Queue Manager defined for WebSphere Application Server, run the **dltmqm.exe** program, if it still exists:
      ```
      <drive>:\IBM\WebSphere MQ\bin\dltmqm.exe queuemanagername
      ```
      2) Use Add/Remove Programs to remove the following programs, if listed:
      ```
      IBM WebSphere EMPS
      IBM WebSphere MQ
      ```

         Note: If you installed IBM WebSphere MQ as a separate product, do not remove it now.
   c. Remove the WebSphere MQ Windows tray icon if it present.

      The MQ Windows tray icon in the lower right corner indicates that a WebSphere MQ process (**amqmtbrn.exe**) is running. Remove the tray icon to close the process.

      To remove the tray icon, right click the icon and click **Hide**.
7. **(Optional)** Delete WebSphere Application Server product registry entries.

   This step is optional in a coexistence environment. Perform this step only if you are removing the image that you last installed.

   Remove any WebSphere Application Server product or feature that appears in the Add/Remove Programs panel, which is available from the Control Panel.
8. **(Optional)** Invoke `regedit.exe` from a command prompt, to edit the Windows System Registry.

   This step is optional in a coexistence environment. Perform this step only if you are removing the image that you last installed.

   Handle the Registry with care!: You can easily make a mistake while using the regedit.exe editor to view and edit Registry contents. The editor does not warn you of editing errors, which can be extremely dangerous. A corrupt Registry can disrupt your system to the point where your only option is to reinstall the Windows operating system.
   a. Search (using **Ctrl-F**) for all instances of *WebSphere*, *IBM HTTP Server*, or *IBM MQSeries*, to determine whether you should delete each entry. You might not be able to remove all of the entries related to WebSphere Application Server, which is not a problem.
   b. Expand and select keys related to **WebSphere Application Server** and its features, **IBM HTTP Server**, and **IBM WebSphere MQ**.

      Keys to delete include:

- **HKEY_LOCAL_MACHINE\ SOFTWARE\IBM\ HTTP Server\ 1.3.26.x**
- **HKEY_LOCAL_MACHINE\ SOFTWARE\IBM\ WebSphere Application Server\ 5.0.0.0**
- **HKEY_LOCAL_MACHINE\SOFTWARE\ IBM\ WebSphereEmbeddedMessagingPublishAndSubscribe**
- If you have not installed IBM WebSphere MQ as a separate product on this machine, delete the following key for WebSphere embedded messaging.

   If you have installed IBM WebSphere MQ as a separate product, do not delete this key.
    - **HKEY_LOCAL_MACHINE\ SOFTWARE\IBM\ MQSeries\ CurrentVersion**
   c. Click **Edit > Delete** from the menu bar for each related key.
   d. Click **Yes** when asked to confirm deletion of the key.
   e. Click **Registry > Exit** from the menu bar when you are finished.
9. Restart the Windows system.

   Reboot to install any of the products again.
10. Use the Microsoft MSI Cleanup Utility to remove MRI records.

    The utility is available from the Microsoft Web site as `msicuu.exe`. Click on the `msicuu.exe` file after downloading it, to install the utility. Once installed, the utility appears on the program menu.

    When MSI starts, it lists all products that it knows about. To uninstall using this technique:
    a. Select a product and click **Remove** to remove its MSI record.

       These WebSphere MQ product entries might be present:
       - IBM WebSphere EMPS
       - IBM WebSphere MQ

       Do not remove the entry for IBM WebSphere MQ if you installed the product separately.
    b. Restart your machine.
11. Delete the base product installation root directory, `<drive>:\WebSphere\AppServer` and all subdirectories.
12. Delete the IBM HTTP Server installation root, which is usually in the `<drive>:\Program Files\IBMHttpServer` directory by default.
13. Delete the **<drive>:\IBM\WebSphere MQ\WEMPS** directory.
14. Depending on whether you installed IBM WebSphere MQ as a separate product, delete the WebSphere MQ directory.

    If you do not have IBM WebSphere MQ installed as a separate product on this machine, delete the **<drive>:\IBM\WebSphere MQ** directory for WebSphere embedded messaging.

    If you installed IBM WebSphere MQ as a separate product on this host to use as the WebSphere MQ JMS provider, and do not want to continue using WebSphere MQ, you can uninstall the product as described in the WebSphere MQ information.
15. Edit the `vpd.properties` file.
    a. Locate the `vpd.properties` file in the operating system installation directory.

       For example, `C:\WINDOWS` or `C:\WINNT`.
    b. Remove all lines containing the ″WSB″, ″WSM″, ″WSN″, or ″WSE″ strings.
    c. Save the file and close it.

    Do not delete or rename the `vpd.properties` file because the InstallShield for MultiPlatforms (ISMP) program uses it for other products that it installs.

Installation: Resources for learning

Use the following links to find relevant supplemental information about installation. The information resides on IBM and non-IBM Internet sites, whose sponsors control the technical accuracy of the information.

These links are provided for convenience. Often, the information is not specific to the IBM WebSphere Application Server product, but is useful all or in part for understanding the product. When possible, links are provided to technical papers and Redbooks that supplement the broad coverage of the release documentation with in-depth examinations of particular product areas.

This section contains the following topics:
- Planning, business scenarios, and IT architecture
- Programming model and decisions
- Programming instructions and examples
- Programming specifications
- Administration
- Support

Planning, business scenarios, and IT architecture

- **IBM WebSphere Application Server supported hardware, software, and APIs** at
  http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html

  The official site for determining product prerequisites for hardware, software and APIs for all WebSphere Application Server products.

- **IBM WebSphere Developer Domain** at http://www7b.software.ibm.com/wsdd/

  The home of technical information for developers working with WebSphere products. You can download WebSphere software, take a fast path to developer domain zones, such as VisualAge Java or WebSphere Application Server, learn about WebSphere products through a newcomers page, tutorials, technology previews, training, and Redbooks, get answers to questions about WebSphere products, and join the WebSphere community, where you can keep up with the latest developments and technical papers.

- **IBM WebSphere Application Server library and InfoCenters Web site** at http://www-3.ibm.com/software/webservers/appserv/infocenter.html

  The IBM WebSphere Application Server Library Web site contains links to all WebSphere Application Server InfoCenters, for all versions. It also lets you access each InfoCenter in your native language.

- **IBM WebSphere Application Server home page** at
  http://www.ibm.com/software/webservers/appserv/

  The IBM WebSphere Application Server home page contains useful information, including support links and downloads for e-fixes, tools, and trials.

- **IBM WebSphere software platform home page** at http://www.ibm.com/websphere

  The IBM WebSphere software platform home page introduces WebSphere products and describes how companies can easily transform to an e-business, with software that can grow as fast as the business it supports.

- **Migrating to WebSphere V5.0: An End-to-End Migration Guide, SG24-6910-00** at
  http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246910.html

  This IBM Redbook is the definitive migration guide for migrating earlier versions of WebSphere Application Server to Version 5. Read this book to formulate an optimal migration strategy.

- **Just announced: IBM WebSphere Application Server Version 5!** at http://www-3.ibm.com/software/info1/ websphere/ index.jsp?tab=products/ appserv

  The IBM WebSphere Application Server Version 5 product announcement.

- **What's new in IBM WebSphere Application Server Version 5?** at http://www-3.ibm.com/software/info1/ websphere/ index.jsp?tab=products/ appserv_whatsnew

  The official list of new features at the What's new in IBM WebSphere Application Server Version 5? Web page.

- IBM **WebSphere Application Server Version 5 fact sheet** at http://www-3.ibm.com/software/webservers/ appserv/v5/ appsvrv5factsheet.pdf

  This fact sheet describes IBM WebSphere Application Server Version 5.

- **The power of Edge of Network technology in IBM WebSphere Application Server Version 5** at http://www-3.ibm.com/software/ info1/ websphere/ index.jsp?tab=products/ appserv_edge

  A description of WebSphere Application Server Edge Components Version 5.

- **WebSphere Application Server - Express, V5** at http://www-3.ibm.com/software/info1/ websphere/index.jsp?tab=products/ appserv_express

  A description of WebSphere Application Server Express, Version 5.

- **WebSphere Application Server, Version 5** at http://www-3.ibm.com/software/info1/ websphere/index.jsp?tab=products/ appserv

  A description of the base product, WebSphere Application Server, Version 5.

- **WebSphere Application Server Enterprise, Version 5** at http://www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/ appserv_enterprise

  A description of WebSphere Application Server Enterprise, Version 5.

- **IBM WebSphere Application Server for z/OS, Version 5** at http://www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/ appserv_zos

  A description of WebSphere Application Server for z/OS, Version 5.

- **InfoCenter for WebSphere Application Server Edge components** at http://www-3.ibm.com/software/webservers/appserv/ecinfocenter.html

  The InfoCenter for WebSphere Application Server Edge components contains complete documentation for the Caching Proxy and the Load Balancer in these PDF online books, *WebSphere Application Server Concepts, Planning, and Installation for Edge Components*, the *WebSphere Application Server Caching Proxy Administration Guide*, and the *WebSphere Application Server Programming Guide for Edge Components*.

- **IBM's Web Services architecture debuts** at http://www.ibm.com/developerworks/webservices/library/w-int.html?dwzone=webservices

  Introducing IBM Web Services, a distributed software architecture of service components. This brief overview and in-depth interview on IBM developerWorks covers the fundamental concepts of Web Services architecture and what they mean for developers. The interview with Rod Smith, Vice President of Emerging Technologies at IBM, explores which types of developers Web Services targets, how Web services reduce development time, what developers can do with Web Services now, and takes a glance at the economics of dynamically discoverable services.

- **developerWorks: Patterns for e-business: Redbooks listing** at http://www.ibm.com/developerworks/patterns/library/index.html#redbooks

  This Web page lists links to pattern resources under these categories:
  - Current patterns Redbooks
  - Superseded patterns Redbooks (valid for back-level product versions)
  - Independent analyst reports
  - Patterns CD order offer
  - Back-level version patterns Web site (zip downloads and old Flash tutorial)
  - Customer references
  - White papers
  - Multimedia presentations and screen cams
  - Webcasts
  - Patterns development kit
  - WebSphere technical exchange presentations

- developerWorks: IBM Patterns for e-business at
  http://www.ibm.com/developerworks/patterns/index.html

  The IBM developerWorks site is the source for IBM patterns for e-business, a set of tested, reusable intellectual assets that you can use to design and implement your e-business network and architecture!

- Self-Service: Select application pattern at
  http://www.ibm.com/developerworks/patterns/u2b/select-application-topology.html

  This Web page describes the self-service business pattern, also known as the User-to-Business or U2B pattern. The self-service e-business pattern captures the essence of direct interactions between interested parties and an e-business. Interested parties include customers, business partners, stakeholders, employees, and all other individuals with whom the business intends to interact.

- Self-Service: stand-alone single channel application pattern: Run-time patterns at
  http://www.ibm.com/developerworks/patterns/u2b/at1-runtime.html

  This Web page describes alternative run-time solution patterns for the self-service business pattern. These run-time patterns are important concepts that any e-business designer should become familiar with.

- Patterns for e-business: A Strategy for Reuse at
  http://www.mcpressonline.com/ibmpress/5206.htm

  Get an inside look at how successful businesses build their e-business architectures. In this book, four IBM e-business experts capture years of experience into easy-to-follow guidelines. Deliberately focusing on Business patterns, integration patterns, and application patterns, the authors share with you proven architectural patterns that can help get you up and running quickly, while at the same time reducing your risks. Because today's economy demands that e-business initiatives emphasize profitability and return on investment, the authors also offer guidance on methods to minimize cost, yet verify quality.

- Patterns and Web Services at http://www.ibm.com/developerworks/patterns/guidelines/web-services.pdf

  Patterns architects have reviewed the impact emerging Web Services technologies have on each of the asset layers of the Patterns for e-business designs. Their findings are summarized in this new White paper, in PDF Format.

- developerWorks: Facilitating the application development process using the IBM Patterns for e-business at http://www.ibm.com/developerworks/patterns/guidelines/lord.pdf

  This is the most recent White paper from John Lord, a well known IBM consulting IT architect. The paper analyzes the successful use of the Patterns for e-business in an application development scenario.

- Self-Service Patterns using WebSphere Application Server, Version 4.0, SG24-6175-00 at
  http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246175.html

  This Redbook discusses the stand-alone Single Channel application pattern of the Self-Service business patterns. This application pattern describes a situation where you are building an application that has no need to connect to backend or legacy data. The Directly Integrated Single Channel application pattern extends this discussion to describe the situation where you need to access existing data on legacy or third-party systems. Although we do not implement the Directly Integrated Single Channel application pattern in this project, the discussions here are relevant.

- Patterns: Connecting Self-Service Applications to the Enterprise, SG24-6572-00 at
  http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246572.html

  This Redbook discusses the Self-Service::Directly Integrated Single Channel application pattern, which covers Web applications needing one or more point-to-point connections with backend applications.

- Self-Service Applications using IBM WebSphere Application Server V4.0 and IBM MQSeries Integrator, SG24-6160-01 at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246160.html

This Redbook focuses on the task of designing and implementing a self-service application using the Router application pattern and the Decomposition application pattern, as defined by the IBM Patterns for e-business.

The Router application pattern provides intelligent routing from multiple clients to multiple backend applications using a hub-and-spoke architecture. The interaction between the user and the backend application is a one-to-one relation, meaning the user interacts with applications one at a time. The primary business logic resides in the backend tier. This book shows how to use IBM MQSeries Integrator and IBM WebSphere Application Server to implement a router type application.

The Decomposition application pattern expands on the router pattern, providing all the features and functions of that pattern and adding recomposition/decomposition capability. It provides the ability to take a user request and decompose it into multiple requests to route to multiple backend applications. The responses are recomposed into a single response for the user. This action moves some of the business logic into the decomposition tier, but the primary business logic still resides in the back-end application tier. The decomposition and recomposition functions are illustrated in this book using IBM WebSphere Application Server, IBM MQSeries Integrator and the IBM MQSI Aggregator Plug-In. The JMS listener provided by WebSphere Application Server Enterprise Services is also illustrated in this example.

- **Applying the Patterns for e-business to Domino and WebSphere Scenarios, SG24-6255-00** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246255.html

This Redbook describes Application Integration patterns, and how they form Composite Patterns, together with one or more of the other Patterns for e-business. It looks at run-time patterns for Domino and WebSphere Application Server integration, and identifies Composite Patterns in action. Some of the patterns include Lotus Sametime and Tivoli Policy Director.

A major part of the book describes three real-life scenarios where Patterns for e-business are applied, with Domino and WebSphere Application Server as part of the run-time topology. Starting from the business requirements phase, the book identifies and applies Business, Application, and Run-time patterns to get to the final run-time topology. It starts with a simple scenario, which becomes increasingly complex in later scenarios. It discusses technology options, as well as design and development guidelines.

- **User-to-Business Pattern Using WebSphere Personalization Patterns for e-business Series, SG24-6213-00** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246213.html

This Redbook describes what was once referred to as the User-to-Business Topology 7 Personalization Pattern. At the time the book was written, the pattern was an emerging pattern. It focuses on direct interactions between users and a business. The pattern helps guide the design of systems with a consolidated customer-centric view that you can exploit for sophisticated personalization and cross-selling opportunities.

This Redbook provides examples and guidelines for the User-to-Business Topology 7 Personalization Pattern. It shows how the pattern works and documents the tasks required to build an example of the pattern.

- **Mobile Applications with IBM WebSphere Everyplace Access Design and Development, SG24-6259-00** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246259.html

This Redbook provides application designers and developers with a broad overview of mobile e-business application design and development using the WebSphere Everyplace Access V1R1 offering.

The book gives an overview of the Patterns for e-business and shows how to use the Patterns in the mobile e-business environment. It also discusses the design and development guidelines for mobile e-business applications using the products bundled in the WebSphere Studio and Visual Age for Java offerings.

This book provides detailed information about the Sample application, by discussing scenarios and implementing mobile applications exercising different techniques for several type of clients. It also

provides detailed instructions for setting up the development and run-time environment for WebSphere Application Server, WebSphere Transcoding Publisher and WebSphere Voice Server together with the Sample shipped with the Redbook.

- **Access Integration Pattern using IBM WebSphere Portal Server, SG24-6267-00** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246267.html

  This Redbook describes the Access Integration pattern, which is an emerging pattern that describes services and components commonly required to provide users with consistent, seamless, device-independent access to relevant applications and information.

  This Redbook provides an example and guidelines for the Access Integration pattern. It shows how the Pattern works and documents the tasks required to build an example.

- **WebSphere Commerce Suite V5.1 for iSeries, Implementation and Deployment Guide, REDP0159** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0159.html

  This Redpaper describes the great benefit of the IBM Framework for e-business, to develop applications on the Windows platform and then deploy the application to one of the many supported WebSphere Application Server platforms, without changing the application.

  This Redpaper introduces the Patterns for e-business and the Electronic Commerce composite pattern used for building e-commerce Web sites. The focus of the Redpaper is on the iSeries unique implementation, deployment and development considerations when using IBM WebSphere Commerce Suite V5.1, Pro Edition for iSeries.

  It includes detailed procedures for implementing WebSphere Commerce Suite V5.1, Pro Edition for iSeries in single-tier and multiple tier run-time environments. Advanced configuration instructions are provided for integrating WebSphere Payment Manager, enabling Secure Socket Layer (SSL) for the HTTP Server, and configuring a secure VPN connection for Open Servlet Engine (OSE) Remote. Once the run-time environment is configured, there is a detailed description of the steps necessary to deploy a WebSphere Commerce Suite store to a WebSphere Commerce Suite V5.1, Pro Edition for iSeries run-time environment.

- **e-commerce Patterns for z/Linux Using WebSphere Commerce Suite V5.1 Patterns for e-business series, REDP0411** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0411.html

  This Redpaper describes the installation, configuration and customization of IBM WebSphere Commerce Suite Pro Edition for Linux for e-server z900 and S/390. It is intended for both technicians who need to install and administer Commerce Suite, and for developers who need to design and customize e-commerce sites for deployment on IBM WebSphere Commerce Suite Pro Edition for Linux for e-server z900 and S/390.

  This Redpaper is part of the Patterns for e-business series and reuses information developed in the Redbook B2C e-commerce Composite pattern using WebSphere Commerce Suite V5.1, SG24-6180.

- **Integrating WebSphere Commerce Suite With a backend Order Management Application, REDP0514** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0514.html

  This Redpaper describes a scenario that presents a fictitious manufacturing company called MANCO, which produces bicycle parts and accessories. MANCO uses the J.D. Edwards OneWorld application on an iSeries server for the Enterprise Resource Planning (ERP) system. MANCO wanted to take advantage of the Internet global connectivity to give its business customers a new way to buy products and to check order status while augmenting the manual sales processes with direct electronic sales to its business customers. MANCO requirements best fit the user-to-online buying business pattern, which is a subset of the user-to-business pattern.

- **Connect for iSeries with WebSphere Commerce Suite: BtoB Enabling a WebSphere Commerce Suite Web Site, REDP0127** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0127.html

  This Redpaper describes another MANCO scenario, from another iSeries development team in the IBM Rochester laboratory.

This scenario presents a fictitious manufacturing company called MANCO, which produces bicycle parts and accessories.

The e-marketplace has emerged as a means of connecting buyers and suppliers. Suppliers look for opportunities to grow their business and expand their customer base. An e-marketplace provides that ability. Many suppliers have already invested in e-commerce solutions and would like to leverage their current solution to enter into the e-marketplace arena.

This Redpaper details the migration of the existing MANCO e-Commerce Web site to an e-marketplace-enabled Web site, by utilizing the WebSphere Commerce Suite and Connect for iSeries products.

- **e-Marketplace Pattern using WebSphere Commerce Suite, MarketPlace Edition Patterns for e-business Series, SG24-6158-00** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246158.html

This Redbook describes the Business to Business e-Marketplace Pattern, which supports the development of e-Marketplace hub applications that bring multiple buyers and sellers together for efficient electronic trading of goods and services. Subsets of the application topologies for the Business to Business e-Marketplace Pattern are used to describe different parts of the full marketplace topology, and they represent increasing levels of complexity, functionality and integration in the topology, ranging from a simple e-Marketplace to a fully integrated e-Marketplace.

- **Business-to-Business Integration Using MQSeries and MQSI, Patterns for e-business Series, SG24-6010-00** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246010.html

This Redbook describes what used to be known as Business-to-Business Integration patterns two and three, which form the basis for many complex and more fully functioned B2B patterns. It is relevant to all enterprises dealing with partner integration issues over the Internet.

Application topology 2 describes a scenario in which messages are being passed between two enterprise applications and no routing is performed. Topology 3 extends topology 2 to describe the scenario where routing is required for multiple cross enterprise applications to communicate.

- **Business Process Management using MQSeries and Partner Agreement Manager, SG24-6166-00** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246166.html

This Redbook describes business process management.

Business process management is the answer for addressing the business issues of the economic world. The velocity of change, driven by the dynamics of commerce and e-business, force you to have an IT system that is ready to change rapidly, and continually. Another issue is the integration of the Internet within business processes and more general multichannel delivery. Companies must be able to provide a consistent service across all channels. To achieve this, you need to have an integrated business view. Business Process Management is the externalization and formalization of knowledge and expertise within applications and minds. That externalization makes it possible to stay in control of your business.

This Redbook introduces the concepts of business process management and its relationship with business-to-business technologies. In the second part of the book, it explores an example of a business process built in MQSeries Workflow. The third part of the book extends this intra-enterprise business process to include collaboration with external companies using WebSphere Business-to-Business Partner Agreement Manager.

The final part of the book takes one step back and looks in more general terms at the patterns of developing and deploying business-to-business solutions.

- **Design for Scalability - An Update** at http://www7b.software.ibm.com/wsdd/library/techarticles/hvws/scalability.html

This White paper is from the IBM High Volume Web Sites team. The White paper describes component selection and management techniques you can use to make your Web site ready to adapt to increasing traffic. These techniques are the product of IBM experiences while working with customers seeking to improve the performance and availability of some of the largest Web sites in the world.

Abstract: Optimizing for scalability remains a significant challenge for e-businesses as they balance the demands for availability, reliability, security, and high performance. Vendors are responding with

infrastructure options and supporting hardware and software platforms that address these requirements. This update identifies current products and emerging trends that are most likely to improve the scalability of your e-business infrastructure.

- **IBM WebSphere V4.0 Advanced Edition Handbook** at http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246176.html

  This Redbook describes base application topologies and product mappings for WebSphere Application Server. Refer to the **IBM Redbooks** Web site at http://www.redbooks.ibm.com/ for the latest update.

- **User centered design (UCD) for different project types, part 1** at http://www-106.ibm.com/developerworks/usability/library/us-ucd/

  This Web page is the first of two articles posted to the IBM develperWorks domain that describes useful application design activities for different types of projects.

- **User centered design (UCD) for different project types, part 2** at http://www.ibm.com/developerworks/library/us-ucd2/index.html?dwzone=usability

  This Web page is the latest of two articles that describes design activities that IBM scientists have found most useful in various types of projects. This article defines user interface design elements, including the design prototype, use case model, and design specification document.

Programming model and decisions

- **Designing e-business Solutions for Performance** at http://www.ibm.com/developerworks/patterns/ebusiness-performance-customer-v2.pdf

  This White paper describes how the design or implementation of an e-business application can affect performance.

- **Managing Web Site Performance** at http://www.ibm.com/ developerworks/patterns/guidelines/ HTTP_Session_Best_Practice.pdf

  This White paper contains tips and techniques for developers building applications that use session persistence. It also helps administrators to tune the WebSphere Application Server product appropriately for these applications.

Programming instructions and examples

- **IBM developerWorks** at http://www.ibm.com/developerworks/

  IBM developerWorks contains many excellent resources for developers, including tutorials on Web development-related topics. There is an excellent tutorial on the JDBC API.

- **IBM Redbooks** at http://www.redbooks.ibm.com/

  The IBM Redbooks site contains many WebSphere Application Server related documents.

- **Servlets and JavaServer Pages - A Tutorial** at http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/

  Tutorial from the author of Core Servlets and JavaServer Pages.

Programming specifications

- **J2EE information** at http://java.sun.com

  For more information about J2EE specifications, visit the Sun site.

- **sun.net.inetaddr.ttl property** at http://java.sun.com/j2se/1.4/docs/guide/net/properties.html

  The following Java 2 SDK, Standard Edition 1.4 Web site describes the private sun.net.inetaddr.ttl property, which also works in Java 2 SDK, Standard Edition 1.3.

- **java.net.URLConnection class** at http://java.sun.com/j2se/1.4.1/jcp/beta/

  The *Networking* section of this Java 2 SDK, Standard Edition 1.4 Web site describes a change in the behavior of the java.net.URLConnection class.

Administration

*   **Best Practices Zone on WSDD** at http://www7b.boulder.ibm.com/wsdd/zones/bp/

    The WebSphere Best Practices Zone is a collection of best practices for administering WebSphere Application Server. Over time, the zone is intended to grow to include best practices for using other WebSphere software products, and to cover more topics. Use the feedback mechanism to submit your best practice suggestions.

*   **The IBM Glossary of Computing Terms** at http://www.ibm.com/ibm/terminology/goc/gocmain.htm

    This glossary defines technical terms used in many IBM products. It is not a comprehensive resource of all IBM computing terms. This resource is provided for information purposes only and is updated periodically. IBM takes no responsibility for the accuracy of the information it contains.

Support

*   **AIX Fix Distribution Service Web site** at http://techsupport.services.ibm.com/rs6k/fixdb.html

    A Web facility for downloading AIX Version 4 and AIX Version 3 fixes, with a limited search engine designed with the assumption that you know what fix you need. If you do not know what fix you need, there is a pointer at the Web site to the APAR Database Facility. You can also contact your authorized IBM business partner or IBM Support Center.

*   **Ten Steps to Getting Support for WebSphere Application Server** at http://www7b.boulder.ibm.com/wsdd/support/appserver_support.html

    If you are new to a product, you might have difficulty finding all the information you need. And if you come across a problem, where do you go for help? Whether you are a new user looking for introductory information, or an experienced user looking for a workaround for a specific defect, you can benefit immediately from extensive Web-based support from IBM. It enables you to download fix packs, search on keywords, look up FAQs, Hints and Tips, and so forth. Always use this Web resource before contacting IBM Support directly.

*   **WebSphere Application Server Support page** at http://www-3.ibm.com/software/webservers/appserv/support.html

    Take advantage of the Web-based Support and Service resources from IBM to quickly find answers to your technical questions. You can easily access this extensive Web-based support through the IBM Software Support portal at URL `http://www-3.ibm.com/software/support/` and search by product category, or by product name. For example, if you are experiencing problems specific to WebSphere Application Server, click **WebSphere Application Server** in the product list. The WebSphere Application Server Support page appears.

Chapter 4. Quickly deploying Web components - Try it out!

Here is a quick way to deploy Web components, such as servlets and JSP files. This is not recommended as an official development method. It is provided so that you can sample the product functionality.

In summary, deploy Web components quickly by dropping the individual files into the directory structure of the default application installed by the product. This procedure relies on the Invoker servlet provided by the product. This servlet, enabled by default, lets you access deployed servlets by classname.

For *recommended* methods of developing and deploying Web application components, see the *Using Web applications* topic (tweb_aovr in the InfoCenter).

1. If deploying a servlet, first compile your servlet.
2. Copy the servlet or JSP class file into the directory of the default application.

   A servlet should be placed in the directory (split for publication):

   ```
   install_root/
       installedApps/
           cell_name/
               DefaultApplication.ear/
                   DefaultApplication.war/
                       WEB-INF/
                           classes
   ```

   If your servlet has a package statement, then create a subdirectory in the above directory for each level in your package statement.

   A JSP file should be placed in the directory:

   ```
   install_root/
       installedApps/
           cell_name/
               DefaultApplication.ear/
                   DefaultApplication.war
   ```

3. Open a browser window and request your servlet or JSP file.

   The URL is:

   ```
   http://your_host_name:9080/servlet/class_name
   ```

   The *class_name* variable is the Java class implementing the servlet or JSP file.

209

Chapter 5. Samples Gallery

The Samples Gallery available with the Application Server offers a set of samples that demonstrate common Web application tasks.

When you install the Enterprise product, you can choose to add samples to the gallery that demonstrate enterprise extension features. During a full installation of the base Application Server product or a typical installation of the Enterprise product, the Samples are installed on your local machine by default; however, if you choose a custom installation for either product, the Samples are optionally available. During an Enterprise product installation, a sample is available for each of the enterprise extensions that you choose to install. When you install an extension, it is then possible to add its sample to the Samples Gallery.

The Samples for the base WebSphere Application Server product install on your local machine by default if you choose a typical installation in the product installation wizard; however, if you choose a custom installation, the Samples are available as an option.

The Samples Gallery includes the following samples:
- The **Plants by WebSphere** application, which demonstrates several J2EE functions, using an online store that specializes in plant and garden tool sales.
- **Technology Samples**, which showcase enterprise beans, servlets, JavaServer Pages technology, message-driven beans, and J2EE application client.
- The **Java Pet Store Application**, which demonstrates J2EE technology, using an online pet store.
- The **message-driven beans** Samples demonstrate message-driven beans receiving messages from the point-to-point and Publish Subscribe messaging models. It also demonstrates Java Message Service (JMS) inside the client container.
- Samples that demonstrate new Version 5 enterprise extension features such as Extended Messaging, Dynamic Query Service, ActivitySessions Service, Application Profiling, JTA Extensions, Asynchronous Beans, Scheduler, and Process Choreographer.
- Improved samples for previously-available enterprise extension features such as Business Rules Beans, WorkArea Service, Internationalization Service, and CORBA C++ SDK.

Finding the Samples Gallery. Once the Samples are installed on your local machine, they are available to try out. Locate them at http://localhost:9080/WSsamples/. The default port is 9080. If you do not find the Samples on your localhost, confirm their installation and the port number for the internal HTTP server. On Windows platforms, you can also find the Samples by clicking **Start > Programs > IBM WebSphere > Application Server v5.0 > Samples Gallery**.

Client Samples. A separate Samples Gallery is available for the client Samples. To view these Samples, install the WebSphere Application Server client. The Client Samples Gallery demonstrates the following:
- J2EE application client.
- Java thin client.
- Applet client.
- ActiveX to EJB Bridge client.
- CORBA C++ SDK Client.

Code Examples. In addition to the Samples in the Samples Gallery, you can find other code examples in the InfoCenter by clicking **Quick reference > Examples** in the InfoCenter navigation.

The Samples are for demonstration purposes only. The code provided is not intended to run in a secured production environment. The Samples support Java 2 Security, therefore the Samples implement policy-based access control that checks for permissions on protected system resources, such as file I/O. The Samples do not support global security.

The Samples are not supported in a multi-server, clustered environment. Many of the Samples use Cloudscape as a persistent data store on the server. Only one instance of Cloudscape is supported per Java Virtual Machine (JVM). As a result, the second server in the node will fail to start the Sample applications, because an instance of Cloudscape has already been created with the first server in the node.

Additional WebSphere Application Server Samples are available on the IBM WebSphere Developer Domain, which is available at http://www.ibm.com/websphere/developer/library/samples/AppServer.html.

IBM ®

Printed in U.S.A.