© Copyright International Business Machines Corporation 2005. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Editorial/production supervision: *MetroVoice Publishing Services*
Cover design: *IBM Corporation*
Publisher: *Jeffrey Pepper*
Editorial assistant: *Linda Ramagnano*
Marketing manager: *Robin O'Brien*
IBM Consulting Editor: *Tara Woodman*

PRENTICE HALL PTR

Published by Pearson Education, Inc.
Publishing as Prentice Hall Professional Technical Reference
Upper Saddle River, NJ 07458

Prentice Hall PTR offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact: U.S. Corporate and Government Sales, 1-800-382-3419, corpsales@pearsontechgroup.com. For sales outside of the U.S., please contact: International Sales, 1-317-581-3793, international@pearsontechgroup.com.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other coutries, or both: DB2, Lotus, Tivoli, WebSphere, Rational, IBM, the IBM logo, and IBM Press.

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 the Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logo), 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.

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

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

First Printing

ISBN 0-13-144604-5

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd.
Pearson Education Canada, Ltd.
Pearson Educación de Mexico, S.A. de C.V.
Pearson Education — Japan
Pearson Education Malaysia, Pte. Ltd.

Foreword

This book was written by a team of experts that I've known for several years now and have only come to respect more and more. Leigh, Lavena, Roger, Shawn and Chris represent *the* driving force behind the WebSphere management system. The contribution of time and effort that they've put into this book is evidence of the dedication and confidence they have in what we're trying to accomplish with the WebSphere Application Server. The clarity and completeness of this book is an exquisite example of the precision and thoroughness they bring to their work. The resounding and enthusiastic acceptance we've gotten from customers of WebSphere, and especially its management system in Version 5, are a testament to their skill and leadership within our industry.

The Java Application Server industry is coalescing. The principles of application integration middleware and more specifically the use of component based technologies for implementing applications is beyond its infancy. Using an application server has become a critical success factor for production application development. Component based programming enables program modularity, composition and re-use. Application serving separates application business logic concerns from the concerns of the underlying information technology. Both of these attributes net increased programmer productivity. More importantly, they enable businesses to exploit information systems to benefit their business needs in a more timely fashion. Enterprises no longer spend time questioning whether they *should* use middleware—it's just accepted that middleware application servers yield tremendous benefits to business application developers and the businesses that run those applications.

However, application serving is about more than just developing applications to use component architectures. It is about creating a hosting environment for deployment and execution in which those applications and the resources they depend on can be managed to optimize the economy of those applications. The best managed hosting environments maximize the efficient utilization of computing resources and total systems throughput—minimizing the cost of the hosting environment at the same time as maximizing the productivity benefit to the business. The management system is essential to enabling those benefits. The degree to which the system can be controlled deterministically has a profound impact on the degree to which these positive economic results can be achieved under different circumstances. The degree to which this flexibility is encapsulated in well formed and intuitive concepts determines the degree to which these results can be achieved easily and reliably in any given circumstance.

With Version 5 of WebSphere we embarked on a significant re-architecture and re-engineering of the WebSphere management infrastructure. The decision to do so was painful. The prior management infrastructure was proving to be unscalable, unextensible, inconsistent across editions, and more fragile than we wanted for a mission-critical product. On the other hand, tearing into the management system at that level meant ripping up core elements of the application server itself. It meant investing in a massive undertaking, and taking on a significant risk. The team engaged in an intense debate over different approaches, factored in a tremendous number of new concepts and technologies that had to be integrated, and worked through huge technical challenges to find a balance between preserving release-to-release compatibility vs. enabling the value of other approaches as we transitioned from the old to the new. The team invested enormous hours and made tremendous personal sacrifices. The result is the stable, scalable, extensible and durable base for managing WebSphere that we see in the product now—one that continues to demonstrate those traits as we've moved through Versions 5.0, 5.1, 6.0 Technology for Developers and as we develop the next release of WebSphere. Not only has the WebSphere management system proved valuable to the foundation product, but this same management system is the foundation of the entire WebSphere Platform solution suite—including WebSphere Business Integrator Server and WebSphere Portal Server.

To be clear, the WebSphere management system is about *middleware management*, not *enterprise system management*. The distinction is subtle, but important. Middleware management confines itself to managing the concepts introduced by the middleware and the applications that are hosted by that middleware. Enterprise system management is about managing the entirety of the information system—including the application server, the data management or information integration system, any legacy enterprise integration systems, the network infrastructure, the underlying operating system and hardware, and everything else that contributes an end-to-end flow of information and business automation.

Nonetheless, the middleware management system plays a vital role in enabling the application server. First, it allows customers to manage the application server right out of the box—even if they don't have or use an enterprise management system. It also establishes a consistent and stable definition of the information model on which the application server can base its execution model. For example, the application server can assume that applications will be deployed in a particular way, prepared for execution, and located in a place where they can be safely loaded by the application server. The information model establishes the configuration attributes that affect server behavior, and basic topological concepts such as clustering that affect scaling assumptions for the server. The middleware management system establishes the basic lifecycle elements that affect availability, failover, recovery, and maintenance. Without middleware management, the application server is just an execution shell with limited potential to add value to the enterprise. On the other hand, when the application server is coupled with a management system, its potential for adding value to the business is amplified many fold.

Finally, the middleware management system plays an equally vital role to enterprise systems management as well. In particular, it proxies the application server and its information model to the enterprise—enabling the enterprise management system to leverage that definition as a point of consistency within its overall understanding of the information system.

Managing anything, whether it is people, your checkbook, a business, an information system, or anything else, can only ever be accomplished successfully if you have a clear understanding of the thing that you're managing. That means you have to conceive an image of what it is, how it behaves, including both its actions and reactions, what it is capable of doing (and not), what you want it do, and how to get it to do that. Managing WebSphere is no different.

The basic conceptual model of managing WebSphere is very simple: you create servers, you configure the resources available to each server, you deploy applications to those servers and attach them to the resources available on those servers, and then you start your applications. WebSphere gains a lot of value from the simplicity of that conceptual model.

The practical realities of management can also involve a lot more variation. For example, to get horizontal scalability and continuous availability you will likely want to incorporate several computers and cluster your servers. To protect your applications you will have to create a user registry, set authorization policies, and configure the level of protection you need. To interconnect your applications you will have to configure your messaging network. To optimize the performance and utilization of your system you will have to tune the execution environment. To gain efficiency you may collocate two or more applications on the same server. To maintain isolation between applications you may choose to place different applications on different servers. To maintain your applications you will need to manage the lifecycle of your applications.

Keeping track of all of the possible topology options, configuration parameters, control commands and deployment processes isn't easy. This book is the definitive bible and reference for understanding the conceptual model of administering WebSphere—starting from the architectural premise for topology and administration, through to the largest set of common activities and best practices that I've ever seen on WebSphere administration, and on to a complete listing of all of the user, scripting, command line and programming interfaces supported by WebSphere for managing a WebSphere installation. This is a book worth reading, and leaving at your finger tips for whenever you need guidance on how to manage WebSphere.

My greatest regards go to this team, and to all of you that have found ways to leverage the value of the WebSphere Application Server to benefit your enterprise.

—Rob High
IBM Distinguished Engineer and WebSphere foundation, Chief Architect
Member, IBM Academy of Technology

Preface

This book is part of a series intended to extend your understanding of WebSphere Application Server Version 5. It was written by some of the same team that created the administrative functions of the WebSphere Version 5 product. A great deal of effort went into the WebSphere product, and we put the same effort into making this book. You will find many pieces of inside information about how the administrative tools work and how to get the most from the product on a day-to-day basis.

Intended Audience for This Book

Administration of a product as sophisticated as WebSphere Application Server Version 5 can mean many things to different people. Some people just want to install their applications and run them. Others need to understand the details of the product architecture to plan deployment of their environment. Still others want to write custom Java code that controls one or more instances of WebSphere.

The primary target audience for this book is the administrator who wants to learn about all of the tools shipped with WebSphere Version 5 and how to make the most of those tools. As the title implies, this is a handbook for product administrators. We have tried to create a book that administrators will keep by their side, ready to consult at a moment's notice to recall the details of any specific task.

Substantial information is included about the Java programming interfaces for administration, and those who are interested in writing their own custom administration programs in Java should be able to use the material in Chapter 6 to accomplish this. However, the primary focus of the book is to help the administrator use the tools available with the product out of the box.

The focus of this handbook is on the how, more than the why. We attempt to compile information about how to perform any of the administrative tasks available for the product, using any of the several tools that come with WebSphere. Another book in this series, *IBM® WebSphere®: Deployment and Advanced Configuration* (forthcoming Prentice Hall, 2005), discusses many of the best practices for product deployment strategies, particularly in large enterprise-scale environments. In this administrator's handbook, we attempt to cover the details of how to accomplish each task involved in the various strategies described in that and other books.

What You Need to Know Before Reading This Book

To get the most out of this book, you should have an understanding of basic J2EE concepts. This book does not go into the details of J2EE application components or their development. Another book in this series, *Enterprise Java™ Programming with IBM® WebSphere™,* Second Edition (Addison-Wesley, 2004), is an ideal book to read to learn about the J2EE architecture and development techniques.

To get the most from Chapter 6 of this book, you should know the Java language and how to create and run general-purpose programs (not necessarily J2EE applications) in Java. Because this chapter describes how to create a program that performs administrative functions for WebSphere Version 5, and because all Version 5 administrative tasks are based on Java Management Extensions (JMX), you will be able to get more from the chapter if you have some familiarity with JMX and the JMX programming style. Several books in print currently provide a good foundation in this subject, including *Java™ and JMX: Building Manageable Systems* (Addison-Wesley, 2003).

Software Needed for the Examples

The only software needed to test the sample code in the book is the WebSphere Application Server Version 5 product itself. Most examples work in the Base Application Server edition, although some examples require the Network Deployment edition and these are indicated in the text. The required software (WebSphere Application Server Version 5) is available on the CD-ROM that accompanies this book.

Since Version 5.0 was released in late 2002, many subsequent fix packs and point releases have been delivered. Any of the Version 5 releases can be used to try the tasks described in this book. We have made an attempt to specifically indicate when a task is only applicable to a single Version 5 fix-pack release level.

CHAPTER 1

Introduction to WebSphere Administration

This book continues the series on WebSphere® Application Server Version 5 by focusing on the details of system administration for the product. Once you have developed your J2EE application, and ensured the quality of your application through testing, you are ready to put it into production and use the information from this book to deploy, monitor, tune, and manage your application and the WebSphere Application Server Version 5 environment in which it runs.

IBM® WebSphere® Application Server (hereafter called Application Server) Version 5 provides enhancements to scalability, reliability, Web services, J2EE™ 1.3 certification, and many other areas. Version 5 also provides a completely rewritten infrastructure for you to manage and administer your servers and applications. An open-standards-based management framework, Java™ Management Extensions (JMX), is at the core of the Version 5 management capabilities. New administration tool sets built for Version 5 take advantage of this framework. You can also use the Version 5 administration tool capabilities for your own custom administration programs.

This system administration book discusses a variety of ways to use the Application Server Version 5 management features. Chapter 1 introduces the basic system administration concepts needed to understand Version 5 features. The first important concept to grasp is the new packaging structure for Application Server 5. To understand Application Server administration, you also need to familiarize yourself with the following concepts: servers, nodes and node agents, cells, and the Deployment Manager. It is important that you understand the various processes in the administrative topology and the operating environment in which they apply. Chapter 1 also introduces the four administrative tool sets that are shipped as part of the WebSphere Application Server product.

Chapter 2 completes the foundation concepts needed throughout the rest of the book by delving into the details of the Application Server process internals, distributed administration service product features, administrative security, and the structure of the product configuration files.

Chapter 3 provides a complete reference to the Application Server command-line tools. Each tool is discussed, along with the details of its command-line options.

Chapter 4 is a reference for the Administrative Console program, a sophisticated J2EE Web application that provides the graphical management console for the product. All of the various tasks exposed in the console are covered. This chapter also introduces a set of scenarios for typical administration functions that are carried over to the subsequent chapters for scripting and programmatic administration. These scenarios provide a comparison between how a function can be accomplished through the console, through scripting , and using Java programming interfaces.

Chapter 5 covers the powerful administrative scripting capabilities built into WebSphere Version 5. The `wsadmin` scripting program supports three operating modes, multiple languages, full extensibility, and complete access to all of the WebSphere Version 5 administration functions.

Chapter 6 delves into the details surrounding programmatic administration, such as writing your own custom management program, extending the Application Server administration system, and using the same Java administrative programming interfaces (APIs) as are used to build all of the other tools shipped with the product.

WebSphere Application Server Structure

Application Server Version 5 provides an entirely new packaging structure. Several installation images build on one another to incrementally expand the features available to you. Start with the base product installation, and then add features (e.g., extended programming model enhancements or multi-node network deployment capabilities) as you need them. The two basic packages are Base Application Server and Network Deployment.

The WebSphere Version 5 product for z/OS and OS/390 includes both packages, along with an interactive configuration wizard, called the Customization Dialog, with which the WebSphere for z/OS user can configure base application servers and add network deployment functions.

The Base Application Server Package

A Base WebSphere Application Server installation includes everything needed for a single Application Server instance. Additional server definitions can be logically grouped into

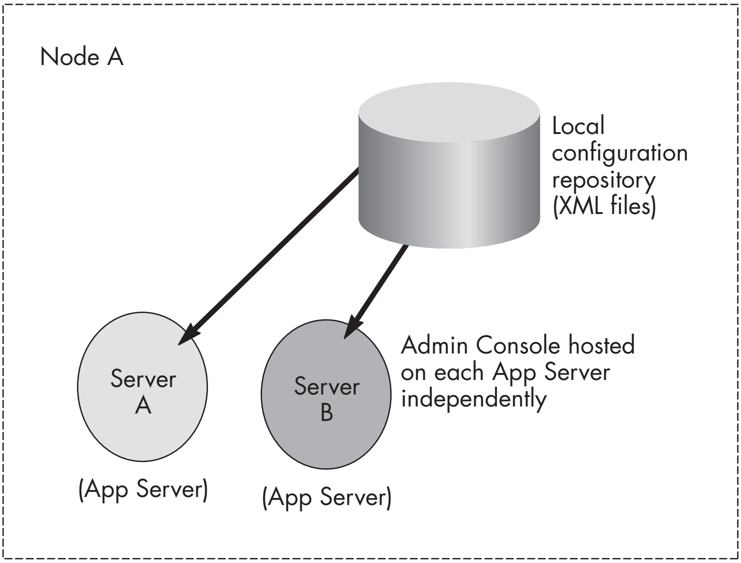

Node A
Local configuration repository (XML files)
Server A
(App Server)
Server B
(App Server)
Admin Console hosted on each App Server independently

Figure 1.1
Basic Application Server environment.

nodes. A node can contain many servers, but cannot span multiple computer systems. A single computer system can have multiple nodes installed on it, each with multiple managed servers. For example, multiple nodes defined on a large multiuser enterprise server computer makes better use of the system resources, and can isolate projects from one another. Figure 1.1 depicts the base environment.

A limitation of this package on its own is that it does not support the coordination between Application Server instances. Administration is limited to a single server at a time. Although you can create new server definitions from the Base console, you cannot use the console that is running in one server to start, stop, or otherwise manage a different server. The Network Deployment package extends the Base Application Server with capabilities for multiprocess, multinode configuration and control.

The Network Deployment Package

A Network Deployment installation can support a network of computer systems that are configured to run collaborating instances of the Base Application Server installation. The Network Deployment package provides centralized administration and workload management for a set of nodes, referred to as a *cell*. A cell has a master administrative repository that stores the configuration data for all of the nodes in the cell. Figure 1.2 depicts multiple systems in a Network Deployment cell, and shows that a base server can be added to a Network Deployment cell.

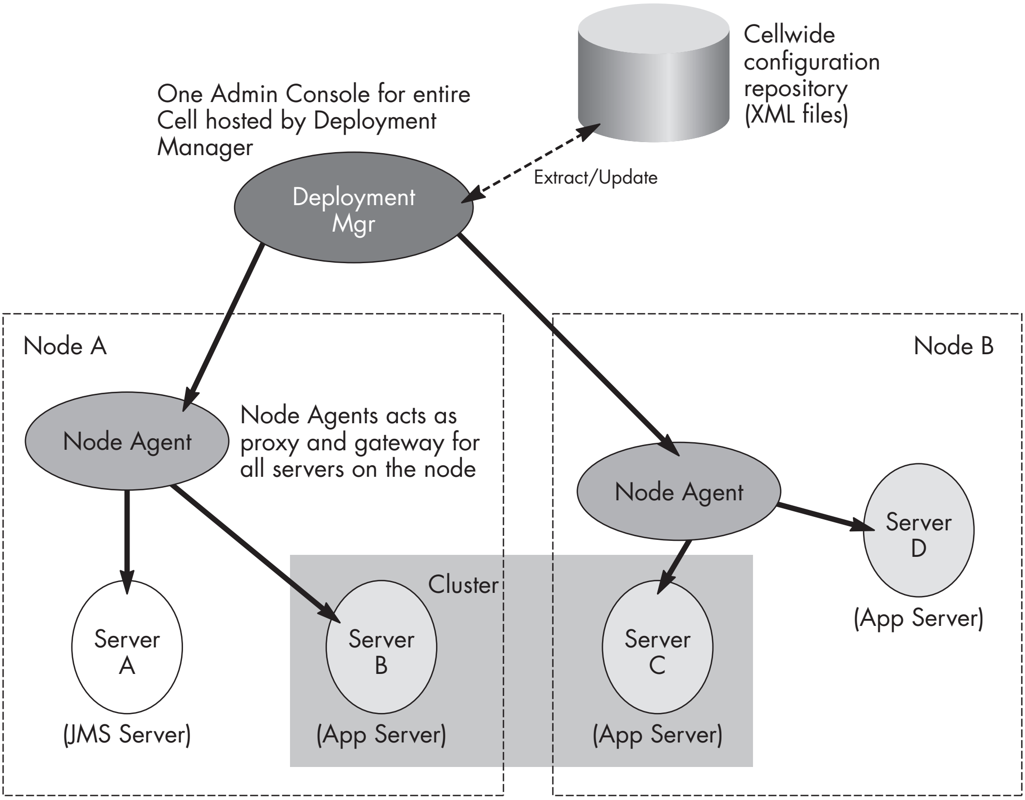

Cellwide configuration repository (XML files)
One Admin Console for entire Cell hosted by Deployment Manager
Extract/Update
Deployment Mgr
Node A
Node B
Node Agent
Node Agents acts as proxy and gateway for all servers on the node
Node Agent
Server D
(App Server)
Cluster
Server A
(JMS Server)
Server B
(App Server)
Server C
(App Server)

Figure 1.2
Network Deployment environment.

One computer system is designated as the central Deployment Manager machine onto which the Network Deployment package is installed. The central Deployment Manager program that is included in this package manages all of the nodes in the cell. Note that the same computer system that supports the Deployment Manager program can also support one or more federated Base Application Server nodes. The issue of whether or not to configure the Deployment Manager onto a separate computer system is more of a best practice concept pertinent to the subject of availability. For instance, large-scale customers such as those on z/OS enterprise servers might be unwilling to dedicate an entire logical partition (LPAR) to running the Deployment Manager alone. They might want to share system resources of that LPAR with other application server nodes. As with many aspects of Application Server Version 5, the choice of location for the Deployment Manager node and other application server nodes is one for each customer to make based on particular environment and operational policies.

To add a Base Application Server installation to the cell, run the `addNode` program on the base installation (the `addNode` program is fully described in Chapter 3 of this book). After this is completed, a separate node agent administrative server is created that serves as an intermediary between the application servers on the node and the Deployment Manager. Administrative logic that runs in the node agent keeps the configuration data for a node synchronized with the master configuration data for the cell.

Besides grouping servers into nodes, another logical grouping of servers is the *cluster*. A cluster can contain servers on different nodes. All of the servers in a cluster must have the identical application deployment configuration, because the purpose of a cluster is to define servers that collaborate for workload balancing and failover capabilities.

Comparison of the Administration in Versions 4 and 5

Before getting into the details of Application Server 5, it is useful to compare the implementation of system administration in Version 5 with Version 4. Those familiar with Version 4 administration, especially with Version 4 Advanced Edition, will be pleasantly surprised by all of the new management features in Version 5.

There are significant differences between how you would handle administration in Application Server Versions 4 and 5. One of the main differences is that the Version 4 Advanced Edition (AE) requires a database to hold configuration data, whereas no edition of Version 5 requires a database. Version 4 AE administration is based on J2EE Enterprise Java Beans (EJBs) and all of the Version 4 administrative programs are EJB client programs. Version 5 does not use EJBs to store configuration data; therefore, none of the Version 4 administration programs, such as the Swing console, wscp scripting (smapi for v4 on z/OS), and XMLConfig, are compatible with Version 5. Instead, Version 5 relies on Extensible Markup Language (XML) configuration files and industry-standard JMX components to handle management functions.

The Version 4 administration program is a single AdminServer program that serves several functions simultaneously. In Version 4, the AdminServer runs on every node, and every instance of the AdminServer is equivalent to any other. In Version 5, the same functions that were combined in the Version 4 AdminServer have been separated into different specialized administrative programs. The node agent program discussed earlier runs on every node and is specialized to perform node-specific administration functions, such as server monitoring, configuration synchronization, file transfer, and request routing. The single Deployment Manager program manages the entire cell, coordinating with the node agents for the various nodes in the cell.

Unlike Version 4, all administrative functions and programs are applicable to all editions of the product in Version 5. The same scripting program, `wsadmin`, that works for the Version 5 WebSphere Express edition also works for the full Enterprise package, even on the enterprise-class zSeries server machines. The same Administrative Console program, a J2EE Web application based on Java Server Pages (JSPs) and the Jakarta struts framework, works for all editions of Application Server Version 5.

Table 1.1 provides summary comparisons of some of the administrative features between Version 4 AE and Version 5.

Table 1.1 Comparison of Version 4 and Version 5 Administration

Administrative Function	Application Server Version 4 AE	Application Server Version 5
Administrative processes	AdminServer	Node Agent and Deployment Manager
Location of repository data	Relational database	XML configuration files
Graphical interface	"Fat" Swing client	"Thin" Web application
Scripting program	wscp (Tcl syntax) (REXX on z/OS)	wsadmin (Tcl syntax)
Backup of repository data to XML	XMLConfig (different format on z/OS)	None (already in XML)
Debugging utility	DrAdmin (IPCS on z/OS)	wsadmin
Java API	WscpCommand (Part of SMAPI on z/OS)	AdminClient
Program to start the server process	startServer (MVS START command on z/OS)	startServer
Install images	One	Two: Base install and ND install (except for z/OS which is a single image on tape)
Application binary distribution	None	On by default

Overview of Administration Tools for Version 5

System administration provides a variety of tools for managing WebSphere Application Server. These tools can be categorized into four general tool sets that are available with most editions of the product:

- Command-line tools
- Administrative Console
- Scripting tool
- Java programming APIs

Individual chapters of this book focus on in-depth details of each of these tool sets, but the following overview provides an introduction to the intended usage for the different tools.

Command-Line Tools

Command-line tools are simple programs that you run from a command prompt to perform specific tasks. Using the command-line tools, you can start and stop application servers, check server status, add or remove nodes, and complete similar tasks. The command-line tools provided with Application Server Version 5 are restricted for use on a single local node.

All of the command-line tools are Java programs that use the same Application Server Version 5 administration APIs as the console and the `wsadmin` tool, which are discussed in the next section. Chapter 3 provides a full list of the command-line tools available with Application Server v5.

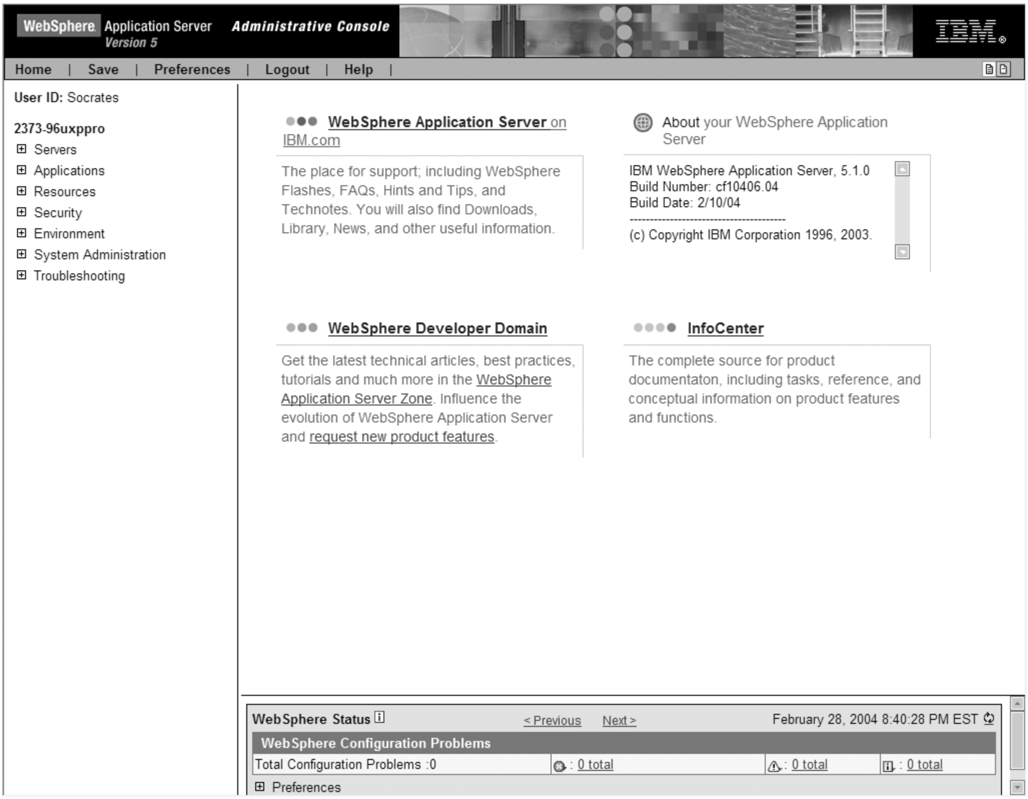
Most of the command-line tools print a usage syntax statement if you invoke them with the help option (by entering either `-?` or `-help` as part of the command). Most command-line tools also log their activity under the logs directory for the product. All command-line tools require authentication data when product security is enabled.

Administrative Console

The Administrative Console is a graphical interface that provides many features to guide you through deployment and systems administration tasks. It is extremely useful for helping you start exploring the available management options. Various wizards guide you through the more complicated processes. The Administrative Console program is documented in the Application Server Version 5 InfoCenter. Figure 1.3 shows the WebSphere Administrative Console home page.

The separation of run-time operations from configuration changes is an important concept in Version 5. Run-time requests are delivered to the running server components through JMX operations and take effect immediately. These run-time attribute changes are transient in nature and do not survive a server restart. Configuration changes, which are made in the XML configuration files for the server, are persistent across server restarts. Configuration changes do not take effect immediately; you must restart the server for the new values in the XML configuration files to be picked up.

Most run-time attributes have corresponding persistent configuration settings. However, there are considerably more configuration settings stored in the XML files than there are run-time attributes available for dynamic modification on managed objects while they are running in a server. Separating the two functions makes the distinction clear when you are changing something that will take effect immediately, but is transient as opposed to when



User ID: Socrates

2373-96uxppro

WebSphere Application Server on IBM.com

The place for support; including WebSphere Flashes, FAQs, Hints and Tips, and Technotes. You will also find Downloads, Library, News, and other useful information.

About your WebSphere Application Server

IBM WebSphere Application Server, 5.1.0
Build Number: cf10406.04
Build Date: 2/10/04

(c) Copyright IBM Corporation 1996, 2003.

WebSphere Developer Domain

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server and request new product features.

InfoCenter

The complete source for product documentaton, including tasks, reference, and conceptual information on product features and functions.

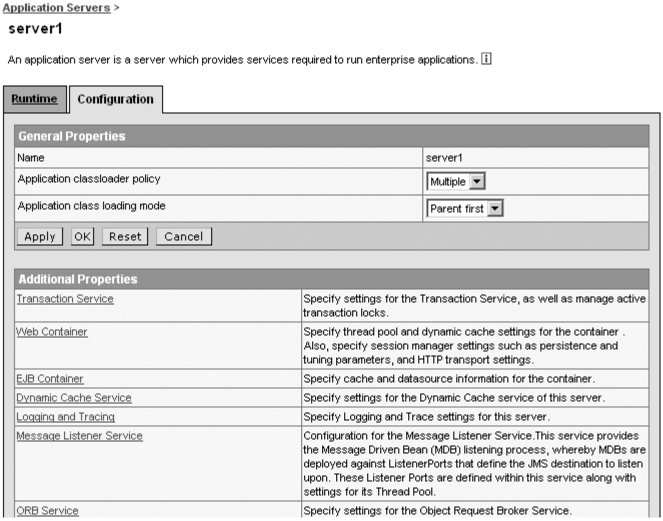
WebSphere Status February 28, 2004 8:40:28 PM EST

WebSphere Configuration Problems			
Total Configuration Problems :0	: 0 total	: 0 total	: 0 total

Figure 1.3
WebSphere Administrative Console.

you are making a persistent change. If you want both an immediate run-time change and a persistent configuration change, you need to perform both operations. Figure 1.4 shows a server page in the Administrative Console displaying both the server Runtime and Configuration tab views.

The Base Application Server version of the Administrative Console provides single-server administration capabilities. This Web application runs in the same server that it manages in the base environment. In a Network Deployment environment, the Administrative Console executes in the Deployment Manager server. This lets the console create server clusters that span multiple nodes and manage any process configured on any node within the cell. The Network Deployment environment allows you to manage multiple servers across multiple nodes whether they are clustered or not.



Application Servers >

server1

An application server is a server which provides services required to run enterprise applications.

General Properties	
Name	server1
Application classloader policy	Multiple
Application class loading mode	Parent first

Additional Properties	
Transaction Service	Specify settings for the Transaction Service, as well as manage active transaction locks.
Web Container	Specify thread pool and dynamic cache settings for the container . Also, specify session manager settings such as persistence and tuning parameters, and HTTP transport settings.
EJB Container	Specify cache and datasource information for the container.
Dynamic Cache Service	Specify settings for the Dynamic Cache service of this server.
Logging and Tracing	Specify Logging and Trace settings for this server.
Message Listener Service	Configuration for the Message Listener Service.This service provides the Message Driven Bean (MDB) listening process, whereby MDBs are deployed against ListenerPorts that define the JMS destination to listen upon. These Listener Ports are defined within this service along with settings for its Thread Pool.
ORB Service	Specify settings for the Object Request Broker Service.

Figure 1.4
Server page in WebSphere Administrative Console.

Use the Administrative Console client program to become familiar with the product and all of its capabilities. You can explore all of the various aspects of the environment in a graphical presentation. Once you have learned many of the details of Application Server Version 5 using the graphical console application, you might find that some of the other administrative tools provide faster access for day-to-day activities.

Scripting Tool

The Application Server administrative scripting program, `wsadmin`, is a powerful, nongraphical command interpreter environment that lets you execute administrative operations interactively or from a script file. The `wsadmin` tool is intended for production environments and unattended operations. The `wsadmin` tool is documented in the Application Server Version 5 InfoCenter. It is built on top of the Bean Scripting Framework that ships with Version 5. This lets the program support several languages for scripting Application Server administrative functions. The initial Version 5 release only supported the Tcl syntax, but additional scripting language support was added in release 5.1 when the Jython syntax was supported.

The `wsadmin` scripting tool has three modes of operation:

- **Interactive mode.** This lets the user enter commands and view the response on a command-line prompt. This mode is useful for learning the scripting tool and its capabilities. It is also useful for prototyping command syntax to verify the options before building a larger script.

- **Batch mode.** This lets the user supply a set of script commands in a file that the tool executes as a program.
- **Command mode.** This lets the user enter a single command from the regular operating system command window and executes this one command, returning control to the operating system command shell.

The `wsadmin` tool is most often executed as a client attached to a running server. You can also run it in a "local" execution mode where a running server is not required. In this mode, however, the function is limited to only configuration changes because a server run-time is not available to receive operational requests.

The `wsadmin` tool is primarily intended for rapidly assembling small control programs using the available Application Server administrative functions. You can develop more sophisticated administration programs using the Java API for Application Server administration (described next). However, the combination of full scripting language constructs, such as loops and variable evaluation, along with Application Server administration functions, provides powerful capabilities.

Java Programming API

Application Server v5 supports a Java programming interface for developing administrative programs. All of the administrative tools supplied with the product are written according to the API, which is based on the industry-standard JMX specification.

Using the administrative programming API, you can do the following:

- Write your own custom administration client to perform specific administration functions. The command-line tools available with Application Server Version 5, including the `wsadmin` tool and the console, are client programs that use the public administration APIs to carry out their tasks. Custom administration client programs can be simple or extremely complex. For example, you could write a client that only lets you start and stop clusters. You could also write a specialized administration client program to monitor certain metrics in the server and adjust configuration settings if the metrics exceed some threshold.
- Extend the basic Application Server administration system with your own custom MBeans that expose the management interface specifically aligned for your requirements. For example, your application might have its own run-time properties that you can adjust to tune the application while it is executing. Your application can implement a JMX MBean that exposes these attributes and other useful operational requests. Using the Version 5 administration programming interfaces, you can add your MBean to the set provided with Application Server, and control your application, along with the rest of the system, using the `wsadmin` scripting client.

You can even write a custom server extension and expose its functionality to the Application Server administration system as a JMX MBean.

The Application Server administration programming API is fully documented in the javadoc, which is provided with every installation (it is located in the `web/apidocs` directory under Application Server's root installation directory). The Application Server administration API is based on standard JMX interfaces and classes, and the JMX javadoc is also provided with each installation. The `com.ibm.websphere.management` package contains the public Application Server management interface.

Many helper classes and interface definitions are associated with the Version 5 administrative programming APIs. If you plan to create custom administration code, you should familiarize yourself with the public javadoc for the product. Chapter 6 in this book explores the details of how to use this administration programming API, and provides detailed examples of custom administration programs and system extensions.

Index

References to tables are indicated with a *t*; references to figures are indicated with an *f*; and references to code listings are indicated with a *c*.